On Inferring Browsing Activity on Smartphones via USB Power Analysis Side-channel

Qing Yang[†]* Paolo Gasti[‡] Gang Zhou[†] Aydin Farajidavar[‡] Kiran S. Balagani[‡♭]

[†]College of William and Mary – {qyang, gzhou}@cs.wm.edu
[‡]New York Institute of Technology – {pgasti, afarajid, kbalagan}@nyit.edu

Abstract—**In this paper, we show that public USB charging stations pose a significant privacy risk to smartphone users even when no data communication is possible between the station and the user's mobile device. We present a side-channel attack that allows a charging station to identify which webpages are loaded while the smartphone is charging. To evaluate this side-channel, we collected power traces of Alexa top 50 websites on multiple smartphones under several conditions, including: battery charging level, browser cache enabled/disabled, taps on the screen, WiFi/LTE, TLS encryption enabled/disabled, time elapsed between collection of training and testing data, and location of the website. The results of our evaluation show that the attack is highly successful: in many settings, we were able to achieve over 90% webpage identification accuracy. On the other hand, our experiments also show that this side-channel is sensitive to some of the aforementioned conditions. For instance, when training and testing traces were collected 70 days apart, accuracies were as low as 2.2%.**

Although there are studies that show that power-based side-channels can predict browsing activity on laptops, our work is unique because it is the first to study this side-channel on smartphones, under smartphone-specific constraints. Further, we demonstrate that websites can be correctly identified within a short timespan of two to six seconds, which is in contrast with prior work which uses 15-second traces. This is important because users typically spend less than 15 seconds on a webpage.

Index Terms—**Side-channel attacks, Privacy, Activity recognition, Smartphone security**

I. Introduction

Users commonly rely on their smartphones for a variety of energy-intensive activities such as video streaming, web browsing, connection sharing (hotspot), gaming, and VoIP. As a consequence, recharging smartphones multiple times a day has become common practice. A recent survey [1] found that users in the United States charge their smartphones anywhere from 1.8 to 2.6 times a day on average. To address users' charging needs, USB charging stations (or kiosks) are becoming ubiquitous in many public areas, including airports [2], parks [3], [4], hotels [5], and hospitals [6].

While charging stations undoubtedly provide convenience to smartphone users, they also expose them to privacy threats. For example, in an attack called "juice-jacking", an untrusted charging station exfiltrates data by covertly setting the smartphone into USB transfer mode [7]. To counter data exfiltration attacks, hardware devices such as SyncStop [8] are designed to physically interrupt data wires at the USB port, thereby blocking all data transfers. In this paper, we show that even physically interrupting USB data wires does not prevent a charging station from extracting sensitive information from the smartphone. In particular, we demonstrate that a malicious charging station can infer smartphone browsing activity by analyzing USB power consumption patterns. In our experiments, we were able to identify with high accuracy which webpage the user was visiting, by applying standard machine learning techniques to power traces.

We believe that our attack represents a serious threat to privacy, because: (1) USB charging stations are becoming widespread around the world, and therefore more and more users can be targeted by this attack; and (2) the modifications required to implement the attack can be easily concealed (e.g., see [9]), and can therefore go unnoticed by users.

A. Contributions and Findings

To fully characterize the side-channel associated with USB power consumption, we analyzed how webpage identification accuracy is impacted by variables pertinent to mobile devices, such as battery charging level, wireless connection (WiFi or LTE), and taps on the screen. To our knowledge, this is the first work that tests the feasibility of this attack against constraints commonly encountered in a mobile environment. In addition, we determined the impact of other variables that have not been considered in prior studies on power-based side-channels [10]. These include availability of browser cache, training and testing signals collected on different smartphones, the time elapsed between the collection of training and testing signals, geographical proximity between the user and the web server, duration of power traces, and availability of encrypted (TLS) connections on identification accuracy. Next, we summarize our findings.

Impact of Battery Charge Level and Taps on Screen. Both charging the battery and tapping on the screen reduced webpage identification accuracy. Identification accuracy decreased when the smartphone's battery was charging at 30% level,

This work was supported by the National Science Foundation Secure and Trustworthy Cyberspace (SaTC) Grant #1619023, and by a 2015 New York Institute of Technology Institutional Support for Research and Creativity (ISRC) Grant.

*This work was carried out while visiting the New York Institute of Technology.

[♭]Corresponding author; Mailing address: Room HSH 226C, Northern Blvd., Old Westbury, NY 11568, USA. Phone: 561-686-1373. Fax: 516-686-7933.

compared to when the battery was fully charged. However, even with the decrease in accuracy, it was still possible to reliably infer browsing information. Similarly, taps on the screen added significant noise to the power traces, making webpage identification challenging. Our results show that this factor caused a significant degradation in identification performance.

Impact of Other Variables. The time elapsed between collection of training and testing traces had a major impact on webpage identification accuracy, with training traces older than 30 days leading to a significant drop in identification accuracy. This suggests that traces used to train the classifiers should be updated frequently to improve the attack's success rate.

Increasing the duration of power traces for training and testing led to improved identification accuracy.

We were able to reliably identify webpages under both WiFi and LTE, even when the training power traces were collected using one type of connectivity (e.g., LTE), and the testing traces were obtained with another (e.g., WiFi).

When using different smartphones for training and testing, accuracies dropped significantly. However, using two smartphones for training, and a different smartphone for testing reduced the drop in webpage identification accuracies.

When the user did not tap on the screen, enabling browser cache improved identification accuracy. However, for power traces collected when the user tapped on the screen and while the smartphone was charging, enabling cache led to a decrease in webpage identification accuracies.

Our results show that increasing the geographical distance between the smartphone and the host serving the webpages reduced identification accuracy. When we divided webpages in foreign (located outside of the continental United States), and local (within the United States), we observed that webpages hosted locally had slightly higher identification accuracies than foreign-hosted webpages.

Finally, retrieving webpages via secure connections (HTTPS or, more specifically, TLS) did not have a measurable impact on identification accuracy.

Experiment Results. In our experiments, we used machine learning algorithms to identify which webpage the user visited out of a closed set of fifty webpages [11]. We were able to achieve identification accuracies as high as 98.8% with 2-second traces. Even in the worst case, i.e., when the cache was enabled, the user tapped on the screen, and the battery was charging from 30%, we achieved an identification accuracy of 54.2% with 6-second traces. When training and testing traces were collected using different smartphones, identification accuracy was at least 44.5%. This is significantly higher than choosing one out of fifty webpages at random (which leads to 2% baseline accuracy).

B. Organization.

We present the related work in Section II. Our setup and data collection procedure is detailed in Section III. Section IV describes our webpage identification technique. Evaluation of our technique is presented in Section V. The impact of (1) different devices for training and testing, (2) WiFi and LTE, (3) training data aging, (4) website location, and (5) connection type (HTTP/HTTPS) is analyzed in Section VI, while the normalized accuracy of various ranks is discussed in Section VII. We conclude in Section VIII. Details on the dataset using in this paper are provided in Appendix.

II. Related Work

In this section, we review prior work on: (1) side-channel attacks using power analysis; (2) webpage fingerprinting; and (3) attacks on smartphones via USB port.

A. Side-channel Attacks Using Power Analysis

Clark et al. [10] identified webpages loaded on a computer by measuring power consumption at the AC wall outlet. Power traces were analyzed in the frequency domain, and matched using a SVM classifier. The evaluation results showed that this side-channel attack achieved 86.75% precision and 74% recall.

The main differences between our work and [10] are as follows. (1) Ours is the first to study this side-channel on smartphones, which drastically differ in architecture from desktops and laptops, and have hardware (e.g., a touchscreen) not usually associated with traditional computers. Additionally, smartphones adopt aggressive dynamic power optimization techniques [12] that could interfere with the side-channel. (2) In contrast with [10], our work explores how the identification accuracy is affected by variables such as battery charging level, user's interaction with the touchscreen, trace length, time between training of our identification model and testing, type of wireless connection (WiFi and LTE), and website characteristics (HTTP vs HTTPS, geographical location). (3) The results reported in [10] were obtained with 15-second traces, compared to 2- to 6-second traces in our work. We consider this an important distinction, since trace length has a large impact on identification accuracy, and most users spend *less* than 15 seconds on average on a webpage [13].

Several papers analyzed power data from sources other than USB charging ports or AC outlets to extract cryptographic keys or other private information. For instance, Genkin et al. [14] demonstrated a side-channel attack based on electric potential from computer chassis to extract RSA and ElGamal keys. Their work is based on the observation that the fluctuation in electric potential of the chassis correlates with computation.

Michalevsky et al. [15] introduced PowerSpy, a tool that analyzes aggregate power consumption on the phone during a period of several minutes to infer the user's location. The authors observed that the power consumption of a smartphone is measurably affected by the smartphone's distance from the surrounding cellular base station. To obtain instantaneous power consumption, PowerSpy uses unprivileged Android APIs. For this reason, PowerSpy infers location information without requesting any explicit permission to the user, in contrast with the use of *privileged* location APIs—which require explicit user confirmation.

Lin et al. [16] demonstrated that it is possible to identify running apps, details about on-screen content, password lengths, and geographical locations by measuring instantaneous power consumption. Our work differs from [16] in the following ways. (1) Li et al. do not focus on website identification. (2) They use a different adversary model, which assumes that the adversary is able to install a malicious app on the smartphone, or is able to measure power consumption directly at the battery connectors inside the smartphone, rather than at the USB port. Our attack uses USB charging port to obtain power traces, and does not require the smartphone to run malicious software. (3) Li et al. do not consider factors common to smartphones, such as the current battery level or the type of network used to retrieve data. These factors, as demonstrated by our experiments, affect the accuracy of the power-based side channel attack.

B. Webpage Fingerprinting via Traffic Analysis

Several side-channel attacks use network traffic analysis to infer user's web browsing activities. For example, Hintz et al. [17] demonstrated that transferred file sizes could be used as a reliable fingerprint for webpages. Similarly, Lu et al. [18] exploited both packet size and packet ordering information to improve webpage identification success rate.

Additionally, studies have shown that encrypted channels do not protect against traffic analysis. For example, Chen et al. [19] were able to infer browsing activity via packet analysis on traffic encrypted using HTTPS and WPA.

Our work adds to the current body of work on webpage fingerprinting by demonstrating a new and complementary side-channel.

C. USB Data Port Vulnerabilities

Karsten et al. [20] and Andy et al. [21] demonstrated that it is possible to install malicious software on smartphones via their USB port by exploiting vulnerabilities in mobile operating systems. To prevent these attacks, the authors suggest to disable/remove data pins in the USB cable. Unfortunately, our work shows that even if data pins are removed, the charging station can still learn information about the user's browsing activity.

III. Experiment Setup

We collected power traces while webpages were loading on two types of smartphones: Samsung Galaxy S4, and Samsung Galaxy S6. We used the homepages of the 50 most popular (non-adult) websites, based on Alexa ranking [11] (see Table IX in Appendix). These websites represent over 30% of the page views on the Internet [10]. To collect power traces during webpage loading, we instrumented the USB charging circuit as shown in Figure 1. Our circuit connects a DC power supply, a smartphone, and a data acquisition card (DAQ), and measures voltage variations (and therefore the corresponding power consumed) across a 0.1 Ω shunt resistor. To satisfy the USB charging specifications [22], we connected the data pins of the USB cable using a 200 Ω resistor.

Most smartphones use lithium-ion (Li-ion) batteries due to their high energy density. The charging profile of Li-ion batteries encompasses two stages [23]. In the first stage, the smartphone charging circuit applies *constant current* to the battery. This stage ends when the battery reaches a specific charging voltage (usually between 3.7 V and 4.2 V). In the second stage, the battery is charged at a *constant voltage*, and the current gradually decreases until it reaches a termination value. Because the battery charging process could take several hours, the current used to charge the battery does not vary significantly while the smartphone is loading a webpage.

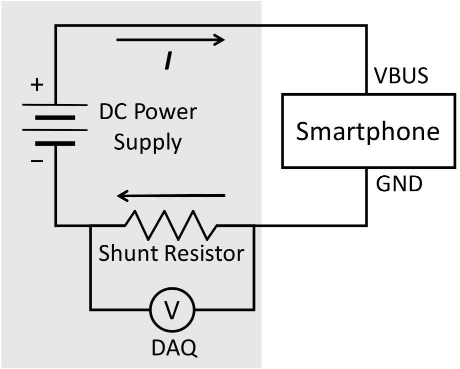


Fig. 1. Overview of the setup used to collect power traces.

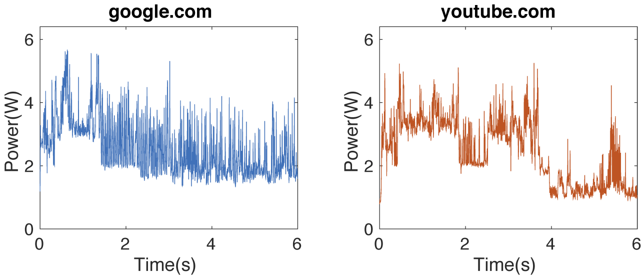


Fig. 2. Power traces collected during the first 6 seconds of automated webpage loading activity. The left and right panels show power traces collected while loading `google.com`, and `youtube.com`, respectively. The x axis shows time (in seconds) from the beginning of the webpage loading activity, and the y axis shows the power consumed by the smartphone.

We used an Agilent E3630A DC power supply [24] as the power source. We measured the voltage drop across the shunt resistor using a National Instrument USB-6211 (DAQ) [25] at a sampling rate of 200 kHz. We set the power supply to output a fixed voltage of 5.5 V. This voltage is higher than the nominal USB voltage of 5 V to compensate for the voltage drop introduced by the shunt resistor. The resulting voltage was between 5.32 V and 5.48 V, which is within the tolerance of many modern smartphones [22]. The DAQ's data output was connected to a laptop, which stored data for offline analysis using LabVIEW. Figure 2 shows the power consumption traces collected while loading the homepages of `google.com` and `youtube.com`.

We collected power traces in two modes: *user-actuated*, and *automated*. With user-actuated traces, the user initiates webpage loading by typing a URL in Mobile Chrome's address bar. To collect automated traces, we developed an Android application that launches the Chrome browser, and uses it to load the intended webpage. Our application allows 10 seconds for webpage loading (only the first 6 seconds of data were recorded), and then loads the next webpage. Before each measurement, we closed all other applications on the smartphone, and set the screen brightness to a constant level.

User-actuated and automated traces were collected under two conditions: battery level (30% vs. 100%),[1] and browser cache (enabled vs. disabled). We chose these conditions because they impact smartphone energy consumption. When the battery is fully charged, almost all power from the charger is used to load webpages. In contrast, when the smartphone is charging, a sizable (almost constant) amount of power is used to charge the battery, hence affecting the traces. Cache availability was chosen because cache misses increase network activity, and therefore radio activity. Retrieving data wirelessly requires more energy than loading it from local flash memory.

We collected 40 automated traces per webpage for each of the following combinations: 30% battery, cache; 30% battery, no cache; 100% battery, cache; 100% battery, no cache. Additionally, we collected 10 user-actuated traces for the same combinations. To collect traces, we used four Samsung Galaxy S4 devices (in the rest of the paper, we refer to these devices as D1, D2, D3, and D4) in most of the experiments. To analyze the impact of different smartphone models on the attack, we also conducted experiments on a Samsung Galaxy S6 (referred to as D5). We collected all traces in Old Westbury (NY), and in Williamsburg (VA). Further details on our datasets and their usage in our paper are provided in Table X in Appendix.

IV. Webpage Identification

Our webpage identification process consists of *training* and *testing* phases. In the training phase: (1) we extracted frequency-domain features from the power traces; and (2) we trained a classifier (Random Forest [26]) on the extracted feature vectors. During the testing phase, we use the trained classifier to predict webpage labels on new data. To improve identification accuracy, we performed feature extraction on partially overlapping segments from traces, and implemented classifier voting. Next, we provide details on feature extraction, classification, and trace segmentation.

Classifier Training and Testing. We used Random Forest [26] to classify power traces because in our experiments it outperformed other commonly used classifiers, such as SVM [27], [28], and Dynamic Time Warping (DTW) [29]. We used the WEKA [30] implementation of Random Forest.

We experimented with four training-testing scenarios. The first involved 40 power traces per webpage, collected using automated webpage loading; 20 traces were used for training the classifier, and the remaining 20 traces for testing. This scenario is used when training and testing are performed with data from the same smartphone, such as in tables I, III, VI, VII, VIII.

In the second scenario, we trained the classifier using all 40 automatically-collected traces, and performed testing with 10 traces collected via user-actuated page loading. This scenario was used with data is collected with user taps, i.e., in Table II.

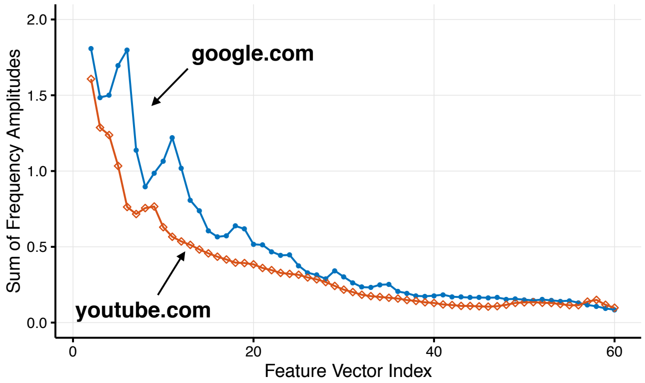


Fig. 3. Aggregate amplitudes corresponding to the first 60 bins computed from the power traces of `google.com` and `youtube.com`. (Corresponding time-domain traces are illustrated in Figure 2.) The x axis represents the index of each bin.

In the third scenario, we trained our classifier using 40 traces per webpage, collected using automated webpage loading, on one smartphone device; we then used 40 traces collected from a different smartphone device for testing. This scenario was used in tables IV and V, and in Figure 6.

Finally, in the fourth scenario, we trained the classifier using 80 traces from two smartphones (40 traces from each device), and tested on 40 traces from a different smartphone. This scenario was used in tables IV and V, and in Figure 6.

Feature Extraction. We transformed each power trace to its corresponding frequency-domain representation using Fast Fourier Transform (FFT) [31]. To reduce the impact of noise on individual frequencies, we divided the frequency range into equal-size bins. For each bin, we calculated the average amplitude of all the frequencies in it. The average amplitudes were used as features. We experimented with different numbers of bins, and settled on using 125 bins (resulting in 125 features) when using the same device for training and testing, because this value led to the highest identification accuracy. When using different devices and smartphone models during training and testing, we used the first 15 of 125 bins (i.e., first 15 features) corresponding to the 15 lowest frequencies, which allowed us to achieve the highest identification accuracy for this setting. Figure 3 shows the result of feature extraction on the data in Figure 2. Each data point in Figure 3 represents a feature.

Trace Segmentation and Voting. Variable network conditions, web-server load, and smartphone background applications introduce intermittent noise in power traces. To mitigate the effects of noise, we divided each trace into overlapping 0.5-second segments. Feature extraction was performed on each segment, and the classifier was trained using segments from all traces. For testing, we classified individual segments of a session, and used the classification results as votes. Each trace was then assigned to the class that received the largest number of votes. This strategy of segmenting and using majority voting led to an improvement in accuracy between 0.1% and 7.7% compared to using entire traces.

[1] We implemented the 30% battery level condition as follows. We charged the smartphone battery to 30% level, and then started collecting power traces while still charging the smartphone. As the battery reached 35% level, we interrupted data collection, discharged it to 30%, and resumed the collection of power traces.

Evaluation of Identification Performance. To evaluate classifier performance, we calculated Rank 1 and Rank 5 identification accuracies. With Rank 1, a trace is classified correctly if the most popular label assigned to the trace's segments is the correct label for the trace. In case of ties, the classifier outputs the label with the highest associated confidence, defined as Random forest leaf node probability. With Rank 5, we consider a trace as correctly classified if the correct label appears within the 5 most popular labels.

Because our identification task involves predicting 50 webpages, the baseline accuracies, obtained by randomly guessing the webpages, are 2% for Rank 1 and 10% for Rank 5.

Finally, for each rank, we also present the Normalized Rank-n Accuracy, which is defined as follows. Let p_n be the probability that the classifier correctly labels a trace for Rank-n. The probability of correctly guessing the website loaded by the smartphone is computed as p_n/n, and represents the probability that the adversary guesses the correct website label given the Rank-n output of the classifier.

We consider this metric because it represents the uncertainty an adversary encounters in identifying the website correctly as the rank increases. For example, if Rank 2 accuracy is 60%, and Rank 5 accuracy is 95%, then the normalized Rank 2 accuracy is 30%, while the normalized Rank 5 accuracy is 19%. This shows that if the adversary is interested in identifying a unique website visited by the user, Rank 2 leads to better results. On the other hand, if the adversary only needs to know if the user is visiting any of the five webpages identified in Rank 5 (e.g., because they are all from social networking websites), then the Rank 5 result is more meaningful. Therefore, the information provided by this metric complements the Rank 1 and Rank 5 accuracy results presented throughout the paper.

V. Evaluation

We evaluated identification accuracy under three variables: (1) trace length (2 s, 4 s, and 6 s), (2) cache enabled vs. disabled, and (3) battery charge level (30% vs. 100%). In this section, we report how these variables impact identification accuracy for *automated* and *user-actuated* data collection. Identification accuracies under other variables, such as different smartphones used in training and testing, type of wireless connectivity, time elapsed between collection of training and testing traces, location of the hosts serving the webpages, and HTTP vs. HTTPS, are reported in sections VI-C and VI-D.

A. Identification Accuracy on Automated Dataset

Our results for the automated dataset are reported in Table I. We achieved identification accuracy of at least 82.8% for Rank 1, and at least 92.8% for Rank 5 using a Samsung Galaxy S4, with 2 second traces. With a Samsung Galaxy S6, accuracies were at least 75.2% for Rank 1, and at least 91.8% for Rank 5. Next, we discuss how each variable affects identification accuracy.

Trace Duration. Increasing the duration of the traces led to an improvement of identification accuracy. Although we achieved the highest identification accuracy with six-second traces, we also had good identification accuracies (Rank 1: 82.8%-98.8%) with 2-second traces, as shown in Table I. Given that most webpages load within a few seconds on smartphones (2.9 s to 5 s on average, according to [32]), our results indicate that the attack correctly identifies a webpage within the typical loading time.

TABLE I
Identification Accuracy (in %) using frequency-domain features and classifier voting for automated dataset collected using D1. For comparison, results from a Samsung Galaxy S6 (D5) are also reported. All experiments were performed using 125 features.

	Rank 1			Rank 5		
	2 s	4 s	6 s	2 s	4 s	6 s
Charged, no cache	96.7	99.0	99.3	99.0	99.4	99.8
Charged, w/ cache	98.8	99.6	100	99.6	100	100
Charging (30%), no cache	82.8	91.7	95.7	92.8	96.7	98.5
Charging (30%), w/ cache	87.6	94.4	96.2	94.5	97.4	97.6
S6, charged, no cache	84.3	94.8	97.1	94.5	98.6	99.5
S6, charging (30%), no cache	75.2	82.8	90.5	91.8	93.7	97.8

Caching. Our results show that enabling cache improved identification accuracy (see Table I). This is further validated by our results, reported in Table VII, where enabling cache improved identification accuracy for foreign-hosted websites more than for websites located within the United States. This is because the farther the host serving the content, the more network-related noise is added to the traces (see Section VI-D for further details).

Battery Level. Users connect their smartphones to charging ports at various battery levels [1]. Our experiments show that this variability impacts identification accuracy. In particular, we were consistently able to classify traces with higher accuracy when the battery was fully charged (see Table I).

This can be explained by comparing the power traces illustrated in Figure 4. The USB charging specifications set an upper bound of 1.8 A to the amount of current that the smartphone can draw from the USB port. If the phone is charging, a substantial amount of the available current is directed to the battery, and therefore the fluctuations in power consumption due to webpage loading is limited. This is evident in Figure 4, where the signal corresponding to 30% battery has considerably lower variability compared to the signal collected with fully charged battery.

B. Identification Accuracy on User-Actuated Dataset

Once we included user activity in the form of taps, identification accuracy dropped significantly due to tap-induced noise. This is because tap characteristics (e.g., tap location on the screen, timing, and duration) are different in each trace, which leads to noisy traces. To validate this observation, we computed the average (intra-class) Dynamic Time Warp (DTW) distance between pairs of user-actuated traces, and between pairs of automated traces, under different caching and charging conditions. The average distance between user-actuated traces was consistently higher than the distance between automated traces. With cache enabled and fully charged battery, the

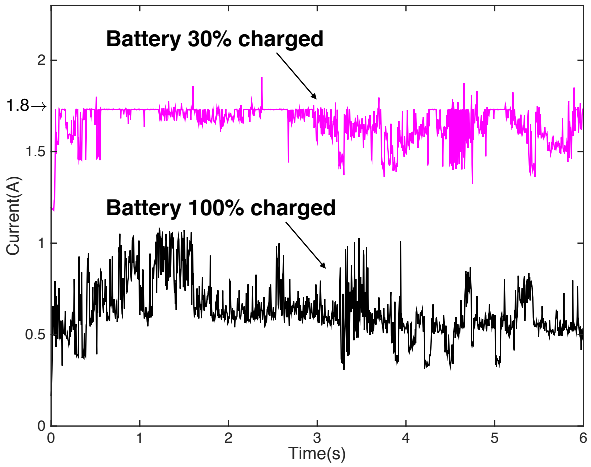


Fig. 4. Power traces obtained while loading the same webpage (`yahoo.com`) with 30% and 100% battery level. The figure shows that the trace corresponding to 100% battery level exhibits a relatively higher dynamic range, because it is not capped by the 1.8 A limit which is often reached by the trace corresponding to 30% battery level.

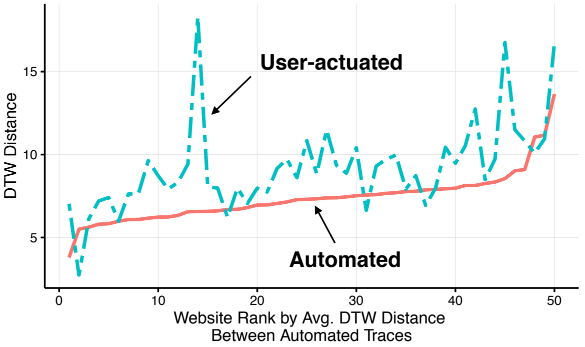


Fig. 5. Average inter-class DTW distance on automated traces for each webpage. Measurements are performed with battery fully charged, and with cache enabled. Data is sorted in ascending order of average DTW distance.

TABLE II
IDENTIFICATION ACCURACY (IN %) USING FREQUENCY DOMAIN FEATURES AND CLASSIFIER VOTING FOR USER-ACTUATED TRACES (I.E., WITH TAPS) COLLECTED USING D1

. All results were obtained using 125 features.

	Rank 1			Rank 5		
	2 s	4 s	6 s	2 s	4 s	6 s
Charged, no cache	9.0	56.0	82.6	19.0	79.2	92.8
Charged, w/ cache	7.0	58.6	88.4	22.4	83.0	95.0
Charging (30%), no cache	5.8	39.2	70.0	18.8	70.0	88.4
Charging (30%), w/ cache	4.2	35.8	54.2	15.8	53.6	72.0

TABLE III
IDENTIFICATION ACCURACY (IN %) USING FREQUENCY-DOMAIN FEATURES AND CLASSIFIER VOTING FOR AUTOMATED DATASET COLLECTED USING D2. ALL EXPERIMENTS WERE PERFORMED WITH A FULLY-CHARGED BATTERY, AND NO CACHE, USING 125 FEATURES.

	Rank 1			Rank 5		
	2 s	4 s	6 s	2 s	4 s	6 s
LTE training, LTE testing	82.7	98.2	99.7	94.2	100	100
LTE training, WiFi testing	24.9	55.2	73.9	45.2	72.8	86.0
WiFi training, LTE testing	23.8	68.6	84.8	50.3	85.1	95.7

average DTW distance between user-actuated traces was 17%-38% higher than the average distance between automatically-collected traces. When the smartphone was charging from 30% level, and with cache enabled, the DTW distance increased by 34%-86% when including taps. Once cache was disabled, the average DTW distance increased by 5%-15% with taps and with the battery fully charged. With no cache and battery charging from 30% level, the average DTW distance between automated and user-actuated traces increased by 31%-56%. Figure 5 shows the relative distance for each webpage with fully charged battery and cache enabled.

While two seconds were sufficient to classify webpages with high confidence using automated traces, this was not the case with traces from the user-actuated dataset. Regardless of caching and charging, we achieved good Rank 1 accuracy with six-second traces. In this setting, we correctly identified webpages between 54.2% and 88.4% of the times (for reference, selecting a webpage at random out of 50 leads to 2% accuracy). We achieved good Rank 5 accuracy (53.6% to 95%, compared to 10% with random chance) with four- and six-second traces. Charging had still a measurable impact on identification accuracy: identification on a fully charged phone consistently led to better results, all other variables being the same. Enabling cache with fully charged battery led to a 2.2%-5.8% improvement in identification accuracy. However, when we enabled cache in traces collected at 30% battery level, webpage identification accuracy dropped significantly. We hypothesize that this happened because taps on screen and battery charging contribute to a substantial increase in noise in power traces. Under these conditions, networking becomes an important source of discriminability between website traces. Enabling cache reduces the contribution of the network component to the power traces, in turn reducing classification accuracy.

Overall, our experiments show that although the presence of taps substantially reduces identification accuracy compared to automated collection of power traces, it is still possible to accurately classify six-second user-actuated traces.

VI. IMPACT OF OTHER VARIABLES

We examined the identification accuracies according to the following variables: (1) different smartphones used for training and testing (training traces were collected from one or more smartphones that are not used for testing); (2) LTE and WiFi training and testing; (3) aging of training traces; (4) domestic vs. foreign websites, and (5) websites accessible via unencrypted connections (denoted as "HTTP") vs. accessible through TLS-encrypted links (denoted as "HTTPS"). Table IX in Appendix indicates which websites are local, and/or accessible over HTTPS. Next, we provide details on our findings with respect to each variable. For all experiments in this section, we used only automated traces.

A. Training and Testing Traces from Different Devices

Tables IV and V summarize our result when using different smartphones for training and testing. Using one smartphone for training, and a different smartphone for testing led to a significant drop in identification accuracy. On the other hand,

TABLE IV
IDENTIFICATION ACCURACY (IN %) USING FREQUENCY-DOMAIN FEATURES AND CLASSIFIER VOTING FOR AUTOMATED DATASET. ALL EXPERIMENTS WERE PERFORMED WITH A FULLY-CHARGED BATTERY, AND NO CACHE, USING 15 FEATURES.

Testing / Training	D3			D4			D2		
	2 s	4 s	6 s	2 s	4 s	6 s	2 s	4 s	6 s
D3	64.3	89.9	94.9	48.8	73.0	77.5	34.3	60.0	69.7
D4	48.4	73.1	79.9	67.7	93.5	96.6	43.1	71.1	75.0
D2	34.3	63.0	69.8	44.1	74.2	75.9	66.0	91.1	94.6
D4+D2	52.7	78.4	84.3		—			—	
D3+D2		—		56.7	84.4	85.8		—	
D3+D4		—			—		45.9	75.9	81.1

TABLE V
IDENTIFICATION ACCURACY (IN %) USING FREQUENCY-DOMAIN FEATURES AND CLASSIFIER VOTING FOR AUTOMATED DATASET. ALL EXPERIMENTS WERE PERFORMED WITH THE BATTERY CHARGING FROM 30%, AND NO CACHE, USING 15 FEATURES.

Testing / Training	D3			D4			D2		
	2 s	4 s	6 s	2 s	4 s	6 s	2 s	4 s	6 s
D3	48.5	67.8	79.6	40.3	56.6	59.9	31.9	38.8	50.4
D4	40.6	58.7	67.8	41.7	61.8	69.6	22.8	36.6	51.2
D2	28.4	42.3	51.3	23.2	35.4	44.5	40.3	56.4	68.2
D4+D2	44.0	62.7	72.1		—			—	
D3+D2		—		40.1	56.5	65.8		—	
D3+D4		—			—		33.7	42.2	58.4

by training on two devices, and testing on a third, we were able to achieve identification accuracies above 80% with 6-second traces. This is likely because the classifier generalizes better when trained on multiple devices, which account for more variety within the traces.

B. *Training and Testing using WiFi and LTE*

We collected power traces while accessing websites over both WiFi and LTE and experimented with three training-testing configurations: (1) LTE training and LTE testing, (2) LTE training and WiFi testing, and (3) WiFi training and LTE testing. Our results (see Table III) show that accuracy obtained when training and testing on LTE is comparable to that of training and testing on WiFi (in Table I). Further, though there was reduction in accuracies with WiFi training-LTE testing and LTE training-WiFi testing, the overall results were significantly higher than the baseline (2%) at Rank 1, and at least 73.9% for 6 second traces.

C. *Aging of Training Traces*

Many of the webpages considered in this work contain content that changes over time. For example, the home page of `yahoo.com` shows recent news, and is therefore updated several times a day. For these webpages, a training dataset collected at a specific point in time might not be representative of the webpage's behavior at a later point, when the testing traces are collected. To determine the impact of aging on training data, we collected testing traces 32 and 70 days after training, with cache enabled and fully charged battery. The corresponding results are summarized in Table VI. For reference, all traces used to obtain the results reported in tables I and II were collected within a 48-hour timeframe.

TABLE VI
IDENTIFICATION ACCURACY (IN %) USING 70-DAY-OLD AND 32-DAY-OLD TRAINING DATASETS COLLECTED USING D1. ALL RESULTS WERE OBTAINED USING 125 FEATURES.

	Rank 1			Rank 5		
	2 s	4 s	6 s	2 s	4 s	6 s
32-day old, charged, w/ cache	8.0	11.9	13.0	24.7	27.8	32.3
70-day old, charged, w/ cache	3.6	3.7	2.2	13.8	20.2	24.3

TABLE VII
RANK 1 WEBPAGE IDENTIFICATION ACCURACY (IN %) FOR DOMESTIC (INDICATED AS "Dom.") AND FOREIGN ("For.") WEBSITES. TRACES WERE COLLECTED USING D1. ALL RESULTS WERE OBTAINED USING 125 FEATURES.

	2 s		4 s		6 s	
	Dom.	For.	Dom.	For.	Dom.	For.
Charged, no cache	98.1	92.9	99.7	97.1	99.7	98.2
Charged, w/ cache	99.4	97.1	99.4	100	100	100
Charging (30%), no cache	82.5	83.9	92.5	89.3	96.4	93.9
Charging (30%), w/ cache	87.5	87.9	94.2	95.0	96.1	96.4

When we trained the classifier on traces collected 32 days before testing, Rank 1 identification accuracy dropped below 13%. Even worse, the identification accuracy with training traces collected 70 days before testing was consistently below 4%. This suggests that, in order to achieve good identification accuracy, training traces should be updated frequently.

D. *Foreign vs. Domestic Websites*

We tested this variable because the distance between the client and the host serving a webpage is known to affect packets' delay and jitter [33]. More specifically, the farther the host serving the content, the more variable will be its measured bandwidth and delay. In turn, this variability affects page loading, and hence the corresponding power traces.

We determined whether a webpage was served from a host (e.g., a server, or a CDN) within the continental United States or outside the United States using the Whois databases from ARIN [34] and APNIC [35]. Our dataset is composed of 36 domestic websites, and 14 foreign websites.

The results of our analysis are summarized in Table VII. Our experiments show that the location of the host serving a webpage has a very small impact on identification accuracy. Enabling cache led to a larger improvement in webpage identification accuracy for foreign websites than for domestic websites.

E. *HTTPS vs. HTTP Websites*

We tested this variable because the use of encryption between the smartphone and the server can introduce noise in power traces. In particular, TLS requires additional communication rounds to exchange TLS session keys before a connection can be established. This can potentially increase the variability of power traces.

A total of 15 webpages in our dataset allow users to retrieve content using HTTPS, while the remaining 35 webpages only allow plain HTTP connections. Our results, reported in Table VIII, show that there is no significant difference in identification accuracy between the two types of websites. This indicates that the attack is as effective for identifying securely transmitted webpages as with webpages transmitted without encryption.

TABLE VIII
RANK 1 WEBPAGE IDENTIFICATION ACCURACY (IN %) FOR WEBPAGES RETRIEVED VIA HTTPS AND PLAIN HTTP. TRACES WERE COLLECTED USING D1. ALL RESULTS WERE OBTAINED USING 125 FEATURES.

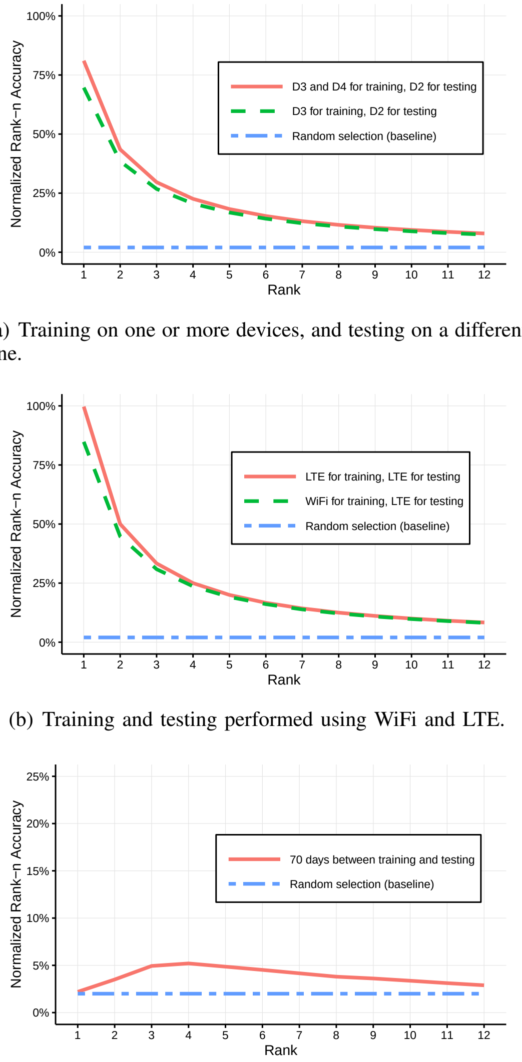
	2 s		4 s		6 s	
Scenario	HTTPS	HTTP	HTTPS	HTTP	HTTPS	HTTP
Charged, no cache	97.7	96.1	99.7	98.7	99.3	99.3
Charged, w/ cache	99.7	98.4	98.7	100	100	100
Charging (30%), no cache	81.7	83.4	90.0	92.3	95.0	96.0
Charging (30%), w/ cache	89.3	86.9	94.0	94.6	96.0	96.3

VII. NORMALIZED RANK-n ACCURACY

Figure 6 shows the Rank vs. Normalized Rank-n Accuracy plots for our classifier, obtained using frequency-domain features. All traces used in this evaluation were obtained with no cache, and with fully-charged battery.

Figures 6(a) and 6(b) show results of experiments where training and testing traces were from different devices, and different connectivity, respectively. The two figures show that increasing the rank leads to a decrease in normalized Rank-n accuracy. This means that the adversary has lower probability of correctly guessing a webpage as the rank increases. This is because Rank 1 accuracy is already substantially higher than the baseline. Therefore, increasing the rank provides almost no benefits towards correctly guessing the webpage. Although the normalized Rank-n accuracy approaches the baseline as the rank increases, figures 6(a) and 6(b) show that, even at Rank 12, our technique still outperforms the baseline.

The traces used in Figure 6(c) were collected on a single smartphone, with training and testing data collected 70 days apart. This figure shows that when outdated traces are used for training, Rank 1 classification no longer provides the highest normalized Rank-n accuracy. This is because of two reasons: (1) Rank 1 accuracy in this setting is close to the baseline; and (2) as the rank increases, the classifier outputs a better uncertainty set compared to random choice. As a result, the Rank 2 to Rank 4 accuracies increase more than the uncertainty due to the increased rank. For instance, when going from Rank 1 to Rank 4, the output of the classifier includes three additional webpages that are far more likely to be correct than three pages chosen at random. Therefore, in this particular case, the adversary is more likely to correctly guess the webpage by first increasing the rank from 1 to 4.



(a) Training on one or more devices, and testing on a different one.

(b) Training and testing performed using WiFi and LTE.

(c) Training and testing with traces collected 70 days apart.

Fig. 6. Rank vs. Normalized Rank-n Accuracy plots show the results of our experiments with automated dataset, using frequency-domain features and classifier voting. All experiments were performed with a fully-charged battery and no cache. The plots in the top figure were obtained using 15 features, and the plots in the middle and bottom figures were obtained using 125 features, because these settings led to the best results.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we demonstrated that it is possible to accurately infer browsing activity on a smartphone using USB power consumption measurements. Our work is the first to study this side-channel attack on smartphones, and to analyze a multitude of factors that affect the traces that are collected during the attack, such as: battery charging level, user interaction with the touchscreen, trace length, time between collection of training and testing traces, WiFi and LTE connectivity, training and testing device mismatch, and website characteristics such as type of connection (HTTP or HTTPS) and location of the host serving the webpage relative to the smartphone.

We performed extensive experiments to validate our approach. Our results show that the attack successfully identifies webpages loaded using the standard Android mobile browser at least 91.7% of the times within four seconds for the automated dataset. For our user-actuated dataset, identification accuracy was between 54.2% and 88.4% with six seconds of power consumption data. Factors such as the availability of cache, the use of secure connections (HTTPS) and the location of the websites had a small effect on identification accuracies. Other factors, such as mismatch between training and testing traces due to the use of different devices and the type of wireless connectivity negatively impacted accuracies. However, training on multiple devices allowed us to achieve accuracies substantially higher than the baseline for Rank 1.

Overall, our results show that the attack is highly effective, because webpage loading generates power signatures that are: (1) **distinctive**: different webpages generate different power traces due to factors such as the amount of data (text, images, and videos) being retrieved, the number of TCP connections required to retrieve all webpage components, and the computational cost of the scripts running within the webpage; and (2) **consistent**: each time a particular page is loaded, it generates a power trace that is similar to its previous power traces.

Though our work focuses on webpage identification, we believe that the same side-channel can also be used to detect other user activities, such as which application is currently being used, and the timing of touch events (including typing information). We consider this work as the first step towards a deeper understanding of the extent of the information leaked through power analysis of the USB port of a smartphone. Clearly, there are a multitude of factors that we did not address in this paper, such as number of applications installed on the smartphone, background processes, network congestion, WiFi/LTE signal strength, and specific user interaction. We leave the analysis of these factors to future work.

References

[1] "Cell phone battery statistics across major us cities," https://veloxity.us/phone-battery-statistics/, accessed: 2015-09-07.

[2] "Power up: A guide to US airport charging stations – Cheapflights," http://www.cheapflights.com/news/power-up-a-guide-to-us-airport-charging-stations/#ewr, accessed: 2016-04-04.

[3] "Briant Park Blog: Solar-powered charging stations land in Bryant Park," http://blog.bryantpark.org/2014/07/solar-powered-charging-stations-land-in.html, accessed: 2016-04-04.

[4] "Solar-powered phone charging stations launch in union square," https://www.dnainfo.com/new-york/20130620/union-square/solar-powered-phone-charging-stations-launch-union-square, accessed: 2016-04-04.

[5] "Chargeport hotel charging station," http://www.teleadapt.com/hospitality-products/power-charging/chargeport, accessed: 2016-04-04.

[6] "Behind the charge: A big challenge for hospitals," http://www.mkelements.com/blog/behind-charge-big-challenge-hospitals, accessed: 2016-04-04.

[7] "Beware of juice-jacking," http://krebsonsecurity.com/2011/08/beware-of-juice-jacking/, accessed: 2016-04-04.

[8] "SyncStop: Charge your mobile phone safely," http://syncstop.com, accessed: 2016-04-04.

[9] "All about skimmers – krebs on security," http://krebsonsecurity.com/all-about-skimmers/, accessed: 2016-03-02.

[10] S. Clark, H. Mustafa, B. Ransford, J. Sorber, K. Fu, and W. Xu, "Current events: Identifying webpages by tapping the electrical outlet," in *Computer Security - ESORICS 2013*, ser. Lecture Notes in Computer Science, J. Crampton, S. Jajodia, and K. Mayes, Eds. Springer Berlin Heidelberg, 2013, vol. 8134, pp. 700–717. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40203-6_39

[11] "Alexa global website traffic ranking," http://www.alexa.com/topsites/global, accessed: 2015-09-07.

[12] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, ser. USENIXATC'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 21–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855840.1855861

[13] "What you think you know about the web is wrong," http://time.com/12933/what-you-think-you-know-about-the-web-is-wrong/, accessed: 2016-04-04.

[14] D. Genkin, I. Pipman, and E. Tromer, "Get your hands off my laptop: Physical side-channel key-extraction attacks on pcs," in *Cryptographic Hardware and Embedded Systems - CHES 2014*, ser. Lecture Notes in Computer Science, L. Batina and M. Robshaw, Eds. Springer Berlin Heidelberg, 2014, vol. 8731, pp. 242–260. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44709-3_14

[15] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly, "Powerspy: Location tracking using mobile device power analysis," in *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 785–800.

[16] L. Yan, Y. Guo, X. Chen, and H. Mei, "A study on power side channels on mobile devices," in *The Seventh Asia-Pacific Symposium on Internetware*, ser. Internetware'15, 2015.

[17] A. Hintz, "Fingerprinting websites using traffic analysis," in *Proceedings of the 2Nd International Conference on Privacy Enhancing Technologies*, ser. PET'02. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 171–178. [Online]. Available: http://dl.acm.org/citation.cfm?id=1765299.1765312

[18] L. Lu, E.-C. Chang, and M. C. Chan, "Website fingerprinting and identification using ordered feature sequences," in *Proceedings of the 15th European Conference on Research in Computer Security*, ser. ESORICS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 199–214. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888881.1888898

[19] S. Chen, R. Wang, X. Wang, and K. Zhang, "Side-channel leaks in web applications: A reality today, a challenge tomorrow," in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, ser. SP '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 191–206. [Online]. Available: http://dx.doi.org/10.1109/SP.2010.20

[20] "Badusb - on accessories that turn evil," https://pacsec.jp/psj14/PSJ2014_Karsten_Nohl_141112.BadUSB-Pacsec.KN01.pdf, accessed: 2015-09-07.

[21] "Usb attacks need physical access right? not any more..." https://www.blackhat.com/docs/asia-14/materials/Davis/Asia-14-Davis-USB-Attacks-Need-Physical-Access-Right-Not-Any-More.pdf, accessed: 2015-09-07.

[22] "Max77818: Dual input, power path, 3A switching mode charger with FG," http://datasheets.maximintegrated.com/en/ds/MAX77818.pdf, accessed: 2015-09-07.

[23] "Charging lithium-ion batteries: Not all charging systems are created equal," https://www.microchip.com/stellent/groups/designcenter_sg/documents/market_communication/en028061.pdf, accessed: 2015-09-07.

[24] "Triple output power supply Agilent model E3630A," http://cp.literature.agilent.com/litweb/pdf/5959-5329.pdf, accessed: 2015-09-07.

[25] "NI USB-6211 DAQ," http://sine.ni.com/nips/cds/view/p/lang/en/nid/203224, accessed: 2015-09-07.

[26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/A%3A1010933404324

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: http://dx.doi.org/10.1023/A:1022627411411

[28] J. C. Platt, "Advances in kernel methods," in *Advances in Kernel Methods*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208. [Online]. Available: http://dl.acm.org/citation.cfm?id=299094.299105

[29] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '00. New York, NY, USA: ACM, 2000, pp. 285–289. [Online]. Available: http://doi.acm.org/10.1145/347090.347153

[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[31] P. Duhamel and M. Vetterli, "Fast fourier transforms: A tutorial review and a state of the art," *Signal Process.*, vol. 19, no. 4, pp. 259–299, Apr. 1990. [Online]. Available: http://dx.doi.org/10.1016/0165-1684(90)90158-U

[32] "Average mobile page load time for a fortune 100 company is about 5 seconds." http://cl.ly/2v0g2B1c0R00, accessed: 2016-02-09.

[33] S. Kaune, K. Pussep, C. Leng, A. Kovacevic, G. Tyson, and R. Steinmetz, "Modelling the internet delay space based on geographical locations," in *Parallel, Distributed and Network-based Processing, 2009 17th Euromicro International Conference on*. IEEE, 2009, pp. 301–310.

[34] "Arin (american registry for internet numbers) whois tool," https://whois.arin.net, accessed: 2015-12-12.

[35] "Apnic (asia pacific network information centre) whois tool," https://wq.apnic.net/apnic-bin/whois.pl, accessed: 2015-12-12.

Appendix

Table IX reports which websites are domestic, and which ones are accessible via HTTPS. We summarize all other dataset characteristics in Table X.

TABLE IX

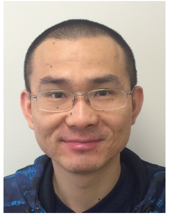
ALEXA TOP 50 NON-ADULT WEBSITES, AS OF JUNE 2015. THE TABLE REPORTS WHETHER EACH WEBSITE IS FOREIGN OR DOMESTIC, AND WEATHER AND THE CONNECTION TYPE IS HTTP OR HTTPS.

Webpage	Domestic	HTTPS	Webpage	Domestic	HTTPS	Webpage	Domestic	HTTPS	Webpage	Domestic	HTTPS
google.com	✓	✓	sina.cn			vimeo.com	✓	✓	fc2.com	✓	
facebook.com	✓	✓	weibo.com			imgur.com	✓		snapdeal.com		
youtube.com	✓		tmall.com			wordpress.com	✓	✓	ask.com	✓	
yahoo.com	✓	✓	ok.ru			cnet.com	✓		alibaba.com	✓	
baidu.com			ebay.com	✓		msn.com	✓		espncricinfo.com	✓	
wikipedia.com	✓	✓	about.com	✓		pinterest.com	✓	✓	360.cn		
amazon.com	✓		hao123.com			yandex.ru			stackoverflow.com	✓	
twitter.com	✓	✓	reddit.com	✓		paypal.com	✓	✓	netflix.com	✓	✓
taobao.com			bing.com	✓		microsoft.com	✓		163.com		
live.com	✓		etsy.com	✓	✓	vk.com			slideshare.net	✓	
qq.com			instagram.com	✓	✓	aliexpress.com	✓		craigslist.org	✓	
bankofamerica.com	✓	✓	sohu.com			apple.com	✓				
linkedin.com	✓	✓	tumblr.com	✓	✓	imdb.com	✓				

TABLE X

DETAILS ON THE DATASET USED IN THIS PAPER. THE DEVICES USED FOR DATA COLLECTION ARE FOUR SAMSUNG GALAXY S4 (LABELLED D1, D2, D3, AND D4 IN THE TABLE), AND ONE SAMSUNG GALAXY S6 (D5).

Location	Device(s)	Network	Collection Method	Browser Cache	Battery Level	Collection Time	Usage in our Paper
Old Westbury (NY)	Galaxy S4 (D1)	WiFi	Automated	Disabled	100%	May 2015	Table VI
						June 2015	
						July 2015	Table I, VI, VII, VIII
					30%	July 2015	Table I, VII, VIII
				Enabled	100%	Aug. 2015	Table I, VII, VIII
					30%		
			User-actuated	Disabled	100%	July 2015	Table II
					30%		
				Enabled	100%	Aug. 2015	
					30%		
Williamsburg (VA)	Galaxy S4 (D2)	WiFi	Automated	Disabled	100%	Aug. 2016	Table III
		LTE					
	Galaxy S4 (D2, D3, and D4)	WiFi	Automated	Disabled	100%	Aug. 2016	Table IV
					30%	Sept. 2016	Table V
	Galaxy S6 (D5)	WiFi	Automated	Disabled	100%	Aug. 2016	Table I
					30%	Sept. 2016	



Qing Yang is pursuing a Ph.D. degree in the Computer Science Department at the College of William & Mary. He received his B.S from Civil Aviation University of China in 2003 and M.S. from Chinese Academy of Sciences in 2007. His research interests are smartphone security, energy efficiency, and ubiquitous computing.

Dr. Paolo Gasti is an assistant professor of Computer Science at the New York Institute of Technology (NYIT), School of Engineering and Computing Sciences. His research focuses on behavioral biometrics, privacy-preserving biometric authentication and identification, secure multi-party protocols, and network security. Before joining NYIT, he worked as a research scholar at University of California, Irvine. His research has been sponsored by NSF and DARPA. He received his B.S., M.S., and Ph.D. degrees from University of Genoa, Italy. He is a Fulbright scholar, and member of the IEEE.

Dr. Gang Zhou is an Associate Professor, and Graduate Director, in the Computer Science Department at the College of William and Mary. He received his Ph.D. degree from the University of Virginia in 2007. He has published over 80 academic papers in the areas of sensors & ubiquitous computing, mobile computing, wireless networking, internet of things, and smart healthcare. The total citations of his papers are 5988 per Google Scholar, among which five of them have been transferred into patents. He is currently serving in the Journal Editorial Board of IEEE Internet of Things, Elsevier Computer Networks, and Elsevier Smart Health. He received an award for his outstanding service to the IEEE Instrumentation and Measurement Society in 2008. He is a Senior Member of ACM and a Senior Member of IEEE.

Dr. Aydin Farajidavar received the B.Sc. and M.Sc. degrees in biomedical engineering, in 2004 and 2007, respectively. He received the Ph.D. degree in biomedical engineering from the joint program of the University of Texas at Arlington and the University of Texas Southwestern Medical Center, Dallas in 2011. He is currently an Assistant Professor of Electrical and Computer Engineering and the Director of Integrated Medical Systems Laboratory at New York Institute of Technology (NYIT). Before joining NYIT, he was a Post-doctoral Fellow in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. His research experience and interests cover a broad range, from Medical Cyber Physical Systems (implantable, wearable and assistive technology) to modeling biological systems.

Dr. Kiran S. Balagani is an assistant professor of Computer Science at the New York Institute of Technology. His research interests are in cyber-behavioral anomaly detection (e.g., unauthorized user-access behaviors), behavioral biometrics, and privacy-preserving biometrics. Balagani's work has appeared in several peer-reviewed journals, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Information Forensics and Security, and the IEEE Transactions on Knowledge and Data Engineering. He holds three U.S. patents in network-centric attack detection. Balagani received the Ph.D. degree from Louisiana Tech University.