Lost Traffic Encryption: Fingerprinting LTE/4G Traffic on Layer Two

Katharina Kohls
katharina.kohls@rub.de
Ruhr University Bochum
Germany

David Rupprecht
david.rupprecht@rub.de
Ruhr University Bochum
Germany

Thorsten Holz
thorsten.holz@rub.de
Ruhr University Bochum
Germany

Christina Pöpper
christina.poepper@nyu.edu
NYU Abu Dhabi
United Arab Emirates

ABSTRACT

Long Term Evolution (LTE) provides the communication infrastructure for both professional and private use cases and has become an integral part of our everyday life. Even though LTE/4G overcomes many security issues of previous standards, recent work demonstrates several attack vectors on the physical and network layers of the LTE stack. We do, however, have only limited insights into the security and privacy aspects of the second layer.

In this work, we investigate the impact of fingerprinting attacks on encrypted LTE/4G layer-two traffic. Traffic fingerprinting enables an adversary to exploit the metadata side-channel of transmissions—with severe consequences for the user's privacy. In multiple lab and commercial network experiments, we demonstrate the feasibility of passive and active fingerprinting attacks. First, passive website fingerprinting allows the attacker to learn a user's accessed website from encrypted transmissions. While being a well-known attack in other contexts, we provide an extensive performance baseline of state-of-the-art website fingerprinting attacks of encrypted LTE traffic in a lab setup and successfully repeat the experiments in a commercial network. Second, in an active identity-mapping attack, we inject watermarks and localize users within a radio cell. Our attacks succeed for the current LTE/4G specification and exploit features that also persist in the upcoming 5G standard.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**;

KEYWORDS

LTE, Website Fingerprinting, Identification Attack

ACM Reference Format:
Katharina Kohls, David Rupprecht, Thorsten Holz, and Christina Pöpper. 2019. *Lost Traffic Encryption*: Fingerprinting LTE/4G Traffic on Layer Two. In *12th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '19), May 15–17, 2019, Miami, FL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3317549.3323416

WiSec '19, May 15–17, 2019, Miami, FL, USA

ACM ISBN 978-1-4503-6726-4/19/05…$15.00
https://doi.org/10.1145/3317549.3323416

1 INTRODUCTION

LTE is the latest widely-deployed mobile communication standard and serves diverse use case scenarios, ranging from browsing to the implementation in critical infrastructures. LTE provides high-performance transmissions and sophisticated security features and finds extensive integration into our daily communication. Unfortunately, this integration allows an adversary to achieve tremendous impact in case of successful attacks.

Due to its importance, LTE motivates various attacks that range from denial-of-service through jamming [3, 22, 30, 31], over downgrade attacks that enforce a more insecure communication standard [24, 33, 40], to identification and localization attacks that reveal the presence of a user within a radio cell [40]. The majority of these attacks set a focus on either the physical layer (layer one) or the network layer (layer-three) of the protocol stack and leave a blind spot in-between on the second layer (data link layer), which ranges from the LTE Medium Access Control (MAC) to the Packet Data Convergence Protocol (PDCP). Recently, Rupprecht et al. [38] presented the first collection of attacks on layer two. Besides an active DNS redirection attack (called *aLTEr*), their work also introduces an identity mapping that enables website fingerprinting on encrypted LTE traffic. Their results predict severe consequences for the *privacy* of users.

An adversary with the ability to fingerprint encrypted traffic, either actively [49, 50] or passively [28], is often in a position to recover sensitive information about a user. Privacy leaks by traffic fingerprinting attacks first emerged when Cheng et al. [7] in 1998 found out that—even without access to the encrypted payload of a transmission—we can distinguish websites just from meta information like the number of packets sent over time. Since then, advances in classification techniques [14, 15], models of the user behavior [36], and modern machine-learning algorithms [37, 51] helped to improve the success of fingerprinting attacks in more challenging scenarios. Systems with additional security features, e. g., the Tor anonymity network [46], limit the threat of traffic analysis. Nevertheless, there is a large body of powerful attacks that also succeed in the context of Tor [10, 21, 25, 34, 35]. While this area of research emerged to a state where we find advanced attack concepts, little do we know about the success of state-of-the-art fingerprinting attacks on LTE layer-two traffic.

The usual *website* fingerprinting attack includes a training phase, in which the adversary records a preferably high number of sample traffic that resembles the transmission characteristics for a set of websites. There are two *fundamental differences* between usual website fingerprinting (WF) and fingerprinting LTE layer-two traffic. First, LTE adversaries use a downlink sniffer to access *all* transmissions within one radio cell. Conventional attacks, on the other hand,

often exploit compromised Tor relays, monitor a router of the user's ISP, or record traces in a local network [25]. The latter requires control over physical nodes and adversarial access to the network infrastructure. On the other hand, radio layer attacks use affordable equipment, passive wireless monitoring can hardly be backtracked, and a certain amount of mobility allows the adversary to access different cells of multiple providers. Second, a further significant difference arises from the fact that we no longer obtain transport- and network-layer traffic of the TCP/IP protocol stack, but record a different set of metadata information from LTE layer two. Looking back on a strong series of state-of-the-art fingerprinting attacks, we cannot be sure about their classification capabilities on LTE-specific traffic characteristics.

In our work, we analyze the feasibility and impact of active and passive fingerprinting of LTE layer-two traffic. We begin our work with detailed documentation of the adversary model and provide a first experimental study of the influencing factors for layer-two website fingerprinting. These influencing factors include i) the effects of varying website contents over time, ii) differences between the hardware and software of multiple devices, and iii) the impact of application-layer obfuscation. Our experiments provide a performance baseline of attacks in a controlled private network and serve as an upper-bound benchmark. The results of our performance baseline experiments reveal that state-of-the-art classification techniques can successfully be transferred to the context of LTE. In a closed-world setup with 50 websites, we identify sites with a success rate in the range of 91 % to 95 % for simpler scenarios, but also experience the negative effects of obfuscation (53 % success). Our data set consists of a total of 96,262 traces recorded over seven months including 93,490 traces recorded in our private network setup and 2772 live network traces from a commercial LTE network that we use for two real-world case studies.

Our case studies build the second evaluation step, in which we conduct attacks in a *commercial network*. First, we transfer the passive website fingerprinting to the new network and test whether the attack remains successful in this more challenging setup. Second, we actively inject watermarks in the user data stream to derive the identity and location of a user within the radio cell. Our results demonstrate that the use of layer-two scheduling information helps to reliably identify website traffic within a conventional commercial network with multiple active users with a success rate of 90 %. While we demonstrate the severe privacy issues of an *untargeted* website fingerprinting attack, our second case study proves the ability to identify and localize a *specific* user within a cell. Combining both attacks, an adversary gains a dominant position to learn sensitive information about arbitrary users in a radio cell and can use this information as a starting point for further attacks.

While the first part covers the technical characteristics of LTE traffic fingerprinting, we conclude our work with a detailed discussion of the impact of both attacks. In particular, we address the threat of large-scale adversaries and discuss the consequences of real-world deployment. With the strict security and privacy implications of both fingerprinting attacks in mind, our work is also an appeal to the sustainable design of the upcoming 5G standard. In particular, similarities in the protocol specifications indicate the persisting threat of our demonstrated attack vectors.

In summary, the main contributions of this paper are:

- **Performance Baseline.** Using meta-analysis as a starting point, we conduct a *website fingerprinting* attack in a controlled lab environment and analyze the influencing factors that impact the success of the attack. Our experiments use state-of-the-art classification techniques from different contexts and provide a first performance baseline of website fingerprinting on LTE layer-two traffic.
- **Real-World Case Studies.** We conduct an active and a passive fingerprinting attack in a commercial network and analyze their feasibility under the more challenging circumstances of a real-world cell. Our results show that both website fingerprinting and user identification/localization are possible in practical scenarios with convincing success rates.
- **Discussion.** We provide a detailed discussion of the real-world effects of successful LTE layer-two fingerprinting. In particular, we focus on the capabilities of large-scale adversaries, existing countermeasure options, and the impact of our attacks on the upcoming 5G specification.

2 PRELIMINARIES

Before diving into detail with our experiments, we introduce the technical background of LTE layer-two characteristics and define the adversary model for our website fingerprinting and user identification attacks.

2.1 LTE Layer Two

LTE specifies the transmission procedure for messages exchanged between the phone (User Equipment (UE)) and the base station (Evolved NodeB (eNodeB)) with a layered protocol stack that is comparable to the ISO/OSI reference model. Our interest is in the second layer, i. e., the *data link layer*, that extends the underlying *physical layer* with additional services to manage the medium access and to provide mechanisms for integrity, reliability, and security. Layer two consists of three sub-layers that schedule the medium access (Medium Access Control (MAC)), manage data units (Radio Link Control (RLC)), and perform ciphering and optional IP header compression (Packet Data Convergence Protocol (PDCP)).

The MAC sub-layer is the first point of interest for our fingerprinting attacks, as we find temporary user identities for the management of active radio connections. The Radio Network Temporary Identifier (RNTI) helps an adversary to distinguish multiple UE connections in the radio cell and, eventually, allows to map recorded traffic to different connections. Please note that the user-specific Cell-RNTI, sub-range of the RNTI (C-RNTI) resembles a sub-range of the RNTI, which does not imply any technological differences in the context of traffic fingerprinting. In the remainder of this work, we use the RNTI to distinguish connections both in an *untargeted* attack and focus on a specific user (and the RNTI, respectively) in a *targeted* attack.

The UE obtains the RNTI by performing a Random Access Procedure (RAP) with the eNodeB, which then responds with an unencrypted Random Access Response (RAR). In all subsequent transmissions, e. g., when visiting a website, the MAC layer of the eNodeB identifies which radio resources are available and allocates them to the RNTI of the UE. This allocation is signalled to the UE using the Downlink Control Information (DCI) for all transmissions from the

eNodeB towards the UE. For the *uplink* direction, the UE signals a scheduling request and receives the uplink allocation. We can use this information to distinguish the transmissions of multiple UEs in uplink and downlink direction and derive individual traces from this. In the fingerprinting attacks, we use these traces to classify websites (passive website fingerprinting attack) or identify injected watermarks (active watermarking attack).

In addition to the scheduling information, the eNodeB and the UE decode all information of the underlying layers and decrypt incoming frames of the PDCP layer. Even though this does not grant access to the encrypted payload of a packet, we can still derive information like the PDCP packet length or sequence number and use this as metadata input for traffic fingerprinting.

2.2 Traffic Fingerprinting

The potential of traffic fingerprinting is unexplored in the context of LTE, but we find a large body of prior work in alternative settings.

Problem Statement. The identification of websites from encrypted traffic is a classification problem [37] where the adversary gathers labeled traffic traces of candidate websites for a training set to later test an unlabelled sample trace against it. We identify three fundamental attack characteristics to differentiate prior work. First, we can choose from a series of *classifiers* that take care of the comparison of website traffic. *Features* serve as input for the classifier and can originate directly from recorded traffic information, or they can be processed from the combination of different characteristics. Finally, the *setup* defines the experimental space with an open- or closed-world classification problem and the number of websites in the training and test sets.

State of the art. In 1998, Cheng et al. [7] introduced a first traffic fingerprinting attack that uses a two-dimensional feature space to identify web pages. Iterations of follow-up work continuously improved the above attack characteristics to get closer to a realistic evaluation while maintaining convincing success rates [14, 29, 44]. Up to this point, attacks successfully identify web pages from *simple* encrypted traffic, but are likely to fail with additional protection through, e. g., the Tor anonymity system. Follow-up work uses a k-Nearest-Neighbors classifier [47] or Support Vector Machines (SVM) [36] with success rates around 90 % on obfuscated Tor traffic. Current attacks extend this by automated feature engineering [51] and deep learning [37] with high success rates in large data sets.

Fingerprinting Attacks. We define two separate fingerprinting attacks that exploit the metadata information of encrypted LTE layer-two traffic:

(1) **Website Fingerprinting.** The adversary aims to learn the accessed websites from recorded user data traffic. He compares the unknown trace with a pre-recorded database of labeled website candidates and conducts a closed-world classification.

(2) **Identification and Localization.** The adversary aims to learn the temporary identity and/or presence of a specific user within a cell. This becomes possible by sending a particular traffic pattern to the public identity of the victim, which can then be recognized in the encrypted traffic. The attack is active and uses a sniffing tool in combination with a messaging interface (e. g., WhatsApp).

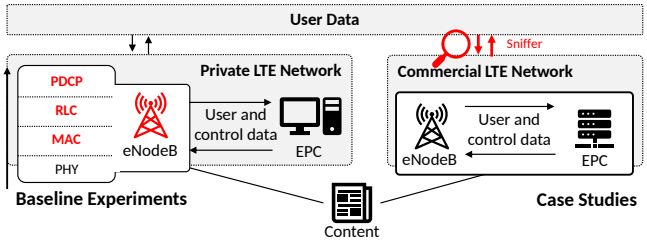

Figure 1: Comparison of experimental setups. The lab setup serves for the baseline experiments of Section 4; the commercial LTE network serves for the case studies of Section 5.

Attacker Capabilities. For the above attacks, we assume an adversary capable of sniffing the downlink and broadcast traffic of at least one LTE cell. The adversary does not know any key material of the victim, i. e., he cannot decrypt transmissions and has no access to the payload or IP header information of a packet. Furthermore, he does not know any internal LTE identities, e. g., the International Mobile Subscriber Identity (IMSI) or Temporary Mobile Subscriber Identity (TMSI), but can only learn the *public identity* of a user to contact him through a third party app (e.g., a messaging service such as WhatsApp). The adversary can access and decode transmissions ranging from the physical layer up to the PDCP layer.

Technical Requirements. The technical requirements for a passive attack can be satisfied by using open-source LTE software stacks such as srsLTE [43] implemented on a Software Defined Radio (SDR) in combination with an analysis framework (sniffer) like Airscope [42], imdea OWL [5, 6], or other commercial systems [39]. In the case of active interference with the victim (identification and localization), the adversary repeatedly sends messages through a suitable interface. As the software stack implementation conforms with the official specification of LTE, both attacks can be conducted in an arbitrary radio cell. Nevertheless, transmission characteristics and, consequently, metadata information, might be sensitive to provider-specific configurations. In particular, this applies to the resource scheduling algorithms influencing the resource allocation.

In contrast to conventional fingerprinting attacks, the radio layer adversary does not depend on the physical access to network nodes. Consequently, attacks are more stealthy (wireless downlink sniffer cannot be backtracked), the required tools are affordable (less than $160), and a certain amount of mobility allows the adversary to cover different cells of multiple providers.

3 EXPERIMENTAL SETUP

In our experiments, we use a private LTE network to create a controlled lab environment in which we can isolate different influencing factors for fingerprinting attacks.

3.1 Network Setup

Our network setup consists of three main components. The (i) UE simulates the website requests of a user, the (ii) LTE radio cell that handles the requests and responses of the UE, and the different (iii) web servers that provide the requested contents (cf. Figure 1). For our attacks, we focus on the transmissions between the UE

and the eNodeB that serves as a base station of the radio cell of the LTE network. By exchanging information between the user's smartphone and the base station (user data), we get access to transmissions up to the PDCP layer and record traces of website requests for the training corpus and our attack. In the case of the private LTE network, we can record traffic in the eNodeB component of the network and directly access the decoded PDCP and DCI information. In the commercial network setup, we do not control the eNodeB component and access transmissions with a sniffing tool [42]. We next define the technical characteristics of our network components and introduce the metadata features that we derive from the traces.

Network Components. In the network setup, we focus on the user's device (UE) and the base station (eNodeB). The core network and web servers are relevant for the network setup, but are not part of the attacks.

- **User Equipment (UE).** The UE is a device, e. g., a smartphone, capable of sending and receiving mobile data via LTE. A programmable SIM card allows us to connect to our private LTE network. In our setup, we test four different smartphones that we either control via the Android Debug Bridge (ADB) or a simulated USB keyboard. The smartphones connect to the eNodeB component of our private LTE network and make requests for a defined set of websites. A detailed device specification is provided in Table 2.
- **Evolved Node B (eNodeB).** The eNodeB functions as a base station that provides a mobile data connection via LTE and connects to the Evolved Packet Core (EPC) with core network functionality. In our private LTE network, we use a `B210 USRP` and the srsLTE software stack version `18.06`.
- **Evolved Packet Core (EPC).** The EPC is the core network that exchanges user and control data, e. g., website requests and LTE specific protocol data, between the eNodeB and the EPC. For user data, it functions as a gateway to forward IP data, e. g., website requests and responses to the Internet. For our attacks, we do not interfere with the core network.
- **Web Servers.** The network forwards all website requests to the original web servers of a site and transmits the response through the LTE network. Again, we do not interfere with the web servers.

Traffic Metadata. We use the RNTI to distinguish different traces and refer to DCI information to understand the resource allocation. Furthermore, we derive metadata features from the decoded PDCP information consisting of five features, i. e., the $(f_1, rnti)$ RNTI, (f_2, seq) PDCP sequence number, (f_3, len) PDCP packet length, (f_4, abs) absolute timestamp, and (f_5, rel) relative timestamp of each packet. Besides this *raw* transmission information, we generate an aggregated representation in which we summarize packets in time-based windows. We apply a window of 500 ms length and aggregate the packet occurrences in each window. Following this approach, we compress the original raw trace into five new features, i. e., the (f_1, win) window index, (f_2, cnt) number of packets in the window, (f_3, iat) average inter-arrival time between packets, (f_4, byt) total amount of data received in a window, and (f_5, seq) average sequence number within the window.

3.2 Recording Procedure

For the experiments of Section 4 and the website fingerprinting case study of Section 5, we follow a general recording procedure to gather the data sets of different scenarios.

(1) **Launch Network.** We launch the *simulated* network using an SDR for the eNodeB component and a separate computer for the EPC component of the network. In the *commercial* setup, we connect to the eNodeB of a provider and do not run our own base station.
(2) **Connect Phone.** By using a programmable SIM card, we connect the UE to the eNodeB component of the *simulated* network; for the *commercial* network, we use a standard SIM card of the provider. Once the mobile connection is established and all other data channels are disabled, the phone is ready to request websites. The follow three steps happen simultaneously:
   (a) **Iterate Websites.** We iterate a fixed list of the Alexa top 50 websites (cf. Table 3). Depending on the number of iterations, we request each website n times and then proceed to the next entry on the list.
   (b) **Timing.** We define a page load timeout of 20 s after which we proceed to the next website request.
   (c) **Record Traces.** We record each website request and save the raw trace in a database. The custom eNodeB of the private setup allows monitoring downlink and uplink traffic; in the commercial setup we are limited to monitoring downlink transmissions.

3.3 Parameters

We compare the success of different attack setups (①–②, cf. Table 1) and then vary the following parameters to help us understand the influencing factors of website fingerprinting attacks (③–⑥):

③ **Hardware & Software.** We vary the devices used as UE components in the network setup (cf. Table 2). The screen resolution (web page rendering), chipset (baseband implementation), or the OS version have a potential impact. *Question: Do phone characteristics influence the attack success?*

④ **Time.** Depending on the type of website, its content might change over time. Such changes also affect the traffic characteristics and influence the quality of a training data set that was recorded over longer periods. *Questions: How much does time influence the quality of traces? Can the adversary gather traces over a longer period and still use them for an attack?*

⑤ **Obfuscation.** While we conduct the attack on layer-two traffic, we cannot be sure whether application-layer security mechanisms influence the traffic features. *Question: Is the attack still successful if we use Tor for additional application-layer obfuscation?*

⑥ **Network.** Creating an LTE training data set induces a higher measurement overhead than conventional recording procedures. The possibility to use WiFi traffic for the training corpus would reduce this overhead. *Questions: Can the adversary mix traffic from different networks and still be successful?*

3.4 Attack

We refer to website fingerprinting classifiers that were previously introduced in the context of Tor.

Classifiers. We focus on three main machine-learning techniques that we find in recent WF approaches, namely, a k-Nearest-Neighbor (k-NN) classifier [47], Support Vector Machines (SVM) [36], and a general neural network. As an initial evaluation step, we compare the performance of all three classifiers. In this and all following experiments, we use the micro average F_1 score that summarizes the *global* number of all results in an experiment:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (1)$$

In all experiments, we use a 20-fold cross validation, i. e., the F_1 score summarizes the average attack performance for 20 random repetitions of one experimental setup. The random selection of traces follows a 80 % training and 20 % testing split.

4 PERFORMANCE BASELINE

In our experiments, we first focus on the comparison of attack setups and then continue with the best performing setup to evaluate a series of use-case scenarios.

4.1 Experiments

Table 1 documents the setups (①–⑥), where we begin with a comparison of three classifiers ① for a first estimation of state-of-the-art attack techniques. We continue with an evaluation of the required measurement effort ② and compare the attack performance for a varying number of traces in the data set. The two initial experiments help us to define a general setup that we use for the analysis of influencing factors (③–⑥). In particular, we address differences by hardware and software ③, the impact of time ④, application-layer obfuscation ⑤, and the effects of different networks ⑥. Our baseline experiments use a data set of 93,490 traces covering website requests recorded in seven months; we cover the Alexa top 50 documented in Table 3 for a closed-world attack.

4.1.1 ① Classifier Comparison. In our first experiment, we compare the performance of three classifiers (k-NN, SVM, Neural Network) on a data set of 60 traces per website of the Alexa top 50; in two iterations, we record website requests with and without browser caches.

Results. We see that the k-NN classifier performs best in the uncached setup, as well as in the more challenging cached setup where website requests are of smaller size because of stored contents. In general, all classifiers perform well and manage to identify websites with a success rate of at least 91 % in the uncached and 78 % in the cached setup. *We continue to use the k-NN classifier in all following experiments.*

4.1.2 ② Recording Effort. Our next experiment targets the required recording effort for an acceptable attack performance, i. e., we measure the improvement of the attack success for an increasing number of traces in the data set.

Results. We find a stronger average improvement of 4 % per step in the range of 5 to 20 traces (each step adds five traces to the data set), which stagnates with an average of 0.3 % for 20 to 60 traces. Considering the overhead of 20 additional measurements, the minimal improvement does not justify the overhead. *We continue the following experiments with data sets of 20 traces per website.*

4.1.3 ③ Hardware and Software. Traffic characteristics not only depend on the underlying network but can also be influenced by the hardware and software of a device. In our experiments, we focus on two aspects: First, we conduct the attack with two different operating systems and, second, we compare the traffic of two identical smartphones. In both cases, we record traces in parallel to limit the effects of transmission characteristics and website contents.

Results. For the comparison of the iOS and Android device, we use an alternative recording procedure in which we control the smartphone through a simulated USB keyboard. In contrast to the Android debug bridge, this method lacks direct feedback for a completed page load, thus, we define a fixed recording duration of 20 s and compare only results of this alternative measurement setup. We achieve $F_1 = 0.696$ for the iOS traffic, and $F_1 = 0.738$ for the reference Android traffic. Both performances fall slightly below other cached classification results, which can be explained by the measurement procedure. Besides, both results are comparable, and we do not see a significant influence of the operating system (or browser). In the second experiment, we record traces with two identical devices in parallel. While different measurement dates or technical issues help to distinguish multiple devices, we do not find significant differences between both data sets recorded in parallel.

4.1.4 ④ Time. Due to changing website contents, we consider time as an important influencing factor and can hinder an attack that uses data sets recorded over a longer period. In the following, we use two different data sets. First, we conduct a long-term experiment with a seven months gap between two recording iterations. Second, we repeat recordings for a continuous period of twelve days and analyze the impact of time. Our long-term data set consists of recordings from March and September 2018; we record the second data set in 12 consecutive days of September 2018.

In an *offset* attack, i. e., the training data originates from one day and does not mix with the testing data from another day. In a *combined* attack, the data sets of multiple recording sessions combine and the test traces originate from one of these recording sessions. While the offset experiments represent a scenario where the adversarial training data set is increasingly old, the combined experiments resemble a case in which recordings happen over a longer period.

Results. In both cases of the *combined* attack, we see a slightly decreased attack success with an average of $F_1 = 0.827$ (uplink and downlink) but still achieve a convincing attack success. In contrast, the *offset* experiments are limited to an average of $F_1 = 0.711$ for twelve days and we see a failing attack with an average of $F_1 = 0.032$ for seven months.

4.1.5 ⑤ Obfuscation. To measure the effects of application layer obfuscation, we setup an `Orbot` [13] as Tor proxy that sends and receives all traffic through a Tor circuit.

Results. Our results show that we experience a lower classification success of $F_1 = 0.532$ when using this additional layer of obfuscation. This drop in performance can be explained with the transmission characteristics of Tor traffic (cf. Figure 2). First, the

Table 1: Experimental Setups and Results

Experiment		Setup				F_1	
ID	**Description**	**Devices**	**Parameter**	**Classifier**	**Cached**	**Uplink**	**Downlink**
①	Classifier	Nexus 5, P9 lite, Moto G4	-	k-NN	Uncached	0.949	0.945
				SVM		0.928	0.928
				NN		0.922	0.919
				k-NN	Cached	0.860	0.815
				SVM		0.822	0.776
				NN		0.842	0.806
②	Recording Effort	Nexus 5, P9 lite, Moto G4	5-20 Traces	k-NN	Uncached	0.849	0.844
			20-60 Traces			0.921	0.933
③	Hardware/Software	iPhone 6s	iOS (USB keyboard)	k-NN	Cached	0.686	0.706
		Moto G4	Android (USB keyboard)			0.751	0.726
		P9 lite A, P9 lite B	Same device	k-NN	Cached	0.835	0.805
④	Time	P9 lite	03/01 - 09/10 Combined	k-NN	Uncached	0.871	0.827
			03/01 - 09/10 Offset			0.037	0.026
			08/29 - 09/10 Combined			0.819	0.790
			08/29 - 09/10 Offset			0.737	0.685
⑤	Obfuscation	Moto G4	Tor	k-NN	Uncached	0.522	0.531
⑥	Network	P9 lite	WiFi Only	k-NN	Uncached	0.946	0.805
			WiFi vs. LTE			0.155	0.123

Tor proxy transmits traffic through three-hop circuits built from relays of the Tor network. Such relays are available in places where users voluntarily offer hardware to contribute to the Tor network, consequently, we find a highly skewed relay distribution towards countries with larger Tor communities. In addition to usual transmission dynamics, this amplifies the effects of varying routes and extends the overall transmission distance from the client to the server and back. Second, transmissions through Tor circuits also affect the endpoint of a connection that eventually connects to the web server. Consequently, we experience different website contents adjusted to different user countries, which increases the diversity of monitored traces.

4.1.6 ⑥ Network. In contrast to standard WF attacks on Tor, we experience a higher measurement overhead for generating a representative database of LTE traces. This is because we cannot use browser automatization, but depend on multiple smartphones to record website requests. An adversary could circumvent this situation if the attack were still successful with, e. g., WiFi training data. We run two simultaneous experiments in which one device connects to the standard LTE network, and the other device fetches websites via a WiFi router.

Results. Beginning with the reference experiment, we see that the attack remains successful with a classification success of $F_1 = 0.946$. This result lets us assume that the metadata information of WiFi traffic (instead of PDCP information we must refer to the frame length (comparable to the PDCP length), absolute and relative timestamp of a packet, and the PCAP frame number) still contains sufficient information to distinguish a set of websites. In the second step, we now train on WiFi traffic and conduct the attack with LTE traffic from the simultaneous recording. The average success of $F_1 = 0.139$ reveals that mixing up transmission protocols does not work out. One reason for this is the ratio between amount of data *sent* and the *num* (number of packets), i. e., $ratio = \frac{sent}{num}$. In the case of WiFi traffic, we find an average ratio of 91.426, which stands in contrast to an average of 1030.139 for LTE traffic. We find another difference in the relation between the bytes and the number

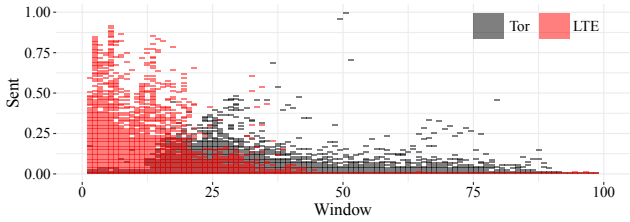

Figure 2: Comparison of Tor and LTE transmission characteristics. Sent data versus transmission duration in comparison for Tor obfuscated and unaltered LTE traffic. We find a wider distribution of Tor transmissions that indicate volatile transmission characteristics for Tor.

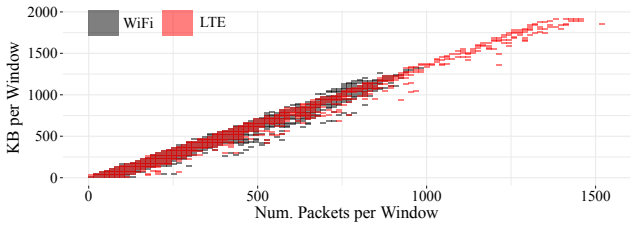

Figure 3: Comparison of WiFi and LTE transmission characteristics. Distribution of ratio between transmission *duration* and the *num* of sent packets for downlink direction.

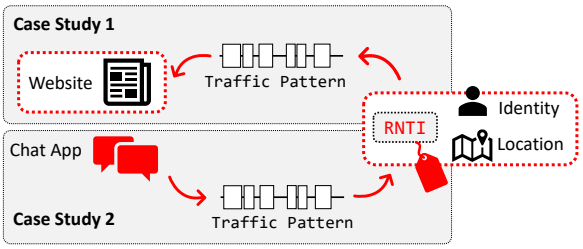

Figure 4: Coherence of commercial network case studies. The results of a targeted user identification/localization (CS2) can amplify the impact of the otherwise untargeted website fingerprinting (CS1) by delivering the identity of a specific user.

of packets sent in a window of 500 ms for all website requests in an experiment(cf. Figure 3). In a direct comparison between WiFi and LTE traffic, we find a maximum transmission count of approximately 1000 packets per window for the WiFi data set, and up to 1500 packets for LTE traffic.

Although the metadata features of WiFi and LTE traffic carry similar information, differences in the amount and size of sent packets confuse the classification process and explain the inability to combine the two data sets. We can explain the different transmission characteristics with the individual duplexing and medium access control mechanisms in both technologies: WiFi uses *time* division multiplexing with a CSMA/CA RTS/CTS[1] scheme to coordinate the transmission between the decentralized clients, while LTE uses *frequency* division duplexing and a centralized way to control the medium access in which the eNodeB allocates resources in a time and frequency domain.[2] In an ongoing downlink transmission, the eNodeB can actively allocate all resources for one client, leading to a large amount of data sent within a window. In contrast, WiFi needs to respect other clients that occupy the medium, thus it is not possible to send a large amount of data within one time frame.

4.2 Summary

We achieve high success rates for a closed-world setup with 50 websites and can assume that LTE layer-two traffic provides sufficient metadata to distinguish browsing traffic reliably. Our experiments cover different parameter setups that address influencing factors like varying website contents over time or application layer obfuscation. Nevertheless, we can only deliver an upper bound for the attack success in a controlled lab environment. Therefore, we continue our evaluation with two case studies in a commercial network setup that demonstrate the feasibility of fingerprinting attacks in a real-world scenario.

5 REAL-WORLD EXPERIMENTS

Figure 4 provides an overview of our real-world case studies. In the following, we introduce the experimental setups of both case studies and discuss their results.

[1]Carrier Sense Multiple Access/Collision Avoidance, Request To Send/Clear To Send
[2]Although the LTE specification also considers time division multiplexing, this setup is not relevant in practice.

5.1 Experimental Setup

While our baseline experiments use a whitebox setting, we now connect the UE to a blackbox commercial LTE network, where we do not deploy our own eNodeB (base station) and EPC (core network), but use a commercial SIM card to connect to one primary provider. Consequently, we cannot record traces in the eNodeB anymore, but need an additional radio-analysis tool [42] to monitor traffic between the UE and the commercial base station. Using this tool, we receive resource allocation information in uplink and downlink direction but can access traffic on the PDCP layer in downlink direction only. This is due to the challenging uplink synchronization between multiple UEs with varying transmission distances to the base station [6, 26].

The experimental setup of our case studies is a closer resemblance of the real-world attack scenario, as the adversary gains access to user data transmissions by passively monitoring traffic in a radio cell with a downlink sniffer. In our experiments, we focus on two attack aims (cf. Section 2.2) to either conduct a passive attack with the goal of fingerprinting websites (CS1), or actively interfere with transmissions to identify specific users from an injected watermark (CS2). We follow the recording procedure of our baseline experiments (cf. Section 4) and record traces for the same candidate set of 50 websites using one smartphone (LG Nexus 5) with an uncached browser setup and train the k-NN classifier on 20 downlink traces. Overall, we record 2772 live network traces with the baseline parameter setup of experiment ②. We limit our live network experiments to this single reference setup, as the recording overhead and cost increases for the commercial network.

5.2 CS1: Real World Website Fingerprinting

We first demonstrate the feasibility of website fingerprinting in a commercial network, which serves multiple active users in addition to the testing phone of our experimental setup. The downlink analysis tool [42] provides us with the DCI scheduling information, allows us to distinguish multiple transmissions through the RNTI, and to derive traffic metadata from the decoded PDCP sub-layer. We recognize the traces of our specific experimental UE using the Qualcomm debug interface and derive the RNTI using a TMSI lookup through SCAT [11]. For the sake of privacy, we do not conduct the

attack on traffic of other active users and, furthermore, do not save these traces in our database.

Experiments and Results. The results of our 20-fold cross-validation reveal an attack success of $F_1 = 0.905$ on the downlink and are comparable to the attack success of the controlled lab environment (92 % to 95 %). During our experiments, an average of 10 active users was present in the commercial cell creating a downlink utilization of approximately 11 %. In comparison, empirical results suggest an average utilization in the range of 25 % (AT&T) up to 58 % (Verizon) [26]. Even though these cells provide a higher load factor, we still expect a successful website fingerprinting attack. The reason for this lies in the fact that we use *scheduling* information, which leads to different traffic patterns as soon as a state of congestion is reached within the cell, e. g., at large public events that not necessarily serve as the standard attack scenario.

Conclusion. Our real-world experiments prove the feasibility of website fingerprinting in a commercial network. Hence, we must expect considerable privacy risks for users, as the attack succeeds in the presence of a passive radio adversary and does not depend on the control over multiple layer-three and -four network switches. Respective hardware is affordable at low prices (less than $160) and depends on an open-source software stack implementation. In contrast to conventional website fingerprinting, this leads to an easy entry point for the attack and creates a substantial impact.

5.3 CS2: Traffic Watermarking

In the second case study, we demonstrate the feasibility of a user identification and localization attack based on injected watermarks. The adversary can conduct such an attack to learn the RNTI of a specific user to, e. g., perform a *targeted* website fingerprinting or derive the TMSI for longer lasting tracking attacks [38]. In contrast to the entirely passive attack of our first case study, the injection of watermarks requires an *active* interference of the adversary.

The adversary can learn the desired RNTI by sending distinct traffic patterns (*watermark injection*) to the public identity of a specific user. While monitoring the downlink transmissions of the radio cell simultaneously, he can identify the injected pattern within all other transmissions. Such patterns consist of repeatedly transmitted messages of n bytes length, e. g., instant messages, that create a specific timing pattern in a transmission. We recognize the injected timing pattern through a threshold approach that compares all received frames with the sent pattern.

Experiments. We focus our experiments on two research questions. First, we demonstrate the feasibility of the identification and localization attack within the radio cell of a commercial network. Second, we analyze the robustness of the attack, i. e., we test the recognition rates for different injected watermarks and measure the success rates for a scenario in which the user is located in the monitored cell. In our experimental setup, we use a `LG Nexus 5` with a commercial SIM card as the user's phone and inject the traffic watermark using WhatsApp as an exemplary instant messaging app (we discuss alternative technical solutions for injecting traffic patterns in Section 6.2). The user's phone connects to the radio cell of the SIM card's provider and receives WhatsApp messages that we send to the respective WhatsApp account.

We use AirScope [42] to monitor the downlink traffic of the radio cell, i. e., we receive traces for all active transmissions within the cell and can distinguish connections by their RNTI. In the attack, we repeatedly send 100 B messages consisting of 100 characters ("AAAA...") using the WhatsApp web interface on a separate computer. Encapsulation and encryption extend the initial 100 B of these raw messages, therefore, we monitor the outgoing traffic of the WhatsApp computer and receive the *final* byte pattern of our watermark. Using 0.5 s delays between each message, we create the timing pattern that later helps to identify the injected watermark.

The monitored downlink traffic consists of all transmissions within the radio cell. To recognize the injected watermark, we apply a threshold decision that defines an upper and lower bound for the size of a received message based on the average message size we monitored at the WhatsApp computer. For the decision, we iterate all RNTIs in the cell and count messages that satisfy the defined threshold. In case the number of messages within the threshold matches the original watermark, we handle this as detection.

Results. By applying the threshold recognition mechanism described above, we identify the RNTI of our test phone through the slightly delayed receive pattern (highlighted in black). We know the RNTI of our specific phone through the Qualcomm debug interface of the phone and use this as the ground truth in all experiments to verify that our attack leads to a correct result. The successful detection of our injected watermark serves as a proof-of-work for the intended identification attack.

In a second step, we analyze the robustness of the threshold recognition mechanism. Within a total of 48 repetitions, we record the sent pattern and compare it with the downlink traffic of the commercial network. With an average of 13 active users in the cell, we achieve a true positive detection rate of 88 % and mismatch the watermark in 13 % of cases. While these numbers indicate a convincingly successful detection rate, we leave alternative injection and recognition techniques for future work.

Conclusion. Active watermarking of layer-two traffic allows us to identify and localize a specific user within a radio cell of 1 km to 10 km radius [45]. In comparison to paging attacks, which cover an entire tracking area, the injection of layer-two watermarks allows for a more precise localization.

6 DISCUSSION

In the following, we discuss how a large-scale adversary can increase the attack impact, document the considerations for a real-world implementation of the attacks, and document possible limitations of our setup parameters.

6.1 Large-Scale Adversaries

The attacker model of Section 2.2 focuses on a single adversary monitoring traffic in one specific radio cell. If we extend this assumption to a large-scale adversary or a malicious provider, the impact of the attacks changes. We discuss the consequences of a stronger adversary for both case studies of Section 5.

Large-Scale Attacker Model. Large-scale attacks can target multiple radio cells at the same time. From a technical perspective, the adversary can accomplish this by deploying various sensors, e. g., downlink sniffing tools with appropriate hardware, within

each cell of interest. We argue that this is a realistic scenario, as the required equipment becomes increasingly affordable and a single sensor ranges around $160 [8]. Potential adversaries are law enforcement agencies targeting individuals (identification and localization) or retail centers interested in the browsing behavior of customers (untargeted website fingerprinting).

Malicious providers are another concept of large-scale adversaries. In contrast to the layer-two fingerprinting attacks introduced in this work, a malicious provider can access transmissions on *additional layers* of the protocol stack up to the IP layer (or further in case of no transmission encryption) and, therefore, is not limited to metadata information derived from the PDCP sub-layer. Furthermore, the malicious provider does not depend on additional actions to localize users within a cell and can analyze traffic using deep packet inspection.

Website Fingerprinting. As the large-scale adversary can analyze multiple radio cells simultaneously, he does not only increase the number of covered users but also receives the ability to derive a more diverse set of accessed websites. This ability improves the data set of the adversary, and, eventually, the overall success of the attack. Consequently, data-hungry classification approaches, e. g., deep learning [37], become a possible option for the attack. Furthermore, the adversary can use his extensive knowledge to learn sensitive information like the correlation between browsing behavior and geographical locations.

Identification and Localization. With a deployed sensor network that, e. g., covers the area of a city, the adversary can use the active fingerprinting to track the whereabouts of a user *consistently*.

6.2 Real-World Considerations

Even though a series of countermeasures against identification attacks on LTE exists, we exploit so far unprotected layer-two characteristics. In the following, we discuss the impact of existing countermeasures and introduce more versatile injection techniques for the identification and localization attack.

Existing Countermeasures. Prior work in the context of localization and user identification attacks exploit the paging channel [26, 40]. A frequent and randomized reallocation of all temporary identifiers [16, 40], e. g., the TMSI, can help to mitigate the threat of paging attacks. As we exploit metadata information of layer-two traffic and do not depend on the control channel and the paging channel, such circumvention techniques do not affect our proposed identification and localization attack. More precisely, it remains successful even with continuously updated identifiers. In contrast to privacy-critical features, e. g., the Globally Unique Temporary ID (GUTI) reallocation, the RNTI reallocation policy is *not* part of the specification [1]. Jover [23, 24] demonstrated that real-world networks do not provide sufficient randomness and tracking between radio sessions based on the RNTI is feasible. Further, a study on user tracking suggests that RNTI tracking is even possible when the user moves to another cell (i. e., based on packet sequence numbers or the RNTI reallocation scheme) [2].

Countermeasures against website fingerprinting aim to obfuscate traffic characteristics that otherwise reveal the similarities between monitored website traces [47, 48, 52]. While we see that the general application layer obfuscation of Tor has a significant effect on the attack (cf. Section 4), we still experience a sufficiently high success rate of $F_1 = 0.532$. Targeted countermeasures against website fingerprinting might increase the obfuscation effect. Nevertheless, the implementation of layer-two obfuscation in LTE leads to an unacceptable performance overhead and cannot be considered a realistic option. The LTE radio layer is optimized for performance [26] and additional countermeasures, especially when focused on the specific use case of fingerprinting attacks, would increase the transmission overhead significantly.

Active Fingerprinting. Even though we successfully demonstrated the identification and localization attack using WhatsApp for the injection of traffic patterns, this method introduces a series of limitations. First, repeated signaling through instant messages can raise the conspicuousness of the user and lead to the blocking of the number. Second, it requires one specific application to conduct the attack. As in general all side channels triggering the reception of data are suitable candidates for the active injection of a fingerprint, alternative applications can extend the range and diversity of the injection mechanism. Examples of such alternatives are Facebook messages or the WhatsApp typing notification [40], but also WebSockets or embedded JavaScript offer the required functionality.

6.3 Upcoming 5G Deployment

The upcoming 5G specification brings new security features like IMSI encryption and initial Non-Access Stratum (NAS) message protection. Such features protect against privacy-invading IMSI catchers and the recently proposed TMSI/IMSI cracking attack [20]. In contrast to these improvements on the NAS layer, the layer two of 5G remains similar to LTE. In particular, the use of RNTIs as radio-layer identity or the downlink control information for managing the resource allocation lead to similar transmission characteristics for both generations.

Another important factor for the persisting threat of fingerprinting attacks is the latency and throughput optimization of 5G. More precisely, the high-performance radio layer and the low-latency transmissions of the core network preserve the timing relations of transmissions and do not allow for the high overhead of common traffic obfuscation. *Consequently, we must assume that our attacks remain successful even in the upcoming 5G mobile generation.*

6.4 Experimental Limitations

Our choice of setup parameters limits the findings of our experiments. In particular, the selection of website candidates for the generation of data sets influences the attack success in the lab and commercial network experiments. Furthermore, we limit out attacks to state of the art machine learning techniques but exclude deep learning from our analyses.

Website Candidates. We use the Alexa top 50 as a candidate set for a closed-world attack. This decision leads to a series of restrictions that on the one hand limit the impact of our results, but on the other hand guarantee a comparable performance baseline for future work in this context. While we know that closed world scenarios can hardly resemble a realistic attack situation [25] and even open-world setups remain unrealistic given the overwhelming number of websites in the Internet, such criticism originates from a

context where adversarial fingerprinting looks back on a comparably long history of attack iterations. This means we already know that, e. g., website fingerprinting with Tor traffic, is possible, and open research questions address improvements towards realism or automatization of attacks [37]. On the other hand, we only stand at the beginning of LTE fingerprinting and, for the time being, need to find out whether such attacks are possible on layer-two traffic. That said, referring to the limited and closed-world selection of websites helps us to make these first steps and provide comparability of results. We leave the use of more sophisticated attack techniques and the processing of follow-up questions to future work.

Deep Learning Deep learning offers new opportunities that classical machine learning does not. Unfortunately, such opportunities come at the price of depending on huge data sets that entail a high measurement effort. For example, Rimmer et al. [37] use a closed-world data set consisting of 3.6 Million pages visits and 1200 website classes. In comparison, our data set provides a total of approximately 90,000 traces and requires around 6 h of recording time for a single experiment (20 repetitions for 50 websites). While we can assume that deep learning is possible in general, we cannot satisfy the data set requirements for a reasonable attack performance.

7 RELATED WORK

In the following, we discuss other privacy-invading attacks on LTE and address fingerprinting attacks with alternative use case characteristics.

7.1 Privacy Attacks on LTE

While we are the first to systematically explore LTE layer-two traffic fingerprinting as proposed by Rupprecht et al. [38], we only find related passive radio-layer attacks with the goal of identification and localization. Paging attacks trigger the wake-up procedure of an idle phone, e. g., through a silent SMS, and localize a *specific* user in a tracking area by observing the paging channel [20, 26, 40]. In contrast to their work, we pinpoint users by exploiting hardly protectable metadata rather than layer-three information. As discussed in Section 6.2, countermeasures like a frequent TMSI reallocation can only limit the period in which the adversary can deanonymize the user [16], but do not prevent the attack. In contrast to our proposed attacks, paging takes place on a layer-three control channel, i. e., we do not depend on the third layer TMSI, but use the RNTI as *radio* session identifier.

Assuming RNTI reallocation techniques with sufficient randomness, such countermeasures still cannot prevent against layer-two fingerprinting. This is since the attacks of both case studies learn sensitive information from hardly protectable traffic metadata and the frequent reallocation of identifiers can only limit the identification periods but does not close the exploited attack vectors.

7.2 Fingerprinting Network Traffic

Fingerprinting attacks exploit similarities in network traffic and help to match otherwise concealed traffic and circumvent the protection of end-to-end encryption. In the following, we discuss passive and active traffic analysis attacks from other contexts in comparison with LTE layer-two fingerprinting.

Passive Traffic Analysis. We find a large body of passive traffic analysis attacks in the context of Tor. Passive flow comparison attacks [27, 41] use the metadata of encrypted traffic and apply distance metrics like Mutual Information or Pearson correlation to identify the relation between connection endpoints. Other examples for traffic confirmation are statistical disclosure [9, 10] or traffic analysis at Internet exchange points [35]. Traffic confirmation attacks have a strong attacker model in common, i. e., they depend on a large-scale adversary to cover a sufficient number of nodes in the network. This stands in contrast to the passive radio attacker that reaches all active users of a provider's cell using low-cost hardware and an LTE software stack.

Active Fingerprinting. Active attacks allow the adversary not only to monitor transmissions through the network but also to interfere with traffic to improve the chances for a successful attack. One example of this are watermarking attacks [4, 17, 18, 50], in which the adversary injects remarkable traffic patterns that help to identify transmissions on their way through the network. While these injected patterns increase the attack success, they might reveal the activity of an attacker on the other hand. Other active attacks use coding techniques to add a specific pattern [32] or exploit dependencies within and between transmission protocols [12, 19]. Active fingerprinting attacks are comparable with the identification and localization attack of our second case study. Nevertheless, active fingerprinting, e. g., in context of Tor, helps to match connection endpoints rather than revealing the whereabouts of users.

8 CONCLUSION

We analyzed the impact of LTE layer-two fingerprinting attacks and their feasibility in a commercial network. Our work revealed serious security and privacy issues and provides proof that passive and active fingerprinting attacks are feasible with high success rates around approximately 90 %. The evaluation of influencing factors in a whitebox network setup revealed the convincing performance of state of the art attack techniques for a *website fingerprinting* attack (91 % to 95 %), but also indicates the potential of application layer obfuscation (53 %) as a protection mechanism available by user choice. Two real-world case studies backed up our whitebox experiments and proved the feasibility of traffic fingerprinting in commercial networks. While the commercial network website fingerprinting remained successful for 90 % of web pages, the demonstration of an active watermarking attack further enables the adversary to identify and localize *specific* users in a radio cell. The combination of both attacks amplifies the impact of traffic fingerprinting of LTE layer-two traffic, yet we learned that existing defenses provide no protection and layer-two countermeasures are too expensive. This situation continues in the upcoming 5G specification, where we find similar layer-two functionality.

ACKNOWLEDGMENTS

This work was supported by Intel as part of ICRI-CARS. In addition, this work was supported by the German Federal Ministry of Education and Research (BMBF Grant 16KIS0664 SysKit) and the Franco-German BMBF project BERCOM (13N13741). Furthermore, we thank Software Radio Systems and Ravi Borgaonkar for their advise and support.

REFERENCES

[1] 3GPP. 2009. *Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification.* TR TR36.321. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/ftp/Specs/html-info/36321.htm

[2] 3GPP. 2009. *Rationale and track of security decisions in Long Term Evolution (LTE) RAN / 3GPP System Architecture Evolution (SAE).* TR TR33.821. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/ftp/Specs/html-info/33821.htm

[3] FarhanF M. Aziz, Jeff S. Shamma, and Gordon L. Stüber. 2015. Resilience of LTE Networks Against Smart Jamming Attacks: Wideband Model. In *Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '15)*. IEEE, Hong Kong, China, 1344–1348.

[4] Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. 2013. Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization. In *IEEE Symposium on Security and Privacy (SP '13)*. IEEE, San Francisco, CA, USA, 80–94.

[5] Nicola Bui. 2017. IMDEA's Online Watcher for LTE (OWL) control channel. https://git.networks.imdea.org/nicola_bui/imdeaowl. (2017). [Online; accessed 15-Nov-2018].

[6] Nicola Bui and Joerg Widmer. 2016. OWL: A Reliable Online Watcher for LTE Control Channel Measurements. In *Workshop on All Things Cellular: Operations, Applications and Challenges (ATC '16)*. ACM, New York, USA, 25–30.

[7] Heyning Cheng and Ron Avnur. 1998. Traffic Analysis of SSL Encrypted Web Browsing. (1998).

[8] Crowd Supply. [n. d.]. LimeSDR Mini. https://www.crowdsupply.com/lime-micro/limesdr-mini. ([n. d.]). [Online; accessed 15-Nov-2018].

[9] George Danezis. 2003. Statistical Disclosure Attacks: Traffic Confirmation in Open Environments. In *Security and Privacy in the Age of Uncertainty: IFIP TC11 International Conference on Information Security (SEC '03)*. Kluwer, Athens, Greece, 421–426.

[10] George Danezis, Claudia Diaz, and Carmela Troncoso. 2007. Two-Sided Statistical Disclosure Attack. In *International Workshop on Privacy Enhancing Technologies (PET '07)*. Springer, Ottawa, ON, Canada, 30–44.

[11] fgsect. 2018. SCAT: Signaling Collection and Analysis Tool. https://github.com/fgsect/scat. (2018). [Online; accessed 15-Nov-2018].

[12] Xinwen Fu and Zhen Ling. 2009. *One Cell is Enough to Break Tor's Anonymity*. Technical Report. Black Hat USA.

[13] Guardian Project. [n. d.]. Orbot: Tor for Android. https://guardianproject.info/apps/orbot/. ([n. d.]). [Online; accessed 15-Nov-2018].

[14] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. 2009. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier. In *ACM Workshop on Cloud Computing Security (CCSW '09)*. ACM, Chicago, IL, USA, 31–42.

[15] Andrew Hintz. 2002. Fingerprinting Websites Using Traffic Analysis. In *International Workshop on Privacy Enhancing Technologies (PET '02)*. Springer, San Francisco, CA, USA, 171–178.

[16] Byeongdo Hong, Sangwook Bae, and Yongdae Kim. 2018. GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier. In *Network and Distributed System Security Symposium (NDSS '18)*. Internet Society, San Diego, CA, USA.

[17] Amir Houmansadr and Nikita Borisov. 2011. SWIRL: A Scalable Watermark to Detect Correlated Network Flows. In *Network and Distributed System Security Symposium (NDSS '11)*. Internet Society, San Diego, CA, USA.

[18] Amir Houmansadr and Nikita Borisov. 2013. The Need for Flow Fingerprints to Link Correlated Network Flows. In *International Symposium on Privacy Enhancing Technologies Symposium (PETS '13)*. Springer, Bloomington, IN, USA, 205–224.

[19] Amir Houmansadr, Chad Brubaker, and Vitaly Shmatikov. 2013. The Parrot Is Dead: Observing Unobservable Network Communications . In *IEEE Symposium on Security and Privacy (SP '13)*. IEEE, San Francisco, CA, USA, 65–79.

[20] Syed Rafiul Hussain, Mitziu Echeverria, Omar Chowdhury, Ninghui Li, and Elisa Bertino. 2019. Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information. (2019).

[21] Rob Jansen, Marc Juarez, Rafael Galvez, Tariq Elahi, and Claudia Diaz. 2017. Inside Job: Applying Traffic Analysis to Measure Tor from Within. In *Network and Distributed System Security Symposium (NDSS '17)*. Internet Society, San Diego, CA, USA.

[22] Roger Piqueras Jover. 2013. Security Attacks Against the Availability of LTE Mobility Networks: Overview and Research Directions. In *International Symposium on Wireless Personal Multimedia Communications (WPMC '13)*. IEEE, Atlantic City, NJ, USA.

[23] Roger Piqueras Jover. 2016. LTE Security and Protocol Exploits. http://rogerpiquerasjover.net/ShmooCon_talk_final_01162016.pdf. (Jan. 2016).

[24] Roger Piqueras Jover. 2016. LTE Security, Protocol Exploits and Location Tracking Experimentation with Low-Cost Software Radio. *arXiv (1607.05171)* (2016). arXiv:1607.05171 http://arxiv.org/abs/1607.05171

[25] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. 2014. A Critical Evaluation of Website Fingerprinting Attacks. In *ACM Conference on Computer and Communications Security (CCS '14)*. ACM, Scottsdale, AZ, USA, 263–274.

[26] Denis Foo Kune, John Koelndorfer, Nicholas Hopper, and Yongdae Kim. 2012. Location leaks on the GSM air interface. In *Network and Distributed System Security Symposium (NDSS '12)*. Internet Society, San Diego, CA, USA.

[27] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. 2015. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. In *USENIX Security Symposium (USENIX '15)*. USENIX Association, Washington, D.C., USA, 287–302.

[28] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew Wright. 2004. Timing Attacks in Low-Latency Mix Systems. In *International Conference on Financial Cryptography (FC '04)*. Springer, Key West, FL, USA, 251–265.

[29] Marc Liberatore and Brian Neil Levine. 2006. Inferring the Source of Encrypted HTTP Connections. In *ACM Conference on Computer and Communications Security (CCS '06)*. ACM, Alexandria, VA, USA, 255–263.

[30] Marc Lichtman, Roger Piqueras Jover, Mina Labib, Raghunandan Rao, Vuk Marojevic, and Jeffrey H. Reed. 2016. LTE/LTE-A Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation. *IEEE Communications Magazine* 54, 4 (April 2016), 54–61.

[31] Marc Lichtman, Jeffrey H. Reed, T. Charles Clancy, and Mark Norton. 2013. Vulnerability of LTE to Hostile Interference. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP '13)*. IEEE, Austin, TX, USA, 285–288.

[32] Zhen Ling, Junzhou Luo, Wei Yu, Xinwen Fu, Dong Xuan, and Weijia Jia. 2009. A New Cell Counter Based Attack Against Tor. In *ACM Conference on Computer and Communications Security (CCS '09)*. ACM, Chicago, IL, USA, 578–589.

[33] Stig F. Mjølsnes and Ruxandra F. Olimid. 2017. Easy 4G/LTE IMSI Catchers for Non-Programmers. In *Mathematical Methods, Models, and Architectures for Computer Network Security (MMM-ACNS '17)*. Springer, Warsaw, Poland, 235–246.

[34] Steven J. Murdoch and George Danezis. 2005. Low-Cost Traffic Analysis of Tor. In *IEEE Symposium on Security and Privacy (SP '05)*. IEEE, Oakland, CA, USA, 183–195.

[35] Steven J. Murdoch and Piotr Zieliński. 2007. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In *International Workshop on Privacy Enhancing Technologies (PET '07)*. Springer, Ottawa, ON, Canada, 167–183.

[36] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. 2018. Website Fingerprinting at Internet Scale. In *Network and Distributed System Security Symposium (NDSS '16)*. Internet Society, San Diego, CA, USA.

[37] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. 2018. Automated Website Fingerprinting through Deep Learning. In *Network and Distributed System Security Symposium (NDSS '18)*. Internet Society, San Diego, CA, USA.

[38] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2019. Breaking LTE on Layer Two. In *IEEE Symposium on Security and Privacy (SP '19)*. IEEE, San Francisco, CA, USA.

[39] Sanjole Inc. 2012. *WaveJudge 4900A LTE analyzer.* Technical Report.

[40] Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi, and Jean-Pierre Seifert. 2016. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. In *Network and Distributed System Security Symposium (NDSS '16)*. Internet Society, San Diego, CA, USA.

[41] Vitaly Shmatikov and Ming-Hsiu Wang. 2006. Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses. In *European Symposium on Research in Computer Security (ESORICS '06)*. Springer, Hamburg, Germany, 18–33.

[42] Software Radio Systems. [n. d.]. AirScope. http://www.softwareradiosystems.com/products/. ([n. d.]). [Online; accessed 15-Nov-2018].

[43] srsLTE. 2018. Open source SDR LTE software suite. https://github.com/srsLTE/srsLTE. (2018). [Online; accessed 15-Nov-2018].

[44] Qixiang Sun, Daniel R. Simon, Yi-Min Wang, Wilf Russell, Venkata N. Padmanabhan, and Lili Qiu. 2002. Statistical Identification of Encrypted Web Browsing Traffic. In *IEEE Symposium on Security and Privacy (SP '02)*. IEEE, Berkeley, CA, USA, 19–30.

[45] Z.H. Talukder, S.S. Islam, D. Mahjabeen, A. Ahmed, S. Rafique, and M.A. Rashid. 2013. Cell Coverage Evaluation for LTE and WiMAX in Wireless Communication System. *World Applied Sciences Journal* 22, 10 (Jan. 2013), 1486–1491.

[46] The Tor Project. [n. d.]. The Onion Router. https://www.torproject.org. ([n. d.]). [Online; accessed 15-Nov-2018].

[47] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. 2014. Effective Attacks and Provable Defenses for Website Fingerprinting.. In *USENIX Security Symposium (USENIX '14)*. USENIX Association, Washington, D.C., USA, 271–286.

[48] Tao Wang and Ian Goldberg. 2017. Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks. In *USENIX Security Symposium (USENIX '17)*. USENIX Association, Washington, D.C., USA, 1375–1390.

[49] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. 2005. Tracking Anonymous Peer-to-Peer VoIP Calls on the Internet. In *ACM Conference on Computer and Communications Security (CCS '05)*. ACM, Alexandria, VA, USA, 81–91.

[50] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. 2007. Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems. In *IEEE Symposium on Security and Privacy (SP '07)*. IEEE, Oakland, CA, USA, 116–130.

[51] Zhanyi Wang. 2015. *The Applications of Deep Learning on Traffic Identification.* Technical Report. Black Hat USA.
[52] Charles V. Wright, Scott E. Coull, and Fabian Monrose. 2009. Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis. In *Network and Distributed System Security Symposium (NDSS '09).* Internet Society, San Diego, CA, USA.

APPENDIX

In the following, we provide additional background information on the experimental setups of this work.

Experimental Setups

In our experiments, we use the devices for recording website traffic as summarized in Table 2. Different screen resolutions or browser versions can influence the rendering of web pages and therefore are a potential influencing factor for transmission characteristics and the success of website fingerprinting.

Table 3 provides an overview of the Alexa website collection at the time of our experiments. Please note that the documented selection of websites can only provide a snapshot and the top 50 pages vary over time.

Mobile Communication Acronyms

DCI Downlink Control Information
eNodeB Evolved NodeB
EPC Evolved Packet Core
GUTI Globally Unique Temporary ID
LTE Long Term Evolution
MAC Medium Access Control
UE User Equipment
NAS Non-Access Stratum
IMSI International Mobile Subscriber Identity
TMSI Temporary Mobile Subscriber Identity
RAP Random Access Procedure
RAR Random Access Response
RLC Radio Link Control
RNTI Radio Network Temporary Identifier
C-RNTI Cell-RNTI, sub-range of the RNTI
PDCP Packet Data Convergence Protocol
SDR Software Defined Radio

Table 2: Specification of Experimental Devices.

Device	Resolution	Chipset	OS	Browser	Release
Motorola Moto G4	1080x1920	Snapdragon 617	6.0.1	Chrome	2016
Huawei P9 Lite	1080x1920	Kirin 650	7.0	Chrome	2013
LG Nexus 5	1080x1920	Snapdragon 800	5.1	Chrome	2013
Apple iPhone 6s	750x1334	Apple A9	12.0	Safari	2015

Table 3: Alexa Top 50 Websites.

1	google.com	18	yandex.ru	35	livejasmin.com
2	youtube.com	19	netflix.com	36	imdb.com
3	facebook.com	20	t.co	37	stackoverflow.com
4	wikipedia.org	21	pornhub.com	38	csdn.net
5	yahoo.com	22	xvideos.com	39	blogspot.com
6	reddit.com	23	ebay.com	40	github.com
7	amazon.com	24	bing.com	41	whatsapp.com
8	twitter.com	25	twitch.tv	42	paypal.com
9	live.com	26	imgur.com	43	wikia.com
10	vk.com	27	msn.com	44	qq.com
11	sohu.com	28	apple.com	45	taobao.com
12	instagram.com	29	wordpress.com	46	craigslist.org
13	sina.com.cn	30	office.com	47	adobe.com
14	jd.com	31	microsoft.com	48	dropbox.com
15	weibo.com	32	ok.ru	49	booking.com
16	360.cn	33	aliexpress.com	50	thestartmagazine.com
17	linkedin.com	34	tumblr.com		