USB Side-channel Attack on Tor

Qing Yang[a,*], Paolo Gasti[b], Kiran Balagani[b], Yantao Li[c], Gang Zhou[a]

[a]*Department of Computer Science, College of William and Mary, Williamsburg, VA, USA*
[b]*School of Engineering and Computing Sciences, New York Institute of Technology, New York, NY, USA*
[c] *College of Computer & Information Science, Southwest University, Chong Qing, China*

Abstract

Tor is used to communicate anonymously by millions of daily users, which rely on it for their privacy, security, and often safety. In this paper we present a new attack on Tor that allows a malicious USB charging device (e.g., a public USB charging station) to identify which website is being visited by a smartphone user via Tor, thus breaking Tor's primary use case. Our attack solely depends on power measurements performed while the user is charging her smartphone, and it does not require the adversary to observe any network traffic or to transfer data through the smartphone's USB port. We evaluated the attack by training a machine learning model on power traces from 50 regular webpages and 50 Tor hidden services. We considered realistic constraints such as different network types (LTE and WiFi), Tor circuit types, and battery charging levels. In our experiments, we were able to correctly identify webpages visited using the official mobile Tor browser with accuracies up to 85.7% when the battery was fully charged, and up to 46% when the battery level was between 30% and 50%. Both results are substantially higher than the 1% baseline of random guessing. Surprisingly, our results show that hidden services can be identified with higher accuracies than regular webpages (e.g., 84.3% vs. 68.7% over LTE).

Keywords: Tor, side-channel attacks, de-anonymization, privacy

1. Introduction

Tor is an application-level low-latency network that enables anonymous communication between a client and arbitrary Internet servers. Tor uses a collection of onion routers [1], hosted by a number of volunteers, to unlink the identity and the geographical location of the client from the server, and to conceal the identity of the server to any adversary that can observe the client's network activity (e.g., from the client's Internet service provider). Users rely on Tor to conceal their activities from hackers, governments, employers, and ISPs, since

*Corresponding author
Email address: qyang@email.wm.edu (Qing Yang)

those might abuse, misuse, or accidentally leak sensitive information. Further, Tor is frequently used to protect the safety and security of political activists, to overcome communication restrictions, and to evade censorship.

Given prior work on the security of widely-available public USB charging stations [2], in this work we investigate whether a malicious charging station can infer which websites are accessed by the Tor user while she charges her smartphones. The ability to determine which website is being accessed through Tor using power consumption information, rather than by observing network traffic, makes our technique a novel, hitherto unexplored, and potentially devastating attack vector. We consider this type of attack significant, because: (1) political dissidents and human right activists rely on Tor to dissociate speech from their identities [3]. Disclosing which website they have visited is sufficient, in many circumstances, to endanger their freedom and life; (2) there is often a correlation between which website a user visits, and some of her sensitive information, including health (e.g., if the user visits a forum for cancer survivors, or a website providing advices to HIV patients), political affiliation (e.g., when visiting a party's website), and sexual orientation (e.g., when visiting an LGBT forum); and (3) the core purpose of Tor, as stated by its authors, is "to frustrate attackers from linking communication partners" [1]. Therefore, disclosing which website is being visited by the user defeats Tor's purpose.

Contributions. In this paper we introduce a new attack on Tor. This attack enables a malicious charging station to identify which website is being visited via Tor by smartphone users. Our attack relies on power measurements performed while the user is charging her smartphone, and allows the adversary to determine which websites are visited.

In our evaluation, we were able to correctly identify websites accessed via the Orbot/Orfox Tor browser [4] with accuracies between 34.5% to 85.7% under realistic constraints, such as different network types (LTE and WiFi) and battery levels (30% to 50%, and 100%). In both cases, our accuracies were substantially higher than the 1% baseline accuracy obtained using random guessing. Further, our attack was successful in identifying not only regular webpages, but also pages served by Tor hidden services, thereby increasing the scope of the threats identified in this work. We consider this a serious attack on Tor because: (1) public charging stations are becoming widely available, making the attack scenario in this paper very realistic and widespread, and (2) the level of privilege required to implement this attack is minimal, as it needs no access to (or manipulation of) network traffic, no malicious servers, and no exploitation of bugs in the Tor software. Because the security of Tor is critical to guarantee the safety and freedom of a large number of users around the world, any low-privilege attack that reliably and accurately infers user activity should be considered very seriously.

Energy and Loading Time Impact of Tor. Accessing web content via Tor has significant effects on webpage loading. When users browse webpages using Tor, all requests and the corresponding responses are forwarded by three Tor relays

in the Tor circuit. Each relay encrypts and decrypts all data in transit. Since the relays are geographically distributed, each packet can potentially travel a long distance before reaching its destination, thus introducing large and variable network delays. Further, because Tor circuits are composed of randomly selected relays, each circuit can introduce different delays, thus adding further uncertainty and inconsistency when loading the same webpage. Finally, the construction of Tor circuits consumes additional energy, thus adding background noises to the power traces for webpage loading.

To illustrate the effects of Tor on webpage loading, we measured the loading time of six webpages on a Samsung Galaxy S6 with and without Tor. We loaded each webpage 5 times. The results are shown in Figure 1.

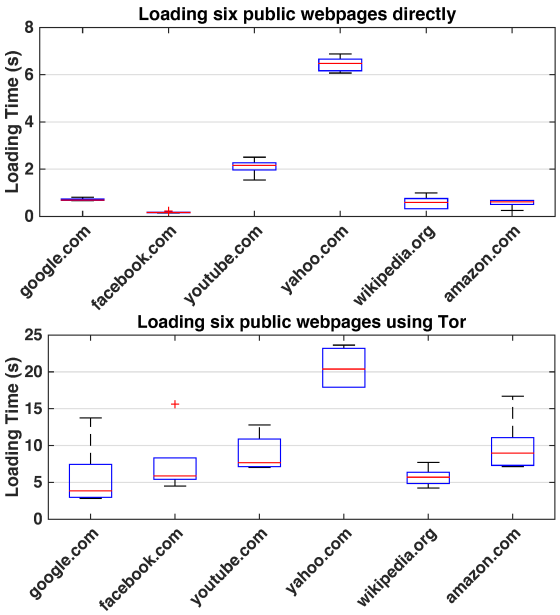

Figure 1: Loading time for six public webpages without Tor (upper plot) vs. using Tor (lower plot).

Using Tor not only increased the loading time (from 2 seconds to 10 seconds, on average), it also introduced a larger variation within the loading time. The average relative standard deviation of loading time was 21.98% without Tor, and 40.54% with Tor. The effects of loading webpages with Tor are further reflected in the power traces. Figure 2 shows the power traces collected while loading the homepage of `google.com`. We compared the power trace when loading the same webpage directly and using Tor (on the same smartphone and mobile browser). When loading `google.com` without Tor, most of the energy is consumed within the first second. When using Tor, the energy consumption is spread across a longer period (the first 7 seconds). The appearance of such random power patterns leads us to question whether it is possible to identify webpages based

on power signatures when using Tor.

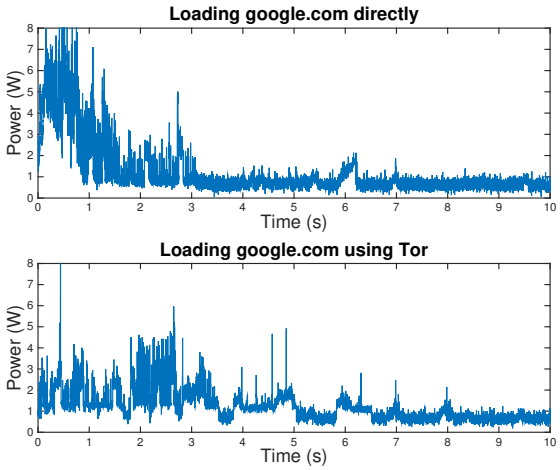

Figure 2: Power traces collected during the first 10 seconds of loading google.com without Tor (upper plot) vs. using Tor (lower plot). The x axis shows time from the beginning of the webpage loading, and the y axis shows the power drawn from the USB port.

Organization. The remainder of this paper is organized as follows. We introduce our experiment setup and collected datasets in Section 2. Section 3 details our webpage identification technique. Evaluation of our technique is presented in Section 4. We review the related work in Section 5. Our conclusion and future work are in Section 6.

2. Data collection

In this section, we first introduce the hardware, software, and network setup for the collection of power traces. We then explain how webpages and Tor circuits are selected. Finally, we present details on all datasets used in this paper.

2.1. Experiment Setup

Power supply. We powered the smartphone using a Rigol DP832 power supply [5], set to 5.5 V when the smartphone battery was fully charged, and to 9 V when the smartphone battery was charging from 30%. The latter setting is supported by the Samsung Galaxy S6 and other smartphones compatible with Qualcomm Quick Charge [6], and it resulted in a wider power consumption dynamic range.

Device connection. As per USB charging specification [7], we connected the data pins (D+ and D−) of the USB cable using a 200 Ω resistor to allow for charging currents above 500 mA. To measure the instantaneous smartphone power consumption from the USB port, we inserted a 0.1 Ω shunt resistor on

the GND wire of the USB cable, and measured the voltage drop across the resistor using a National Instruments USB-6211 DAQ [8]. The DAQ was set to use a sampling rate of 200 kHz. We connected the DAQ's output port to a Thinkpad T440P laptop, which was used to store the power traces using the LabView software.

Tor software setup. To collect power traces, we used the official Tor apps on Android, Orbot [9] and Orfox [4], on two Samsung Galaxy S6 smartphones, denoted as phone A and phone B in the rest of this paper. Orbot implements a local proxy that provides access to the Tor network. Orfox is a web browser based on the smartphone version of Firefox. It enhances Firefox by including features that improve user privacy, such as HTTPS Everywhere [10]. Further, it disables the execution of JavaScript code by default.

We connected Orfox to Tor using the Orbot instance on the smartphone. To collect data reliably, we modified Orfox by disabling the Android flag `FLAG_SECURE` to enable screenshot once a web page was loaded. This was used to manually verify that all pages were loaded successfully. We also implemented a Tor option that enables manual selection of the second relay, so as to create a "fixed" Tor circuit.

Power trace collection. To load each webpage automatically, we developed an Android background service that cycled through our webpages (listed in tables 1 and 2). After loading each webpage, the service paused for 12 seconds, and then logged which URL that was loaded, together with the corresponding timestamp, to a file on the smartphone.

We synchronized each power trace with the corresponding URL using the following process. Before collecting each dataset (see Table 3), we used the same NTP server to synchronize the clocks of the smartphone and of the laptop used for recording the power traces. We then used the timestamps in the smartphone log file and in the power traces to align the first data point associated with each URL.

Table 1: 50 webpages selected from Alexa top non-adult websites (as of Dec 2016).

google.com	amazon.com	ebay.com	microsoft.com	fc2.com
facebook.com	twitter.com	wordpress.com	vk.com	snapdeal.com
youtube.com	sina.cn	msn.com	apple.com	ask.com
yahoo.com	weibo.cn	pinterest.com	imdb.com	stackoverflow.com
wikipedia.org	ok.ru	paypal.com	office.com	netflix.com
dailymail.co.uk	stackexchange.com	booking.com	indeed.com	salesforce.com
nytimes.com	daum.net	dropbox.com	whatsapp.com	nicovideo.jp
thepiratebay.org	wikia.com	pixnet.net	coccoc.com	adf.ly
espn.com	bbc.com	sogou.com	blogger.com	mail.ru
github.com	cnn.com	naver.com	rakuten.co.jp	adobe.com

Table 2: 50 random selected hidden services (all with *.onion* as domain name suffix).

rougmnvswfsmd4dq	yuxv6qujajqvmypv	nql7pv7k32nnqor2	s5q54hfww56ov2xc	sblib3fk2gryb46d
ityukvsoqjgzcimm	kxojy6ygju4h6lwn	cashis7ra6cy5vye	3g2upl4pq6kufc4m	fdwocbsnity6vzwd
65px7xq64qrib2fx	fzqnrlcvhkgbdwx5	clockwise3rldkgu	libertygb2nyeyay	xmh57jrzrnw6insl
hss3uro2hsxfogfq	kpynyvym6xqi7wz2	fbcy5ylyoeqzqzcr	undergunbgzlc2ey	o6klk2vxlpunyqt6
vu2wohoog2bytxgr	xfnwyig7olypdq5r	54ogum7gwxhtgiya	slwc4j5wkn3yyo5j	c3jemx2ube5v5zpg
answerstedhctbek	tfwdi3izigxllure	gjobqjj7wyczbqie	ll6lardicrvrljvq	aaaajqiyzj34rhjm
drystagepmi5msdm	greendrgfjz7ks5f	4yjes6zfucnh7vcj	abbujjh5vqtq77wg	b34xhb2kjf3nbuyk
usjudr3c6ez6tesi	76qugh5bey5gum7l	djypjjvw532evfw3	grams7enufi7jmdl	w363zoq3ylux5rf5
nare7pqnmnojs2pg	kbvbh4kdddiha2ht	qputrq3ejx42btla	zqktlwi4fecvo6ri	ccxdnvotswsk2c3f
flibustahezeous3	74ypjqjwf6oejmax	tetatl6umgbmtv27	jmkxdr4djc3cpsei	hss3uro2hsxfogfq

Networks. We loaded all webpages using the WiFi network on the campus of The College of William & Mary, and via the T-Mobile LTE network in Williamsburg.

2.2. Datasets

To collect data, we used two types of Tor circuits—fixed, and automatic—to retrieve regular webpages, and to access Tor hidden services [11]. Details follow.

Collection of Data from Tor Hidden Services. In contrast with servers on the public Internet, Tor hidden services are accessible only using the Tor network. Hidden service providers reside on Tor relays or Tor clients, and offer various services including web hosting, instant messaging, and SSH, while hiding the hidden service IP addresses. Each Tor hidden service hides behind several "introduction" relays in the Tor network. When visiting a hidden service, the Tor client first downloads the service's public descriptor (identified by a unique 16-character name followed by ".onion"). Then, it creates a Tor circuit to a randomly selected "rendezvous" relay, and it sends the rendezvous relay's address to the hidden service through one introduction relay. The hidden service creates a Tor circuit to the rendezvous relay, and the client uses the "rendezvous" relay to exchange encrypted messages with the hidden service. In the rest of this paper, we refer to the webpages hosted on public Internet servers as "public webpage", and to web content hosted on hidden services as "hidden service".

We collected power traces while loading selected public webpages and hidden services. Tables 1 and 2 list all websites used in our experiments. For public webpages, we selected the home pages of the 50 most popular non-adult websites accessible via Tor, based on the Alexa ranking. We excluded public webpages that do not display content without JavaScript, because Orfox disables JavaScript by default. For hidden services, we randomly selected 50 websites from The Hidden Wiki [12] that were consistently available during the experiments. Because 100 webpages represent only a small portion of the Web, we consider this work as a proof of concept. However, even with this restriction, our results conclusively show that substantial information is leaked when the adversary is able to monitor power consumption during page load.

Selection of Tor Circuits. When a Tor client builds a circuit, it first selects three relays from a public directory. The client then connects to the first relay ("entry"), and it uses this relay to extend the circuit to the second relay ("middle"). The client finally uses the first two relays to extend the circuit to the last relay ("exit"). As it constructs the circuit, the client shares a unique symmetric key with each relay.

We performed our experiments using two types of Tor circuits: "automatic", and "fixed". For *automatic* circuits, we allowed Orbot to select a new circuit for each webpage loading using the default path selection protocol [13]. By default, the entry relay is selected among a small group of long-term entry servers (*guard* nodes), and it does not change for a relatively long time. However, Orbot settings allow the user to disable using entry guard by setting "UseEntryGuards" to 0. We used this option in our experiments to model the inability of the adversary to use the same Entry Guard as the user.

The circuit used to load a specific webpage changes every 10 minutes by default. Because in our data collection the time between collection of subsequent traces from the same webpage is larger than 30 minutes, power traces from the same website were collected using different circuits.

With the fixed circuits, we manually chose all three Tor nodes in the circuit and used them to load all webpages. In our dataset, we denoted the circuit composed of `anonymiton` (in Germany, entry node), `torfa` (in Hungary, middle node), and `Hermes` (in France, exit node) as "Cir-1". We denoted the following circuit as "Cir-2": `inky` (in Switzerland, entry), `cry` (in Netherlands, middle), and `hessel1` (in Romania, exit). Although in practice Orbot (or any modern Tor implementation) does not use a fixed circuit for loading multiple webpages, we used this type of circuits to evaluate the scenario where training and testing data were collected under conditions that were as consistent as possible. This allowed us to quantify the loss of accuracy due to noise induced by the use of different Tor circuits in training and testing.

The datasets used in our experiments are listed in Table 3. For each configuration, we loaded all 100 URLs once, and then repeated this process 40 times. If a webpage did not load successfully, we replaced that trace with a new one from the same URL, collected at the end of the data collection session. As a result, each dataset is composed of 40 power traces for each of the 100 URLs. The duration of each trace is 10 seconds, because this allowed Orfox to load almost all webpages completely.

Table 3: Configurations used to collect power trace datasets. We collected datasets #7 and #8 under the same settings as #1 and #2. However, to minimize the effect of time difference on the identification accuracy when comparing two different phones, #7 and #8 were collected within 12 hours of each other's time, while #1 and #2 were collected two days apart.

Dataset	Phone	Circuit	Network	Battery Level
1	A	Automatic	WiFi	100%
2	B	Automatic	WiFi	100%
3	A	Automatic	LTE	100%
4	B	Automatic	WiFi	30% to 50%
5	A	Fixed #1	WiFi	100%
6	B	Fixed #2	WiFi	100%
7	A	Automatic	WiFi	100%
8	B	Automatic	WiFi	100%

Effects of User Interaction on Power Traces. Prior work [2] has shown that user interactions, such as taps and swipes on the smartphone's touchscreen, affect power traces, and therefore the success rate of the attack. For example, in [2] user interactions decreased the attack's success rate by 16.7%-25.7%. We expect that user interactions have similar effects on traces collected using Tor.

3. Feature selection and classification

To identify the public webpages and hidden services loaded on the smartphone, we first extracted time- and frequency-domain features from the power traces, and then trained a Random Forest classifier on the resulting feature vectors. We used the trained classifier to predict the webpages on new power traces.

3.1. Feature selection

We experimented with time-domain features, such as mean, RMS, and correlation coefficient, and frequency-domain features based on simple FFT, cepstrum analysis [14], and spectrogram analysis. Our experiments demonstrated that features based on spectrogram analysis led to higher accuracies compared to other techniques. For instance, spectrogram features led to an increase in accuracy of up to 16% compared to FFT features. Figure 3 illustrates that different webpages have distinctive frequency spectrum patterns, where the magnitude values in most frequencies are significantly different among these webpages. We further divided each power trace into several overlapping 0.5-second segments. We calculated the spectrogram of each segment (using window length of 1000 samples, and 50% overlap between windows). The spectrogram results contain the magnitudes of frequencies in the range of 0 Hz $\sim$ 100 kHz. We divided this range into 125 equal-size bins to reduce the effects of noise of individual frequencies, and calculated the average magnitude of all the frequencies in each bin as its corresponding feature, thus transforming each power trace segment into a feature vector of 125 elements, as shown in Figure 4. We can observe that the feature vectors between different webpages are significantly more obvious than between traces for the same webpage.

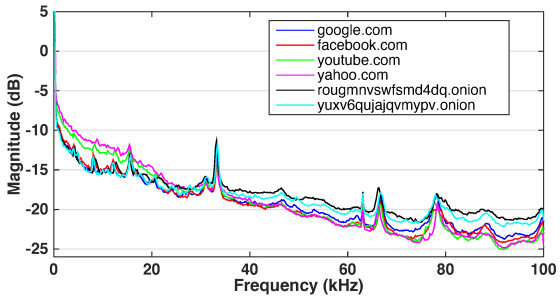

Figure 3: Spectrogram analysis on power traces sampled while loading six different websites.

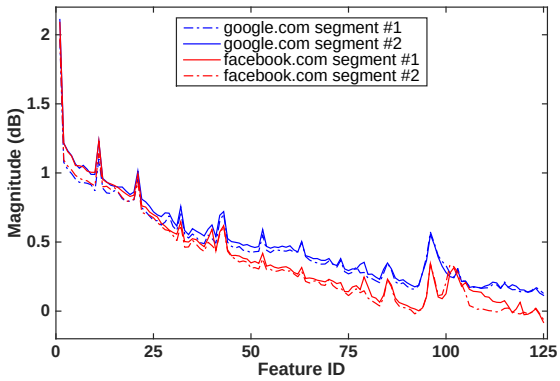

Figure 4: Features extracted from segments of power traces based on spectrogram analysis.

3.2. Classification

The classification problem can be abstracted as follows. We use $X = (x_1, x_2, \ldots, x_p)$ to denote a feature vector with p features. Variable Y represents possible classes $1, 2, \ldots, K$. Given a training dataset S with N observations (X_i, Y_i), we first use S to train a classifier $\hat{C}(X) \in \{1, 2, \ldots, K\}$, and then use $\hat{C}(X)$ to predict the classes of testing feature vectors.

We used Random Forests [15] for classifier training. A random forest consists of a set of decision trees. We use B to denote the number of decision trees to build for a random forest. There are three steps to train the random forest: (1) from dataset S, each time we randomly draw N observations with replacement to create a *bootstrap* dataset S_b for $b = 1, 2, \ldots, B$, (2) from each S_b, we train a decision tree $C(S_b, X)$, and (3) the random forest classifier is the ensemble of all $C(S_b, X)$, and it uses majority vote to make prediction on testing feature vectors.

In the following, we give details of the above step (2). Each decision tree is built from the root node. At each node, we randomly select a subset of all the p features, denoted by $F = \{x_1^*, x_2^*, \ldots, x_m^*\}, m = \lfloor log_2 p + 1 \rfloor$. For each feature $x_i^*, i = 1, 2, \ldots, m$, we use G_i to denote the set of all possible x_i^* values in the

dataset. We try each possible test $x_i^* < g, g \in G_i$ to split the current node, and choose the test that generates the largest "information gain". To calculate the information gain, at each node, assuming P_i is the occurrence probability of class $i, i = 1, 2, \ldots, K$, we first calculate the *Shannon entropy* as:

$$H = -\sum_{i=1}^{K} P_i \log_2 P_i \tag{1}$$

Assume that the entropy at current node is H. After a splitting, there are L percent of observations in the left child and R percent of observations in the right child. We use H_L and H_R to denote the entropy at the left and right child, respectively. Then the average entropy after splitting is:

$$H_{After} = H_L \times L + H_R \times R \tag{2}$$

The *information gain* of a splitting is defined as $(H - H_{After})$. At each node, our goal is finding the splitting to:

$$maximize(H - H_{After}) \tag{3}$$

The above process recurses on each child until a stopping condition is satisfied, such as all observations in the node belong to the same class, or the maximum tree depth has been reached.

After all decision trees are trained, the random forest classifier is defined by:

$$\hat{C}(X) = majority\_vote\{C(S_b, X)\}, b = 1, 2, \ldots, B \tag{4}$$

We used the WEKA [16] implementation of Random Forests. For each of our experiment scenario, we used 20 power traces per webpage to train the classifier, and the other 20 power traces per webpage for testing. We trained the classifier using segments of all training traces. To identify a testing power trace, we first classified all the segments of this trace, and then used majority voting of these segments to determine the class of this trace.

4. Performance Evaluation

We first present the identification accuracies of our technique for our basic configuration (i.e., using WiFi and fully-charged battery). We then discuss how different variables, including using different phones for training and testing, network types, and battery charging levels, affect identification accuracy.

4.1. Identification Accuracy for Basic Configuration

We list the datasets corresponding to our basic configuration as #1 and #2 in Table 3. We evaluated the following three cases: (1) using all traces for both public webpages and hidden services; (2) using traces from public webpages for training and testing; (3) using traces from hidden services for training and testing. In each case, we used half traces for training, and the other half for

testing. The results are shown in Table 4, which includes Rank-1 and Rank-5 identification accuracies. With Rank-1, a trace is classified correctly if its label is the output of the classifier with the highest confidence. With Rank-5, traces are considered correctly classified if their label appear among the 5 labels identified by the classifier with the highest confidence.

Table 4: Webpage identification accuracy using WiFi and 100%-charged battery (basic configuration)

Phone	All webpages		Public webpage only		Hidden service only	
	Rank-1	Rank-5	Rank-1	Rank-5	Rank-1	Rank-5
A	79.05 %	88.7%	76.3%	88.1%	87.3%	93.2%
B	85.7%	92.6%	82.2%	93.2%	89.2%	94.9%

We were able to identify hidden services with higher accuracy (87.3%) than public webpages (76.3%), as shown in Figure 6. There are two possible reasons for the accuracy difference between public webpages and hidden services. First, we measured the loading time of six hidden services (shown in Figure 5). We observed that, compared to public webpages (see Figure 1), the loading time of hidden service is more consistent (19.82% relative standard deviation, on average, compared to 40.54% for public webpages) and has smaller range (from 3.25 s to 10.63 s, compared to 2.81 s to 23.63 s for public webpages). Possible reasons for these differences include: (1) hidden services are mostly simple static webpages; and (2) their contents rarely change over a long time. In contrast, public webpages usually contain large-size elements. Since public webpages need longer loading time than hidden services, they have a higher chance of being influenced by the instability of Tor circuits. As a result, there are more inconsistency and noise in the training and testing power traces of public webpages.

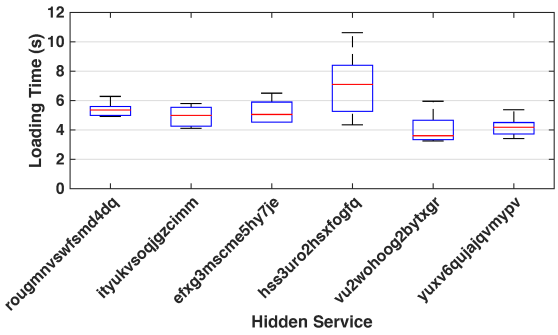

Figure 5: Loading time for six hidden services using Tor.

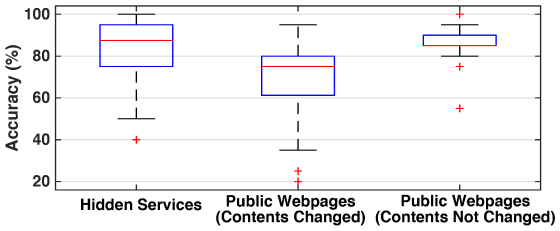

Figure 6: Identification accuracy comparison among (1) hidden services, (2) public webpages with content changes, and (3) public webpages without content changes.

Second, we examined the screenshots for each webpage loading, and we found none of the 50 hidden services changed their displayed contents during the data collection period (about 1 day). In comparison, 31 of the 50 public webpages displayed different contents during the collection, which further affects the consistency among power traces of public webpages. Figure 6 shows that public webpages without content changes have higher average accuracy than public webpages with content changes.

The content changes of public webpages are either due to frequent website updates (e.g. for news websites), or because different versions of the webpage were loaded based on the geographical location of the Tor exit relay. For example, we checked the four public webpages with identification accuracy lower than 40%, and we found two of them were loaded with different versions (based on website language, content, and layout): there were 9 versions for `paypal.com` (with accuracy of 30%), and three versions for `dropbox.com`. (with accuracy of 25%).

To improve the identification accuracy, we tried to use a specific version of these websites for training and testing. For example, we used the traces for `paypal.com/de` instead of `paypal.com` and re-conducted the whole training and testing. The identification accuracy was improved from 30% to 90% for this specific webpage, and it increased from 79.05% to 79.65% for all webpages.

4.2. Impact of Training and Testing on Different Smartphones

When we used the dataset collected from smartphone A to train the model and used the model to identify traces from smartphone B, the original classification method did not provide good results. To address this issue, we modified the model as follows. By examining the spectrograms of power traces from different phones, we observed that beyond a specific frequency (about 3 KHz), the difference in magnitude distribution among spectrograms depends more on the smartphone being used than the webpage being loaded. To address this issue, we increased the frequency resolution of the spectrogram, and we used the magnitudes of the first 250 frequency points (ranging from 0 Hz to 3039.6 Hz) as the feature vector for each power trace. We then used Sequential Minimal Optimization (SMO) algorithm [17] for model training and testing. The resulting identification accuracies using two phones are presented in Table 5.

Table 5: Webpage identification accuracy using different phones for training and testing

Train	Test	All webpages		Public webpage only		Hidden service only	
		Rank-1	Rank-5	Rank-1	Rank-5	Rank-1	Rank-5
A	B	43.13%	70.3%	48.75%	80.15%	47.05%	77.45%
B	A	36.58%	63.78%	43.2%	74.7%	40.35%	69.3%

Even though the accuracy decreases when training and testing on different smartphones (36.58% to 43.13% for all webpages using two phones, compared to 79.05% to 85.7% using the same smartphone for training and testing), it is still significantly higher than that of random chance at 1%.

4.3. Impact of Network Characteristics

In dataset #3, we collected training and testing traces using LTE network. The results in Table 6 show that the identification accuracy when training and testing on LTE (71.75%) is worse than that when using WiFi (79.05%, see Table 4). Consistently with our experiments based on WiFi, we observed that the accuracy for public webpages is lower than that of hidden services, and the accuracy decrease for public webpages (68.7% using LTE, compared to 76.3% using WiFi) is larger than the decrease for hidden services (84.3% using LTE, compared to 87.3% using WiFi).

Table 6: Webpage identification accuracy using LTE

All webpages		Public webpage only		Hidden service only	
Rank-1	Rank-5	Rank-1	Rank-5	Rank-1	Rank-5
71.75%	84.7%	68.7%	84.4%	84.3%	93.2%

One possible explanation is that LTE network contributes to additional noise in the power traces, compared to WiFi. We measured the loading time of six public webpages using LTE. The results are shown in Figure 7. Compared with the results of using WiFi (see Figure 1), the average loading time increased by 9.1%, which indicates that LTE introduced more unpredictable delays than WiFi.

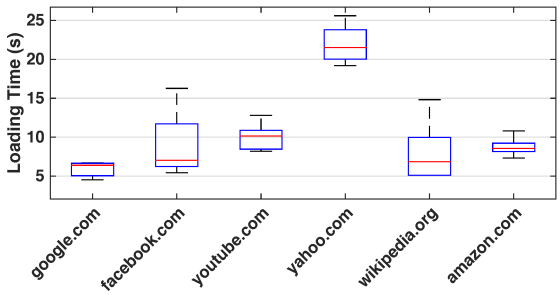

Figure 7: Loading time of public webpages using LTE network.

We also trained the model with WiFi traces, and tested it using LTE traces, and vice versa. The identification accuracies shown in Table 7 are significantly lower than that when using LTE traces for both training and testing. This indicates that the adversary needs to train a different model for each network.

Table 7: Cross testing using LTE and WiFi networks

Train	Test	All webpages		Public webpage only		Hidden service only	
		Rank-1	Rank-5	Rank-1	Rank-5	Rank-1	Rank-5
WiFi	LTE	24.55%	49.4%	24.25%	54.2%	42.9%	69.8%
LTE	WiFi	21.8%	47.25%	21%	47.05%	28.15%	63.2%

4.4. Impact of Battery Charging level

When the smartphone is charging, a large part of power is used to charge the battery. In contrast, when the battery is fully charged, almost all current from the charger is used to power the phone, including loading the webpage. Thus, battery charging level impacts the amount of information that can be inferred from the power trace on webpage loading. We collected the power traces for 30% to 50% battery level. Before each round of the collection, we discharged the phone battery to 30%. Then we shuffled the 100 webpages into a random sequence, and we collected one trace for each webpage following this sequence. After collecting all 100 traces in one round, the battery level increased to about 50%. Then we discharged the battery to 30% again and repeated the above collection process. This process was repeated 40 times in total. Dataset #4 includes the traces collected while the smartphone charging level was between 30% and 50%.

We used half of dataset #4 for training, and the other half for testing. The identification results are shown in Table 8. Although accuracy decreases sharply when the battery is not fully charged, it is still significantly higher than the baseline accuracy using random guessing (i.e., 1% for all the 100 webpages, and 2% for the 50 public webpages or 50 hidden services). One reason is that the maximum charging current is capped by an upper limit (1.15 A in this case), which is imposed by the smartphone's charging circuit (see Figure 8), where 98.17% of the samples in the power trace have current value below 1.15 A. This limitation distorts the power signals and decreases the effectiveness of our technique. Further, we found that there were strong noisy signals periodically appearing in each trace. The same signals did not appear in the power traces collected when the battery was fully charged.

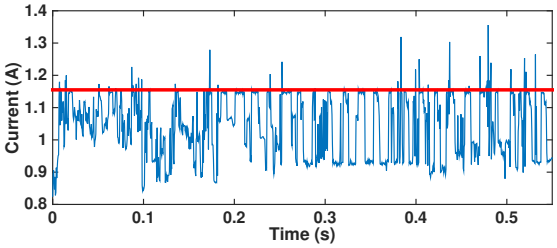

Figure 8: Capped current when smartphone battery is partially charged.

Table 8: Webpage identification accuracy when battery level is from 30% to 50%

All webpages		Public webpage only		Hidden service only	
Rank-1	Rank-5	Rank-1	Rank-5	Rank-1	Rank-5
36%	52.1%	34.5%	58.5%	46%	65.9%

4.5. Impact of Tor Circuits Type

We evaluated the scenario in which training and testing data were collected using the same *fixed* circuit. The corresponding datasets are indicated as #5 and #6 in Table 3. Table 9 reports the identification accuracies when using fixed circuits "Cir-1" and "Cir-2" on different phones. Compared to the results in Table 4 obtained using automatic circuits, accuracies are not significantly higher, or are even lower in some cases. One possible explanation is that the throughput of each relay in the fixed circuits are always changing. This adds unpredictable delays for each webpage loading and increases the inconsistency between power traces for training and testing. In our experiments, we observed that during some periods the fixed circuits could not be used to load any webpages at all. In contrast, automatic circuits are constructed using optimized path selection protocol. In practice, our experiments show that the adversary should use automatic circuits to collect training traces.

Table 9: Webpage identification accuracy using fixed Tor circuits

Phone	Circuit	All webpages		Public webpage only		Hidden service only	
		Rank-1	Rank-5	Rank-1	Rank-5	Rank-1	Rank-5
A	Cir-1	78.1%	89.8%	79.6%	91.6%	84.8%	93.8%
B	Cir-2	82.6%	91.3%	85.7%	93.4%	86.1%	93.7%

4.6. Comparison of the Attack with and without Using Tor

Previous work [2] evaluated the effectiveness of using power traces to identify webpages on smartphones. However, it did not consider web browsing anonymization techniques, such as Tor. In this section, we investigate the impact of Tor on identification accuracies by comparing our results with the accuracies reported in [2].

Collection of power traces in this work and in [2] was performed using the same procedures and parameters, with the exception of trace length. Because

webpage loading is usually completed within 2 seconds when not using Tor, and in 10 seconds with Tor (see Figure 1), we compare our results with 2-second traces results from [2]. Table 10 summarizes the identification accuracy when loading webpages via WiFi.[1]

Table 10: Comparison of webpage identification accuracy with and without Tor.

	Battery Fully Charged	Battery 30% Charged
Not using Tor	84.3%	75.2%
Using Tor	82.2%	34.5%

We observe that when the battery is fully charged, we achieved similar identification accuracies (82.2% with Tor vs. 84.3% w/o Tor). When the battery level is between 30% and 50%, the drop in website identification accuracy is more pronounced with Tor, although still substantially higher than the random-selection baseline of 2%.

5. Related Work

In this section, we review related work on attacks on Tor (Section 5.1), and on side-channel attacks based on power analysis (Section 5.2).

5.1. Attacks on Tor

There are a number of papers that focus on attacks on Tor. These papers can be broadly categorized into *passive* attacks (i.e., based on traffic analysis) and *active* attacks (based on traffic modification).

Passive Attacks Based on Traffic Analysis. Fingerprinting attacks and traffic confirmation attacks belong to this category. Website fingerprinting attacks enable an attacker to detect patterns that are indicative for webpages in Tor traffic. Herrmann et al. [18] presented a method that applies common text mining techniques to the normalized frequency distribution of observable IP packet sizes, so as to reveal requested websites. Panchenko et al. [19] showed that Tor did not offer sufficient security against website fingerprinting. Their attack relies on volume, time, and direction of the traffic to reveal websites. Cai et al. [20] presented a webpage fingerprinting attack that was able to defeat several defenses against traffic analysis attacks, such as application-level defenses HTTPOS and randomized pipelining over Tor. Abbott et al. [21] provided an attack to identify a fraction of the Tor users who used malicious exit nodes. This attack tricked a user's web browser into sending a distinctive signal over the Tor network. Such signal could be detected by traffic analysis.

In traffic confirmation attacks, the adversary must be able to eavesdrop both ends of a communication over a long time period. Levine et al. [22] investigated

[1] In [2], authors report webpage identification accuracy on a Galaxy S6 exclusively on WiFi.

timing analysis attacks on low-latency mixed systems, and proposed a technique named defensive dropping to mitigate timing attacks. Hopper et al. [23] presented two attacks on low-latency anonymity schemes using the network latency information. The first attack allowed a pair of colluding websites to predict whether two connections from the same Tor exit node are using the same circuit. The second attack enabled a malicious website to gain location information about a client when he visits the website. Sun et al. [24] proposed asymmetric traffic correlation attack on Tor with high accuracy, and increased the threat of AS-level attacks significantly. Bauer et al. [25] demonstrated that routing optimization prevents Tor from providing strong anonymity. They proposed attacks using low-resource Tor nodes to compromise the entrance and exit nodes on Tor circuits. Kwon et al. [26] presented a passive attack against hidden services and their users using circuit fingerprinting attack, where the adversary can identify the presence of client or server hidden service activities. Murdoch et al. [27] introduced traffic-analysis techniques based on a partial view of the network. Their attack could infer the nodes used to relay the anonymous streams and therefore reduced Tor anonymity. Chakravarty et al. [28] presented a remotely-mounted attack to expose the network identity of an anonymous client, hidden service, or anonymizing proxy. They employed single-end bandwidth estimation tools and a colluding network entity to modulate traffic directed to the victim.

Some passive side-channel attacks try to infer sensitive information other than user identities or traffic destinations from web application usage. For instance, Schaub et al. [29] presented an attack on web search engines to retrieve the user's search query inputs. They first intercepted and analyzed the packet flow associated with the suggest boxes from the search engine for each input character, then built a probability distribution of packet sizes for each letter. They proposed a stochastic algorithm that utilized the distribution probabilities to infer the complete query text.

Our work differs from the above studies mainly in the following aspects: (1) the attack presented in this paper is based on USB power analysis, rather than network traffic analysis or modification; (2) in our model, the goal of the adversary is to learn which websites have been accessed by Tor users, rather than to obtain the user's inputs entered in a search engine; (3) we focus on web page identification for both regular web pages and web pages served by Tor hidden services; and (4) we study a side-channel attack on smartphones, rather than on desktop or laptop computers.

Active Attacks Against Tor. Wang et al. [30] investigated the fundamental limitations of flow transformations in achieving anonymity. They showed that flow transformations could not necessarily provide the level of anonymity people expected or believed. Barbera et al. [31] introduced a new Denial-of-Service attack against Tor Onion Routers. They exploited a design flaw used by Tor software to build virtual circuits. Their attack only needed a fraction of the resources required by a network DoS attack to achieve similar damage on the Tor network.

Our work differs from the above papers because it does not require active

changes to the content of webpages, or traffic injection or manipulation.

5.2. Side-channel Attacks Based on Power Analysis

Clark et al. [32] measured power consumption data collected from hacked wall outlets to identify webpages loaded on computers. Genkin et al. [33] analyzed electric potential from computer chassis to extract encryption keys. Yang et al. [2] first presented an attack on mobile devices that allows the adversary to identify loaded webpages while the smartphone is charging by controlling the USB charging port. The main differences between our attack, and the attacks presented in these papers are: (1) all power traces used in this paper were collected while using Tor. This affects traces in several important ways, because the use of Tor leads to the generation of complex power patterns due to circuit types and composition, and geographic location of the Tor routers in a circuit; and (2) besides regular webpages, we also consider hidden services that are exclusively existent in Tor network.

5.3. Covert Channels and Attack Countermeasures

Covert channels can be used for surreptitious exfiltration of sensitive data. Spolaor et al. [34] developed a covert system to send data as power bursts from a smartphone to a malicious charging station. Zhang et al. [35] demonstrated that it is possible to build a covert communication channel by adjusting the silence periods of VoLTE traffic on smartphones. In general, covert channel attacks require the smartphone to run a malicious apps for data transmission purpose, while our applies to otherwise non-compromised devices.

Countermeasures to side-channel attacks have been investigated in several papers. Kocher [36] proposed strategies to design and validate cryptographic devices against power-analysis attack, such as using short-lived session keys instead of a long-lived initial key. More generally, Meng et al. [37] discussed how to apply blockchain techniques to protect data privacy among collaborative intrusion detection systems. Because the focus of our work is to propose and evaluate a new side-channel attack on Tor, we consider the study of countermeasures as future work.

6. Conclusion and Future Work

In this paper, we demonstrated a technique based on USB power analysis that allows a malicious charging station to identify which webpages are loaded on a smartphone using Tor. To our knowledge, this is the first work to study attacks on Tor based on smartphone power side-channels.

We validated our attack under realistic smartphone constraints by collecting and analyzing power traces under several scenarios, including different networks (WiFi and LTE), different devices, and different battery charging levels.

We correctly identified webpages visited using the official mobile Tor browser. We achieved accuracies between 36.58% and 85.7% when the battery was fully charged, and between 34.5% and 46% when the battery level was at 30%-50%.

In comparison, accuracy obtained by random website selection is 1% for all websites, and 2% when considering hidden services or public webpages alone.

We consider this work the first step towards a full characterization of power side-channel attacks on Tor. To this end, there are several combinations of variables that we did not consider, including user interactions during webpage loading. We leave this and other configurations to future work. Additionally, we plan to address countermeasure to the attack presented in this paper in future work.

Acknowledgement

This work is supported by U.S. National Science Foundation under grants CNS-1253506(CAREER), CNS-1618300, and CNS-1619023.

References

[1] R. Dingledine, N. Mathewson, P. Syverson, Tor: The second-generation onion router, in: Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13, SSYM'04, USENIX Association, Berkeley, CA, USA, 2004, pp. 21–21.

[2] Q. Yang, P. Gasti, G. Zhou, A. Farajidavar, K. S. Balagani, On inferring browsing activity on smartphones via USB power analysis side-channel, IEEE Transactions on Information Forensics and Security 12 (5) (2017) 1056–1066. doi:10.1109/TIFS.2016.2639446.

[3] R. W. Gehl, Power/freedom on the dark web: A digital ethnography of the dark web social network, new media & society 18 (7) (2016) 1219–1235.

[4] Orfox: A Tor browser for Android, https://guardianproject.info/apps/orfox/, accessed: 2017-03-10 (2017).

[5] DP800 series DC power supplies, https://www.rigolna.com/products/dc-power-supplies/dp800/, accessed: 2017-05-25.

[6] Qualcomm quick charge 2.0 chipset, https://www.qualcomm.com/products/quick-charge-2, accessed: 2017-03-10 (2017).

[7] Battery charging specification revision 1.2, http://www.usb.org/developers/docs/devclass_docs/, accessed: 2017-05-24.

[8] USB-6211 (multifunction I/O device), http://www.ni.com/en-us/support/model.usb-6211.html, accessed: 2017-05-25.

[9] Tor on Android, https://www.torproject.org/docs/android.html.en, accessed: 2017-03-10 (2017).

[10] HTTPS everywhere, https://www.eff.org/https-everywhere, accessed: 2017-03-10 (2017).

[11] Tor: Hidden service protocol, https://www.torproject.org/docs/hidden-services, accessed: 2017-03-10 (2017).

[12] The hidden wiki, http://zqktlwi4fecvo6ri.onion/wiki/index.php/Main_Page, accessed: 2017-03-10 (2017).

[13] Tor path specification, https://gitweb.torproject.org/torspec.git/tree/path-spec.txt, accessed: 2017-03-10 (2015).

[14] S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Transactions on Acoustics, Speech, and Signal Processing 28 (4) (1980) 357–366. doi:10.1109/TASSP.1980.1163420.

[15] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The WEKA data mining software: An update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18. doi:10.1145/1656274.1656278.

[17] J. C. Platt, Advances in kernel methods, MIT Press, Cambridge, MA, USA, 1999, Ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208.

[18] D. Herrmann, R. Wendolsky, H. Federrath, Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial Naïve-bayes classifier, in: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, CCSW '09, ACM, New York, NY, USA, 2009, pp. 31–42. doi:10.1145/1655008.1655013.

[19] A. Panchenko, L. Niessen, A. Zinnen, T. Engel, Website fingerprinting in onion routing based anonymization networks, in: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society, WPES '11, ACM, New York, NY, USA, 2011, pp. 103–114. doi:10.1145/2046556.2046570.

[20] X. Cai, X. C. Zhang, B. Joshi, R. Johnson, Touching from a distance: Website fingerprinting attacks and defenses, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12, ACM, New York, NY, USA, 2012, pp. 605–616. doi:10.1145/2382196.2382260.

[21] T. G. Abbott, K. J. Lai, M. R. Lieberman, E. C. Price, Browser-based attacks on Tor, in: Proceedings of the 7th International Conference on Privacy Enhancing Technologies, PET'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 184–199.

[22] B. N. Levine, M. K. Reiter, C. Wang, M. Wright, Timing Attacks in Low-Latency Mix Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 251–265. doi:10.1007/978-3-540-27809-2_25.

[23] N. Hopper, E. Y. Vasserman, E. Chan-Tin, How much anonymity does network latency leak?, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS '07, ACM, New York, NY, USA, 2007, pp. 82–91. doi:10.1145/1315245.1315257.

[24] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, P. Mittal, RAPTOR: Routing attacks on privacy in Tor, in: Proceedings of the 24th USENIX Conference on Security Symposium, SEC'15, USENIX Association, Berkeley, CA, USA, 2015, pp. 271–286.

[25] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, D. Sicker, Low-resource routing attacks against Tor, in: Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society, WPES '07, ACM, New York, NY, USA, 2007, pp. 11–20. doi:10.1145/1314333.1314336.

[26] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, S. Devadas, Circuit fingerprinting attacks: Passive deanonymization of Tor hidden services, in: 24th USENIX Security Symposium (USENIX Security 15), USENIX Association, Washington, D.C., 2015, pp. 287–302.

[27] S. J. Murdoch, G. Danezis, Low-cost traffic analysis of Tor, in: Proceedings of the 2005 IEEE Symposium on Security and Privacy, SP '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 183–195. doi:10.1109/SP.2005.12.

[28] S. Chakravarty, A. Stavrou, A. D. Keromytis, Traffic analysis against low-latency anonymity networks using available bandwidth estimation, in: Proceedings of the 15th European Conference on Research in Computer Security, ESORICS'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 249–267.

[29] A. Schaub, E. Schneider, A. Hollender, V. Calasans, L. Jolie, R. Touillon, A. Heuser, S. Guilley, O. Rioul, Attacking suggest boxes in web applications over HTTPS using side-channel stochastic algorithms, in: CRiSIS, 2014.

[30] X. Wang, S. Chen, S. Jajodia, Network flow watermarking attack on low-latency anonymous communication systems, in: 2007 IEEE Symposium on Security and Privacy (SP '07), 2007, pp. 116–130. doi:10.1109/SP.2007.30.

[31] M. V. Barbera, V. P. Kemerlis, V. Pappas, A. D. Keromytis, CellFlood: Attacking Tor Onion Routers on the Cheap, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 664–681. doi:10.1007/978-3-642-40203-6_37.

[32] S. Clark, H. Mustafa, B. Ransford, J. Sorber, K. Fu, W. Xu, Current events: Identifying webpages by tapping the electrical outlet, in: J. Crampton, S. Jajodia, K. Mayes (Eds.), Computer Security - ESORICS 2013, Vol. 8134 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 700–717. doi:10.1007/978-3-642-40203-6_39.

[33] D. Genkin, I. Pipman, E. Tromer, Get your hands off my laptop: Physical side-channel key-extraction attacks on PCs, in: L. Batina, M. Robshaw (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2014, Vol. 8731 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2014, pp. 242–260. doi:10.1007/978-3-662-44709-3_14.

[34] R. Spolaor, L. Abudahi, V. Moonsamy, M. Conti, R. Poovendran, No free charge theorem: a covert channel via USB charging cable on mobile devices, CoRR abs/1609.02750.

[35] X. Zhang, Y. A. Tan, C. Liang, Y. Li, J. Li, A covert channel over VoLTE via adjusting silence periods, IEEE Access 6 (2018) 9292–9302. doi:10.1109/ACCESS.2018.2802783.

[36] P. Kocher, Design and validation strategies for obtaining assurance in countermeasures to power analysis and related, in: Attacks, in the proceedings of the NIST Physical Security Workshop, 2005.

[37] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, J. Han, When intrusion detection meets blockchain technology: A review, IEEE Access 6 (2018) 10179–10188. doi:10.1109/ACCESS.2018.2799854.