RUHR
UNIVERSITÄT
BOCHUM

RUB

DISSERTATION

Internet Anonymity and Privacy in the Presence of Large-Scale Adversaries

Katharina Siobhan Kohls, M.Sc.
Ruhr-Universität Bochum
Januar 2019

RUHR
UNIVERSITÄT
BOCHUM

RUB

Internet Anonymity and Privacy in the Presence of Large-Scale Adversaries

DISSERTATION

zur Erlangung des Grades eines Doktor-Ingenieurs
der Fakultät für Elektrotechnik und Informationstechnik
an der Ruhr-Universität Bochum

vorgelegt von

Katharina Siobhan Kohls, M.Sc.
geboren in Berlin

Bochum, 9. Januar 2019

Tag der mündlichen Prüfung: 5. Juli 2019

Gutachter:
Prof. Dr. Thorsten Holz, Ruhr-Universität Bochum

Zweitgutachter:
Prof. Dr. Christina Pöpper, New York University Abu Dhabi

Drittgutachter:
Prof. Dr. Claudia Diaz, Katholieke Universiteit Leuven

Abstract

The increasing accessibility of the Internet creates a space for sharing information and services without technology imposing any moral rules. Regardless of private or commercial interests, a broad spectrum of views judges what might or might not stay within acceptable limits. We recognize the same contradiction in the motivation of authorities that ranges from legal persecution to the censorship efforts of oppressive regimes.

Since the democratization of the Internet, scientific work is dedicated to protecting sensitive information or emphasizing that the same is at risk. Because of its large active user base of approximately three million daily users, the Tor anonymity system has become an essential part of this research. However, the scientific achievements of recent years have led to a critical point. The emerging threat of large-scale adversaries shifts the once only theoretical worst-case attacker models closer towards reality. Furthermore, new attack concepts like deep learning demand solutions that exceed our current technical capabilities. At the same time, scientific defenses fail to find an acceptable compromise for the performance and security trade off in the real world. *A large gap separates science and reality*, and in this ubiquitous security arms race, offensive work starts with an advantage.

In this thesis, we aim to reduce the gap between both worlds. In the first part, we analyze how traffic analysis attacks affect the *anonymity* of users. As a starting point, we introduce the comparison framework *DigesTor* to establish performance benchmarks for otherwise incomparable passive traffic analysis attacks on Tor. For the first time, these benchmarks allow us to analyze countermeasures in the presence of state of the art attacks: We demonstrate that low-latency mixing hinders the success of attacks but increases the end-to-end delays resulting in performance impairments. Geographical avoidance is an alternative to direct traffic obfuscation, i. e., it allows to circumvent the areas of nation-state adversaries. To this end, we introduce the prototype *TrilateraTor* as a comprehensive avoidance system that provably bypasses untrusted areas, overcomes fundamental security issues, and considers demanding real-world requirements. Finally, we consider the increasingly mobile character of the Internet and transfer traffic analysis attacks to the context of mobile networks. With different active and passive fingerprinting attacks, we demonstrate how the same contradictory security and performance trade off leads to open attack vectors in the current and upcoming

mobile generations. In the second part of this thesis, we relax the requirements for anonymity and analyze how users can keep control over their *online footprints*. Through a simulation study and the prototype implementation of *Neuralyzer*, we demonstrate how user-driven access heuristics add a reliable new perspective to recapture the control over online data.

Overall, we find inevitable performance requirements of real-world systems that create a profoundly unbalanced starting position in the arms races between offensive and defensive research. With the emerging threat of large-scale adversaries, this work aims to identify *realistic* research perspectives to achieve technical equality in this contradictive context.

Kurzfassung

Die zunehmende Verfügbarkeit des Internets erschafft einen Raum für Informationen und Dienste, wobei die Technologie selbst dabei keinen ethischen Rahmen definiert. Unabhängig privater oder kommerzieller Interessen entsteht so ein breites Meinungsspektrum über die Grenzen des Legitimen. So entstehende Widersprüche finden wir auch in der Motivation von Behörden wieder, deren Aktivitäten von der Gesetzesverfolgung bis hin zur autoritären Zensur reichen.

Seit das Internet zunehmend im privaten wie auch kommerziellen Bereich etabliert ist, widmet sich die Forschung unter anderem dem Schutz sensibler Informationen und der Aufdeckung möglicher Gefahren für diese. Mit über 3 Millionen täglichen Nutzern stellt dabei das Anonymitätssystem Tor ein wichtiges Ziel dar, wobei die Fortschritte der letzten Jahre nun zu einem entscheidenden Punkt hinführten. Einerseits stellen großskalige Angreifer eine aufkommende Bedrohung dar, die ehemals nur theoretische worst-case Angreifermodelle zu einem realistischen Problem machen. Andererseits erfordern neue Konzepte wie beispielsweise Deep Learning Angriffe technisch anspruchsvolle Gegenmaßnahmen. Gleichzeitig schränkt die Ermittlung eines akzeptablen Kompromisses zwischen den Sicherheits- und Leistungsanforderungen der echten Welt die Entwicklung defensiver Maßnahmen stark ein. So finden wir gegenwärtig eine *große Diskrepanz zwischen der wissenschaft-lichen und der echten Welt*, die ungleiche Ausgangsbedingungen im Sicherheitswettrüsten erzeugt.

Ziel dieser Arbeit ist es, diese vorherrschenden Diskrepanzen zu reduzieren. Dazu analy-sieren wir, wie weit Verkehrsanalyseangriffe die Anonymität von Nutzern gefährden können. In diesem Zusammenhang stellen wir *DigesTor* vor, ein System das bisher nicht vergleichbare Angriffskonzepte gegenüberstellt und den Erfolg von Mixing als Gegenmaßnahme misst. *TrilateraTor* stellt ein alternatives defensives Konzept dar, bei dem kritische geographische Bereiche umgangen werden. *TrilateraTor* ermöglicht eine Abstufung des Schutzes, der von einem beweisbarer Sicherheit bis hin zu minimalen Leistungseinschränkungen reicht, und berücksichtigt die anspruchsvollen Netzwerkcharakteristika von Tor. Abschließend adressieren wir die Bedeutung des mobilen Internets und übertragen Verkehrsanalyseangriffe in diesen neuen Kontext. Durch aktive und passive Angriffe zeigen wir, dass der gleiche Widerspruch zwischen Sicherheit und Leistung auch in mobilen Netzwerken

einen Angriffsvektor für bestehende und zukünftige Generationen erschafft. Im zweiten Teil dieser Arbeit lockern wir die strikten Anforderungen der Anonymität und fokussieren uns auf die Privatheit persönlicher Daten. In diesem Kontext stellen wir *Neuralyzer* vor, ein System das durch nutzerbasierte Metriken eine flexible Kontrolle über Online Daten ermöglicht, um das *Recht auf digitales Vergessen* zu stärken.

Insgesamt stellen wir fest, dass die Leistungsanforderungen der realen Welt die Entwicklung defensiver Mechanismen stark einschränken. Gleichzeitig sind offensive Maßnahmen weitaus seltener dadurch einge-schränkt, was zu unausgeglichenen Startbedingungen im Sicherheitswettrüsten führt. Um dabei der zunehmenden Bedrohung durch großskalige Angreifer entgegenzuwirken, zielt diese Arbeit auf die Entwicklung *realistischer* neuer Forschungsperspektiven ab.

Acknowledgements

First, I would like to express my sincere gratitude to my advisor Prof. Dr. Christina Pöpper for her guidance and support during her time in Bochum and remotely sent from Abu Dhabi. I feel greatly privileged for receiving the opportunity to freely work on a topic of my choice. Likewise, I thank Prof. Dr. Holz for sharing his thoughts in numerous discussions and his time during hours of reviewing work. At the Information Security Group and the Chair for Systems Security I found my scientific home and the chance to work together with many motivated and ingenious colleagues.

In particular, I would like to thank (in chronological order) Kai–Jänner–Jansen for endless nonsense and sports, sports, sports (Maeckes - Partykirche) and Maximilian–Maxi–Golla for *always* helping out (Dire Straints - Money for Nothing). We started together, we finished together, Schweinchen forever. I would like to thank Lea–XOR–Schönherr for knitting at the congress (Red Hot Chili Peppers - Can't Stop), Theodor–Theo–Schnitzler for his humor (Galactic Empire - Cantina Band), Jan–Janni–Wiele for DC-DC-Boosters and all that jazz (Look Mum No Computer - Groundhog Day), Florian–Fabi–Farke for his long sides that fit numerous cream cakes (Porcupine Tree - Lazarus), and Philipp–Freddy–Markert for bringing the fame back to Bergkamen (SXTN - Von Party zu Party).

I would like to thank my parents Erika and Harald Kohls for always being there, for supporting every step I have taken, and for never questioning that a child can grow from disassembling every device in reach to studying computer science. You are my super duper cool role models and gave me the confidence to just try things (Madsen - Kompass). I thank my brother Moritz–Mori–Kohls for simply being the smartest person in the world (Tom Bowness - Sing to Me). Finally, I would like to thank David–Dave–Rupprecht for sharing his fantastic love. You make the world a better and fluffy place (Phil Collins - True Colors).

Contents

I'm busting up my brains for the words.

— David Bowie

1
Introduction

Contents

1.1. Motivation

The technological advances of the past decades shaped the Internet into a room for sharing information and services without enforcing any ethical rules. This results in a broad moral usage spectrum ranging from idealism to illegality and opinions on what stays within appropriate limits contradict each other. The same contrast makes it hard to draw a straight line between the motivation of authorities that ranges from law enforcement [AA17, MR16] to censorship through oppressive regimes [PEL+17, DSA+11, WL12]. This moral ambiguity challenges a clear and straightforward regulation of how much anonymity and privacy is justifiable on the Internet and advances the security arms race between the involved parties.

In this arms race, the academic community is an essential accelerator for advancing offensive and defensive concepts. This helps to identify open attack vectors in theoretical concepts as well as deployed systems for improving future technological generations. Unfortunately, the fact that scientific assumptions can expand beyond the restrictions of reality creates a gap between the academic and the real world. Therefore, the transfer from a theoretical concept to a realistic implementation is often difficult, and we find some problems solved in science but still unanswered in the real world [DMMK18]. In this work, we focus on two topics that emphasize the impact of this gap.

Protection of User Identities

Anonymous communication systems answer the growing concern of being the target of monitoring or censorship [EY09]. Systems like Tor [SDM04] offer users the tools to protect their online activities by hiding sensitive information like their identities. Such additional protection raises the same initial question for a straight line between the legitimate protection of good intentions and illegal gains [Cla99]. Revealing corrupt activities legitimates the development of new de-anonymization attacks, but cases of Internet censorship through oppressive regimes, on the other hand, argue for additional protection mechanisms. Even research, which should follow a neutral position, does not always sustain appropriate decisions [CMU15]. In this ethically challeng-

ing context, traffic analysis attacks are of particular interest. They deliver the tools to learn sensitive information about critical network nodes [JJG+17] or to connect user identities with the accessed contents [NHM17]. Since an early traffic analysis attack was proposed in 1998 by Cheng et al. [CA98], improving technical capabilities created more and more powerful attack scenarios. Today, we face a situation in which the emerging presence of large-scale adversaries shifts once theoretical worst-case adversaries closer towards reality [NSZ+16]. Meanwhile, the strict performance requirements of the real world restrain the development of new defensive concepts, and we find only a few experimental countermeasures deployed [Per11] in reality. Deep learning attacks emphasize the deeply imbalanced starting positions in the offensive and defensive arms race even more and suggest a scenario that challenges the development of efficient countermeasures further [RPJ+18, NBH18].

With countermeasures being at such a disadvantage, the current situation requires to identify *new perspectives* to support defensive aspects. We find one significant reason for this disadvantage in the diversity of scientific evaluation techniques. More precisely, existing attacks indicate a series of open attack vectors [NBH18, RPJ+18, SEV+16], but fundamentally differ in their experimental setups and analysis approaches. This way, it becomes obvious that traffic analysis attacks pose a considerable threat, but we have no means to compare the performance of the existing attack landscape. Consequently, new countermeasure concepts lack a performance baseline and it becomes impossible to measure their impact in the presence of state of the art. With this situation in mind, we define the first research question.

Research Question 1 *How can we overcome the experimental diversity in traffic analysis attacks to facilitate the development of new countermeasure techniques?*

We identify the strict real-world performance requirements as the second factor to slow down defensive progress. Direct defenses often use expensive obfuscation techniques that hinder the success of traffic analysis attacks at the expense of performance impairments, e. g., by delaying existing traffic or injecting dummy packets. While the resulting overhead can be a remedy

in highly critical scenarios, common use cases require performance-efficient concepts. We take this as a reason to define the second research question.

Research Question 2 *Do alternative countermeasures find a realistic compromise in the security and performance trade off that can serve as a long-term defense?*

Our first two research questions focus on the classical anonymity system Tor. Nevertheless, since its deployment several changes in our information culture, e. g., new mobile generations, also emphasized the relevance of other network infrastructures. In particular, the technical achievement of the mobile Internet opens up new perspectives for essential research directions. As the current and upcoming mobile generations face similar problems with arranging security and performance features, we can expect the same vulnerability to traffic analysis attacks as for Tor. This leads us to the third research question.

Research Question 3 *Can we transfer well-known traffic-analysis attack techniques from anonymity networks to cellular networks with the goal to attack encrypted mobile traffic?*

Overall, our initial three research questions cover the feasibility of traffic analysis attacks and possible countermeasures. Leaving the ethical controversies of anonymous communication behind, we switch to the revocation of online data for the second area of interest.

Protection of User Data

When relaxing the security requirements of anonymous communication, we find a situation where users are not concerned about revealing their identity, but need a certain degree of protection against large adversaries. Instead of nation-states, we now focus on large service providers like Google or Facebook and the challenge of handing users control over their online data. Even though the services of such providers are an integral part of our everyday lives, their ubiquitous presence lets users reconsider how to handle their data [TVSS18]. Often, the sheer amount of what we manage online makes it difficult to remember all information that once was

shared [MMG+16]. When confronted with older uploads, users often decide to delete these files [CSMK15]. Nevertheless, large companies remain non-transparent, and a missing control interface leaves us in the dark about the actual whereabouts of our data. Official regulations, e. g., the General Data Protection Regulation (GDPR), help to create awareness for the right to erasure. For the academic world, the current situation creates the opportunity to provide users with the technical means to manifest their new powers. With the goal of building systems that represent real-world needs, we define the final research question.

Research Question 4 *How can we offer users tools to control their online data in the presence of large service providers?*

Overall, current and future research must consider the emerging presence of large-scale adversaries and target systems that leave users with an option to protect themselves. Nevertheless, such advances cannot be made without considering the strict restrictions of the real-world trade off between security and privacy. New approaches must support the consideration of both worlds without sacrificing the unique freedoms of science. In this situation, we rely on technological neutrality so as not to be hampered by ethical controversies. Research has the potential to provide space for the appropriate range of the ethical spectrum.

1.2. Topic and Contributions

In this thesis, we analyze how large-scale adversaries impact anonymity and privacy enhancing technologies. In this context, we focus on the challenging compromise between the real and the scientific world. Our work aims to reduce the existing gap between both worlds by answering the initial research questions. To this end, we analyze the performance of offensive and defensive techniques in different real-world systems and introduce prototype implementations to cover shortcomings in existing work (cf. Figure 1.1). In the following, we provide a brief introduction to each topic.

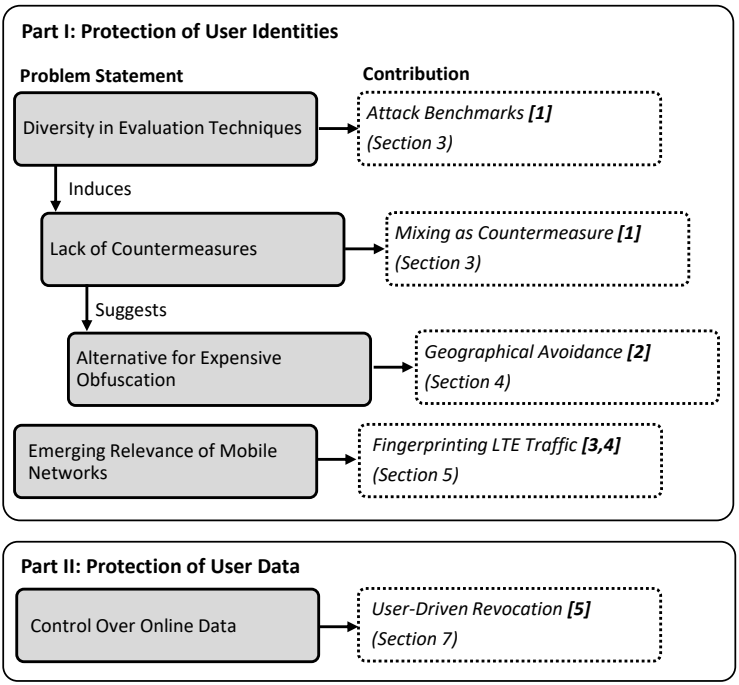


Figure 1.1. Contributions of this work. Part I addresses the protection of user identities; Part II addresses the protection of user data.

1.2.1. Protection of User Identities

In the first part of this work, we focus on anonymity and the protection of *user identities*. Traffic analysis attacks challenge the goal of anonymity and enable an adversary to derive sensitive information from encrypted traffic. While the current lack of countermeasures leaves well-known attack vectors open, today's security and performance trade off affects future systems as well. In the following, we focus on the impact of traffic analysis attacks and overview our contributions.

Attack and Countermeasure Performance

With a large active user base, the Tor anonymity system is a prominent example for a real-world technology that offers users the chances to hide their identities on the Internet. Because of this prominence, Tor is a reasonable target for the development of new scientific attack concepts, and we find a large body of prior work confirming this. Although this high effort might imply that any security issues in Tor are already well understood, the results of this research are difficult to compare. A high diversity in evaluation techniques, e. g., the use of different experimental setups, leads to individual estimates of the attack performances and we are unsure about the actual impact of the current state of the art. The consequences of this are two-fold. First, the lack of a comparison option prevents the creation of global *performance benchmarks*. Second, these missing performance benchmarks hinder the evaluation of new countermeasures, as they should be designed to defend against all relevant attacks.

As an appeal to comparability in security research, we introduce the traffic analysis framework *DigesTor*. Our framework generates representative traffic in a controlled and private Tor network setup that follows the ethical guidelines of Tor research and provides transparency in the design and execution of experiments. The traffic of our network builds the data basis for the evaluation of passive traffic analysis attacks and, in contrast to prior work, enables the comparison of attack performances. *DigesTor* implements a series of state of the art and novel attack techniques that we use for creating the first passive *attack benchmarks*. We use the results of these performance benchmarks for testing the potential of low-latency mixing as a countermeasure to traffic analysis attacks. The results of our experiments reveal how an adversary can use specific knowledge about network setups or applications to improve the attack success. Furthermore, we demonstrate how mixing limits the success of such attacks at the expense of additional transmission delays. *The contributions of DigesTor are joint work with Christina Pöpper.*

Geographical Avoidance

Our mixing experiments revealed that obfuscation helps to hinder the success of an attack, but always comes at the expense of performance impairments. While particularly sensitive situations might justify an additional overhead, this does not apply for everyday use cases. Geographical avoidance is an alternative defensive approach that circumvents adversarial areas. By avoiding the sphere of influence of large-scale adversaries like nation-states, we can overcome the need for expensive obfuscation. While time-based geographical avoidance allows for a provable detection of untrusted connections, real-world network characteristics challenge an easy and uncomplicated deployment of such a system. If not considered, wrong avoidance decisions can lead to severe security and performance issues for the entire Tor user base.

To this end, we assess and analyze the challenges of geographical avoidance and put a specific focus on the unevenly distributed network infrastructure of Tor. In an empirical study, we examine the distribution of relays, the impact of transmission characteristics, and prove that external Geo IP locations are an error-prone source of information. Our findings help to understand all mandatory features that must be considered in the design of a system. We use the assessment of challenges as a starting point to design a new avoidance scheme that overcomes the performance and security issues of existing systems. Furthermore, we address the requirements of a real-world deployment and analyze the performance of *TrilateraTor*, a prototype implementation of our system concept. *The contributions of this work result from a collaboration with Kai Jansen, David Rupprecht, Thorsten Holz, and Christina Pöpper.*

LTE Website Fingerprinting

The increasing establishment of mobile networks opens up new usage perspectives and in particular, enables mobile access to the Internet. In this context, we find the same restricting trade off between security and performance that influences the protection capabilities of current and upcoming mobile generations. One example of this is the communication standard Long Term Evolution (LTE). Similar to the traffic analysis attacks on Tor, LTE layer two traffic provides a comparable set of unencrypted meta data infor-

mation. While this similarity implies a well-known attack vector in the new context of mobile networks, the capabilities of a radio layer adversary add a new perspective to the conventional attack scenario. Rather than depending on the physical access to network components, e. g., routers or Internet exchange points, the radio layer adversary is capable of monitoring wireless transmissions of *all* active users in the cell of a provider.

To assess the impact of traffic analysis attacks on mobile networks, we focus on traffic fingerprinting and analyze its feasibility on LTE layer two traffic. Our experiments include the assessment of a website fingerprinting performance baseline in which we conduct a closed-world attack and examine external influencing factors. In addition to the baseline experiments in a lab environment, we perform two real-world case studies in a commercial network. Our case studies demonstrate that website fingerprinting remains successful in radio cells with multiple active users and that active fingerprinting allows to identify and localize users within a cell. The combination of both attacks enables an adversary to target specific users and serves as a starting point for follow-up attacks. *The contributions of this work result from a collaboration with David Rupprecht, Thorsten Holz, and Christina Pöpper.*

1.2.2. Protection of User Data

In the second part, we switch to the protection of user data. As we face large-scale adversaries once again, we are interested in offering users control over their online data in the presence of service providers like Google or Facebook. Revoking online data is one option to limit the amount of information that we leave on the Internet over time. In the following, we introduce our contributions to the revocation of online data.

Revocation of Online Data

Uploading data to the Internet means handing over private information to an external service provider. Even though such providers offer to delete uploaded data, their external servers remain black boxes with no control interface for the user. This creates uncertainty over the status of data, and

keeping track of all uploaded and shared files becomes increasingly difficult over time. One possible solution to both problems is assigning an expiration date for each file that assures the destruction of information after a particular time. This can be accomplished by encrypting data, uploading the encrypted data, and sharing the key with other users. Unfortunately, we find a lack of flexibility in prior systems that require users to decide a fixed lifetime. Although such fixed lifetimes can solve some of the initial problems, there is still some burden on the user.

To this end, we introduce the revocation system *Neuralyzer* that assigns *user-driven* expiration times to online data. Similar to previous approaches, *Neuralyzer* protects information by uploading only encrypted files and sharing the key in a distributed infrastructure. Nevertheless, one fundamental difference to other systems is the revocation heuristic that reduces the user burden. Instead of depending on pre-defined lifetimes, we update the key material depending on the number of accesses. Different access heuristics allow for, e. g., destroying files after the interest in it dropped over time. During the evaluation of *Neuralyzer*, we analyze the feasibility of different access heuristics and demonstrate its reliability in individual use case scenarios. *The contributions of this work result from a collaboration with Apostolis Zarras, Markus Dürmuth, and Christina Pöpper.*

1.3. List of Publications

This thesis is based on joint academic publications and contains novel and unpublished material. In particular, the proposed work involves the following publications.

[1] **K. Kohls** and C. Pöpper, "DigesTor: Comparing Passive Traffic Analysis Attacks on Tor," in *European Symposium on Research in Computer Security*, ESORICS '18. Barcelona, Spain: Springer, Sep. 2018, pp. 512–530, **CSAW Europe 2018 Finals**.

[2] **K. Kohls**, K. Jansen, D. Rupprecht, T. Holz, and C. Pöpper, "On the Challenges of Geographical Avoidance for Tor," in *Network and*

Distributed System Security Symposium, NDSS '19. San Diego, CA, USA: The Internet Society, Feb. 2019.

[3] D. Rupprecht, **K. Kohls**, T. Holz, and C. Pöpper, "Breaking LTE on Layer Two," in *IEEE Symposium on Security and Privacy*, SP '19. San Francisco, CA, USA: IEEE, May 2019.

[4] **K. Kohls**, D. Rupprecht, T. Holz, and C. Pöpper, "Lost Traffic Encryption: Fingerprinting LTE Traffic on Layer Two," in *Security and Privacy in Wireless and Mobile Networks*, WiSec '19.

[5] A. Zarras, **K. Kohls**, M. Dürmuth, and C. Pöpper, "Neuralyzer: Flexible Expiration Times for the Revocation of Online Data," in *Conference on Data and Application Security and Privacy*, CODASPY '16. New Orleans, LA, USA: ACM, Mar. 2016, pp. 14–25, **Outstanding Paper Award**.

Furthermore, the author contributed to the following publications during the time of this dissertation.

[6] L. Schönherr, **K. Kohls**, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding," in *Network and Distributed System Security Symposium* NDSS '19. San Diego, CA, USA: The Internet Society, Feb. 2019.

[7] **K. Kohls**, T. Holz, D. Kolossa, and C. Pöpper, "Skypeline: Robust Hidden Data Transmission for VoIP," in *ACM Asia Conference on Computer and Communications Security*, ASIACCS '16. Xi'an, China: ACM, May 2016, pp. 877–888.

[8] **K. Kohls** and C. Pöpper, "Poster: Traffic Analysis Attacks in Anonymity Networks," in *ACM Asia Conference on Computer and Communications Security*, ASIACCS '17. Abu Dhabi, UAE: ACM, May 2017.

[9] **K. Kohls** and C. Pöpper, "Poster: Application-Layer Routing Attacks on Tor," in *IEEE Symposium on Security and Privacy*, ASIACCS '17. San Francisco, CA, USA: IEEE, May 2019.

1.4. Outline

The remainder of this thesis is organized as follows. **Part I** addresses traffic analysis attacks. We begin with the preliminaries of this part in **Chapter 2** including required background information. In **Chapter 3** we introduce *DigesTor*, an evaluation framework for the comparison of passive traffic analysis attacks on Tor [KP18]. *DigesTor* overcomes the diversity of scientific evaluation techniques and allows for analyzing the performance of attacks and countermeasures. In **Chapter 4**, we analyze geographical avoidance as an alternative to expensive traffic obfuscation [KJR+19]. From empirical network studies, we introduce the prototype *TrilateraTor* that reliably circumvents untrusted areas. Finally, we transfer traffic analysis attacks to the context of mobile networks and assess the feasibility of traffic fingerprinting on LTE layer two traffic in **Chapter 5** [KRHP19, RKHP19]. In a series of experiments, we demonstrate the threat of website fingerprinting and traffic watermarking in the current and upcoming mobile generations.

Part II addresses the protection of user data in the presence of large service providers. **Section 6** overviews the background and introduces the specific attacker model of revocation systems. In **Chapter 7**, we present *Neuralyzer*, a system that allows to use empirical lifetimes for uploaded data [ZKDP16]. *Neuralyzer* overcomes the shortcomings of prior systems in this context and does not depend on predefined object lifetimes, but triggers the revocation of data through user-driven access statistics.

Finally, we conclude the findings of this work in **Chapter 8** and discuss possible directions for future work in the context of anonymity and privacy preserving systems.

Part I.

Protection of User Identities

A soul in tension that's learning to fly
Condition grounded but determined to try
Can't keep my eyes from the circling skies
Tongue-tied and twisted just an earth-bound misfit

— David Gilmour

2
Traffic Analysis Attacks

Contents

2.1. Motivation

Many human rights declarations contain the concept of privacy and underline its importance for our society [Cla06]. While the origin of privacy as a human right goes back to the last century, we need to adapt the term to the technical challenges of today. While we can assume that users want anonymity and privacy [Cla99], the Internet does not offer this by default, e. g., uses IP addresses that are personal identifiers of users [EY09].

In prior research, many systems were proposed that seek to offer options for anonymity, privacy, or both. They range from early advances in the context of remailers [Cot19] over lightweight network layer implementations [SW14] to overlay networks [CGF10, LBCC+15, SDM04]. All of these systems use elaborate defensive techniques like the injection of cover traffic [FM02, DDTL10] or the establishment of dynamic paths [LBCZ+13, SBS05], but we find only a few of these systems deployed and adopted in the real world. One important reason for this is the challenging compromise between security and performance [DMMK18], as systems must find an economical and efficient way to serve applications like web browsing. To date, we find that an inevitable focus on performance continues to leave long-known attack vectors open.

Traffic analysis attacks exploit these attack vectors and allow an adversary to learn sensitive information from meta data in encrypted traffic. Such side channels emerge from the low-latency transmissions of performance-centric systems, as the instant forwarding of packets maintains their timing relations and creates distinct patterns for different streams. Although countermeasures like random delays [DSS06] help to disrupt such individual traffic patterns, they also impair the performance of a system. Consequently, we find this side channel in many networks that had to give up specific security measures in favor of more efficient transmissions.

From a technical perspective, traffic analysis attacks apply metrics to detect similarities between transmissions. While we can assume that these technical capabilities of adversaries improved over time, the operational effort is a limiting factor to the success of an attack. Attacking a distributed network for the de-anonymization of users requires monitoring traffic at multiple nodes at the same time, as otherwise, the related transmissions are not

part of the adversarial data set. Although prior work often adopted the *theoretical* worst-case concept of a global adversary, we now face an emerging presence of large-scale adversaries that shifts theory further towards reality. Today, we find nation-state adversaries with comprehensive access to nodes in the network infrastructure [NSZ+16] and the ability to manipulate routes in their favor [SEV+16]. On the other side, current countermeasures lack efficiency and fail to find a compromise between security and the required performance. Advances in the context of deep learning attacks emphasize the advantage of adversaries further and predict vulnerabilities with a long-term perspective.

In this chapter, we document the technical background of anonymity systems using the example of Tor and define anonymity in the context of traffic analysis attacks. Furthermore, we overview relevant related work and determine the attacker model.

2.2. Background

Tor is the most prominent example of a deployed and adopted anonymity system, and its active user base of more than 2 Mio. daily users [The19d] makes it a valuable research target. Several iterations of updates and the open source character have improved Tor over time, and today we find a mix of some of the original security concepts and new features in response to technological progress. In the following, we document the technical background of Tor and discuss essential security features. Furthermore, we introduce different attack and countermeasure concepts and define the terms of anonymity and privacy in this context.

2.2.1. Tor

The first generation of onion routing systems was proposed and discussed between 1996 and 2001 [GRS96, SGR97, SRG00, STRL01] including security features for the anonymization of TCP-based applications like web browsing or instant messaging. In this early state, the Onion Routing project already provided many of the core features we still find in the current Tor version,

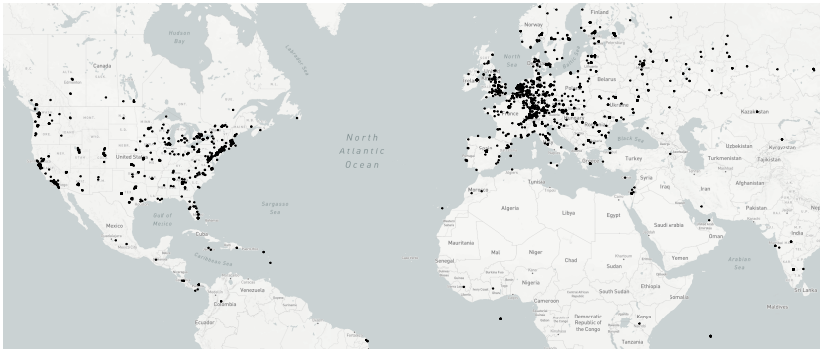


Figure 2.1. Snapshot of Tor infrastructure. Worldwide distribution of 6042 Tor relays as of June 2018 demonstrates the skewed relay distribution with the majority of nodes provided in the Europe and North America.

e. g., circuit connections through multiple dedicated nodes, a padding mechanism that creates fixed-length data cells, or layered encryption for each hop of the circuit. In 2004, Dingledine et al. introduced the second generation of onion routing to overcome some of the initial limitations [SDM04]. The authors introduced different security and performance features to add perfect forward secrecy or congestion control. In contrast to network layer systems [SW14, CAB+15, HKP+12], Tor works on the application layer and is an overlay to the existing network infrastructure of the Internet. We define essential technical characteristics of the Tor protocols and infrastructure and discuss the design decisions that create the system's current security features in the following.

Technical Characteristics

Tor establishes connections through so-called circuits that consist of three dedicated relays, which are voluntarily operated nodes building the central infrastructure of Tor. As the availability of relays depends on the willingness of users to contribute to the network, the distribution of resources correlates with the number of Tor supporters in an area (cf. Figure 2.1). Besides these relays, ten distributed consensus nodes keep track of the network status on a

regular basis and provide information about all available nodes. Tor clients access this information and use it to select relays for the circuit establishment. In the following, we specify the circuit establishment procedure, explain the security relevance of guard nodes, and introduce characteristics of the path selection algorithm.

Circuit Establishment. Three-hop circuits consist of an entry guard connecting to the user, a middle relay connecting the entry and exit, and an exit relay that connects to the endpoint. On the start up of the Tor client, it creates a series of ready-to-use circuits along with several internal maintenance circuits. These pre-built circuits already passed through the cryptographic handshakes that establish individual keys for the different hops of the circuit, i. e., pairwise between the client and entry, client and middle, and client and exit. Circuits remain available for a defined lifetime of 10 minutes.

Preparing circuits as part of the client bootstrap procedure leads to a first important security feature. The circuit build depends on multiple cryptographic handshakes between the client and all three relays that add a computational burden to the network. The key establishment follows the ordering of relays in the circuit and begins with the handshake between the client and the entry relay. After the first handshake finishes, the second key establishment between the client and middle relay follows the first hop of the circuit, i. e., Tor exchanges messages through the entry relay. The final handshake between the client and exit relay follows this principle and uses the entry and middle as intermediate hops to maintain the separation between the endpoints of the circuit. To limit the overhead through these handshakes, the Tor client can reuse circuits for different TCP streams. Nevertheless, sharing one circuit for multiple streams allows a *malicious* exit node to link connections through the consistent `circuit ID` in all of the TCP streams. If one stream carries personal identifiers, e. g., the user's IP address for BitTorrent traffic, the malicious exit can match this sensitive information to the remaining streams with the same circuit ID and relate all traffic to the same origin. To limit the threat of linkable TCP streams, the Tor client marks circuits as dirty upon their initial usage and avoids reusing them after a life-

time of 10 min. We emphasize that the above problem arises from the fact that Tor's network infrastructure consists of *voluntarily* operated nodes, i. e., an adversary can exploit this fact and establish compromised nodes.

Guard Nodes. The first node of a Tor circuit knows the identity of users and therefore threatens their anonymity. Assuming a low rate of compromised entry relays, users increase their risk of incorporating such a malicious node in their circuit when switching through a high number of different relays for the creation of new circuits. Tor avoids this by selecting one fixed *entry guard* and creating individual sets of additional guards as backup candidates. Such guard sets update nodes after a defined lifetime of several months [EBA+12].

Using a dedicated guard set moves the initial problem of using compromised entry relays. Users with a guard set that consists of good entry guards improve their situation significantly and reduce their chances of picking malicious nodes. On the other side, guard sets including compromised nodes provide a security level that is comparable with the initial situation of having no guard sets in use. This limits severe security issues to an unlikely worst case selection of nodes.

Path Selection. Selecting relays for the circuit buildup influences the security and the performance of a connection. In the original Tor design, nodes were picked randomly to withstand relay adversaries [STRL01, RP17]. Unfortunately, the voluntarily operated relays offer different bandwidth features, which leads to poor and varying performance rates for different connections. Since version `0.0.8rc1` (Aug 2004), relays informed about their bandwidth to allow preferring higher performance nodes in a circuit. In the later version `0.2.1.17-rc` (Jul 2009), the system moved from using the local bandwidth information of relay descriptors towards referring to the values in the consensus. Handling this information in the central consensus allows for additional bandwidth estimates that improve the accuracy. Since version `0.2.2.6-alpha` (Dec 2009), active measurements of a bandwidth scanner assign proportional weights for relays and achieve good load balancing and anonymity [MW08, WTBS13].

While the bandwidth scanner improves the overall performance of circuits, additional selection constraints help to overcome possible security issues. One example of this are relays that fall into the same IP address space, or are affiliated with the same organization. If such an organization is adversarial and intends the monitoring and analysis of traffic, this increases the chances for an end-to-end confirmation significantly. Therefore, if more than one node falls into the same `/16` IP address range, these additional nodes are excluded from the circuit establishment. Furthermore, relay providers with multiple nodes can declare so-called families for avoiding connections that traverse one organization twice.

Security

Tor defends against a set of known attack vectors, e. g., it uses onion encryption to make the byte patterns of packet payloads unlinkable, or it applies padding to limit the information leak from packet lengths [GRS96]. However, to satisfy the requirements of everyday use cases like web browsing, Tor must abandon security measures in favor of performance features. In the following, we discuss the initial security decisions of the second generation of onion routing as introduced by Syverson et al. [SDM04].

In contrast to other overlay systems [FM02, GRPS03], Tor does not consider a *global* passive adversary but limits the attacker model to local access to nodes and links. This leads to a fundamental difference in the impact of a traffic analysis attack: Both adversaries can observe network traffic and have the option to either remain in a state of passive monitoring or switch to an active interference including the injection of packets or the modification of existing traffic. With a *global* access to the network, the adversary can use monitoring capabilities to conduct an end-to-end confirmation attack that matches the client and server of an arbitrary connection. In contrast, Syverson et al. consider non-global attacks that help to narrow down specific connections but do not allow for the de-anonymization of users.

Within this attacker model, Tor is restricted by real-world performance requirements. We find one important example of such performance-oriented design decisions in low-latency transmissions. As we know from prior work in

the field of anonymous remailer systems [Cot19, DDM03], low-latency communication without any injection of dummy packets preserves the timing characteristics of network traffic and, therefore, leaves sufficient meta data information for traffic analysis attacks. Nevertheless, the authors refrain from deploying traffic obfuscation mechanisms: "Onion Routing originally called for batching and reordering cells as they arrived, assumed padding between ORs (Onion Router), and in later designs added padding between onion proxies (users) and ORs [SGR97, GRS96]. Trade offs between padding protection and cost were discussed, and traffic shaping algorithms were theorized [STRL01] to provide good security without expensive padding, but no particular padding scheme was suggested. Prior research [ADS03] and deployment experience [BGS19] suggest that this level of resource use is not practical or economical, and even full link padding is still vulnerable [LRWW04]. Thus, until we have a proven and convenient design for traffic shaping or low-latency mixing that improves anonymity against a realistic adversary, we leave these strategies out." While these decisions were made over a decade ago, we still find none of the above mechanisms implemented in the current version of Tor.

2.2.2. Attacks

As a consequence of the initial design decisions of Syverson et al., we find a large body of attacks that exploit the meta data side channel of Tor's low latency transmissions. On the other side, only a few countermeasures were proposed that seek to overcome the economic limitations of traffic obfuscation. In the following, we introduce the main offensive and defensive concepts of prior work.

Traffic Analysis Attacks

Traffic analysis attacks enable an adversary to learn sensitive information from the meta data of encrypted transmissions. In the context of Chapters 3 and 4, we focus on the impact of *end-to-end confirmation* that allows to match the endpoints of a connection to de-anonymize users. For the experiments of Chapter 5, we address *website fingerprinting* attacks in which the adversary

derives the accessed websites of a connection. We introduce the general concept and problem statement of both attack classes as follows.

End-to-End Confirmation. An adversary capable of monitoring transmissions at the endpoints of a connection can analyze their meta data characteristics and try to identify their relations. Such relations match the user identity with the destination of the connection and de-anonymize a user (cf. Figure 2.2). We differentiate *passive* attacks limited to monitoring traffic in compromised nodes, and *active* attacks that allow for an interference with existing transmissions or the injection of new packets.

Problem Statement. End-to-end confirmation uses statistical comparison techniques, e. g., similarity or distance metrics, to match related traffic. The transmission characteristics on the path from the sender to the receiver can influence the success of such comparison metrics, e. g., packet loss or delays through congestion disrupt the traffic patterns during the transmission and increase the discrepancy between the sent and received patterns. We summarize the choice of attack metrics and network characteristics as *technical* influencing factor. Furthermore, the adversarial coverage of the network affects the success of an attack, as the adversary has no prior knowledge of the nodes involved in a connection. While a global coverage allows monitoring the entire network, non-global adversaries increase the success probabilities with increasing network coverage. We define the access to nodes as *operational* influencing factor.

Attack Characteristics. In the following, we define three important attack characteristics that are relevant in the context of this work.

- **Metric.** We use a statistical metric to compare, e. g., one entry trace with a set of exit traces. A high similarity or low error indicates the relation between both endpoints and allows for a de-anonymization attempt.

- **Features.** meta data features provide information about the transmission patterns of monitored traffic. Time series of meta data features are the input for the statistical comparison metrics.

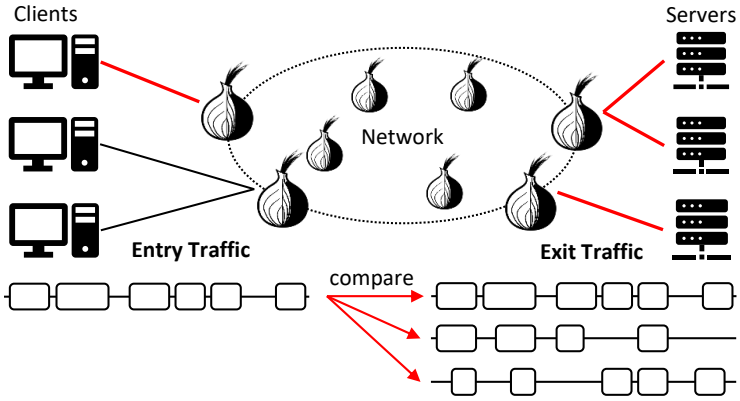


Figure 2.2. General concept of end-to-end confirmation. The adversary monitors entry and exit traffic (red) and identifies matches by comparing traces. The similarity between traffic from endpoints enables the deanonymization attempt.

- **Coverage.** The amount of covered nodes and links, i. e., those that can be the target of monitoring, influences the success probabilities of an attack. Global adversaries can monitor the entire network; weaker adversaries have access to a fraction of nodes and might miss pairs of endpoints that are part of the same circuit.

In this work, we analyze passive traffic analysis attacks and compare their performance in different use case scenarios (cf. Chapter 3), where we focus on different *metrics* and *features* and assume a global *coverage*. In the context of alternative countermeasure concepts (cf. Chapter 4), we assume the general threat of traffic analysis attacks conducted by adversaries with a nation-state coverage.

Website Fingerprinting. Website fingerprinting attacks allow an adversary to derive the accessed *websites* from encrypted traffic. In contrast to end-to-end confirmation attacks, website fingerprinting depends only on the traffic of an entry hop (cf. Figure 2.3) and compares the monitored traces with a pre-recorded data corpus.

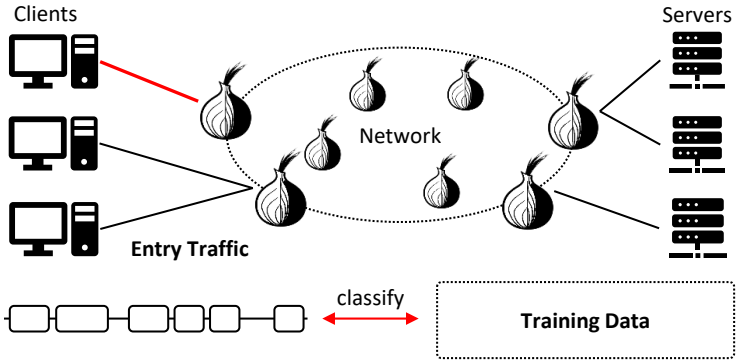


Figure 2.3. General concept of website fingerprinting. The adversary monitors traffic at entry connections (red) and compares traces with a trained and labeled data set. The training data requires a recording and labeling phase in advance.

Problem Statement. The identification of websites from encrypted traffic is a classification problem where the adversary pre-records the traffic of candidate websites for a training set. In the actual attack, an unlabeled sample trace is compared against the training set [RPJ+18]. By using a suitable classification technique, the adversary compares the recorded trace with all candidates of the training set and attempts to classify it.

Attack Characteristics. We identify three attack characteristics of website fingerprinting attacks.

- **Classifier.** Possible classification techniques range from simple distance metrics, over classical machine learning, to deep learning approaches.

- **Features.** Features are the traffic meta data and deliver the classifier input. It is possible to manually define the relevance of features or apply automated extraction mechanisms; additional preprocessing steps help to weight features according to their significance.

- **Setup.** The scientific evaluation of website fingerprinting attacks requires defining a setup. Prior work either uses closed-world setups with

n websites in the training set and the test set limited to the same n sites, or an open world setup with n training sites and $m > n$ possible test sites.

In this work, we transfer well-known website fingerprinting attacks from the context of Tor to attacks on mobile networks. To this end, we adopt successfully evaluated *classifiers* and adjust the *feature* space to LTE layer two traffic meta data. For the sake of simplicity, we apply a closed-world *setup* and test the feasibility of website fingerprinting in this new context.

Routing Attacks

Autonomous Systems (ASes) divide the Internet into large organizational units that forward messages to the desired destination. The routing between ASes is managed by the Border Gateway Protocol (BGP), which defines how ASes connect, i. e., the BGP manages the service agreements between providers. Consequently, the BGP routing tables define the paths a message will take when it is transmitted from A to B. Adversaries use routing attacks [NSZ+16, BMG+07] to manipulate such paths, forcing traffic through areas that are under adversarial control (cf. Figure 2.4).

Throughout this work, we do not directly address routing attacks, but consider it a way to improve the adversary's position in advance. Such improvements emphasize the threat of large-scale adversaries and support the motivation of this thesis.

2.2.3. Countermeasures

Defenses against traffic analysis attacks seek to limit the success of statistical comparison (end-to-end confirmation) or classification techniques (website fingerprinting). To this end, one obvious countermeasure is the obfuscation of transmission characteristics for disrupting the similarities between related streams. In the context of routing attacks, monitoring anomalies in the BGP routing tables or the selection of relays with a specific resilience against routing attacks are possible countermeasures. Following our primary focus on traffic analysis attacks, we define obfuscation techniques as follows.

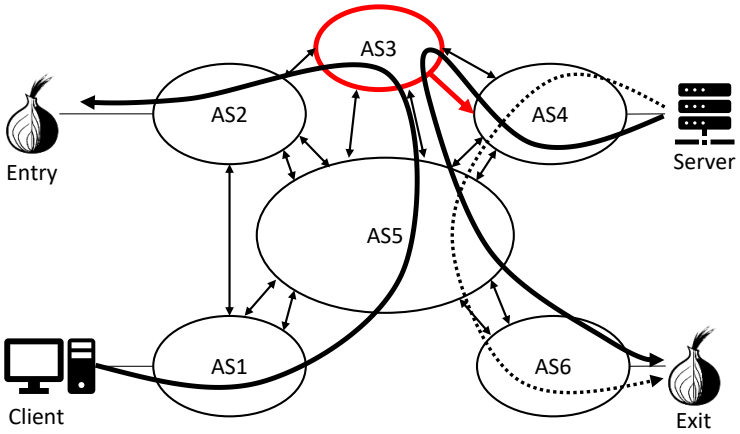


Figure 2.4. Example of routing attacks by means of BGP interception. Adversary intercepts the exit relay's IP address prefix [SEV+16]. Instead of the original route through `AS4,AS5,AS6` (dashed line), traffic now also traverses the malicious `AS3` (solid line).

Traffic Obfuscation. Traffic obfuscation conceals relations between pairs of monitored traces. To this end, it targets the meta data *features* that serve as an input for the comparison *metrics* of an attack, e. g., obfuscation schemes interfere with the timing between packets. Such interference influences the transmission characteristics and can induce latencies or an additional data overhead. As discussed in the second generation onion router [SDM04], the overhead of obfuscation must stay within acceptable "practical and economical" limits.

Problem Statement. Obfuscation techniques directly oppose end-to-end confirmation and website fingerprinting and seek to hinder the success of (statistical) comparison metrics. To preserve the original security and performance features of a system, the means of obfuscation must limit their overhead to an acceptable amount of delay. Furthermore, additional traffic may not exceed the available resources. Extensions to an existing system must not jeopardize its security, e. g., through an information leak.

Throughout this work, we create the starting point for an evaluation of obfuscation-based countermeasures in the context of Tor (cf. Chapter 3). As the success of such obfuscation is strictly limited through Tor's real-world requirements, we further analyze alternative defenses that step back from direct manipulation of traffic meta data (cf. Chapter 4). In the context of website fingerprinting attacks, we focus on demonstrating the feasibility of existing attacks in the new context of mobile networks and only provide a brief discussion of possible countermeasures (cf. Chapter 5).

2.2.4. Anonymity and Privacy

The above-mentioned classes of attacks and countermeasures are directly related to the security features of the Tor anonymity system. In addition to anonymity, Tor also assures confidentiality through the encryption of transmissions with perfect forward secrecy [SDM04]. In the following, we provide general definitions of privacy (Clarke [Cla99]) and anonymity (Pfitzmann et al. [PK01]).

Privacy

Clarke introduces the general term of privacy as an interest in personal space that limits the availability of sensitive information only to people invited to this particular space:

Definition 2.2.1 *Privacy is the interest that individuals have in sustaining "personal space", free from interference by other people and organisations.*

Following this definition, Clarke specifies the above general term of privacy to four dimensions (privacy of the person, personal behavior, personal communications, and personal data) of which two are particularly important for this work.

Definition 2.2.2 *Privacy of personal communications. Individuals claim an interest in being able to communicate among themselves [...] without routine monitoring of their communications by other persons or organisations.*

Definition 2.2.3 *Privacy of personal data. Individuals claim that data about themselves should not be automatically available to other individuals and organisations [...].*

Tor applies cryptography for layered onion encryption in each hop of the circuit, which protects the content of transmissions and creates confidentiality for users. In the remainder of this work, we assume that Tor's cryptographic features are secure and focus on attacks that threaten the *anonymity* of the system.

Anonymity

Pfitzmann et al. introduce anonymity as a state in which subjects successfully hide within a set of candidate subjects, the anonymity set:

Definition 2.2.4 *Anonymity is the state of being not identifiable within a set of subjects, the anonymity set.*

Definition 2.2.5 *The Anonymity Set is the set of all possible subjects.*

To apply the above definitions to the context of anonymity systems, and Tor in particular, we must match the "set of all possible subjects" with the problem statements of each attack. To this end, we focus on traffic confirmation (cf. Sections 3 and 4) and website fingerprinting (cf. Section 5) attacks.

End-to-End Confirmation

In an end-to-end confirmation attack, the adversary matches traffic from two endpoints of a connection. As Tor only applies encryption to the first three hops of the circuit, i. e., between the client and entry, entry and middle, and middle and exit, encrypting the last hop between the exit relay and the server is up to the server and cannot be enforced by Tor. This leads to a situation in which the server is known, whereas users hide within the set of all connections that were active during the time of adversarial monitoring. These active users are the set of all possible subjects and build the anonymity set. The completeness of the adversary's candidate set depends on the coverage of

the attacked network, e. g., a non-global adversary can only analyze a sub-set of all entry connections.

Website Fingerprinting

In a website fingerprinting attack, the adversary tries to classify an unknown trace of traffic within a pre-recorded database of labeled traces. In contrast to an end-to-end confirmation, this requires monitoring entry connections only and leads to a different definition of the anonymity set. As the entry connection already relates to the identity of a user, an attack is successful if the traffic of this origin can be matched to one of the website candidates in the pre-recorded database. We define the set of all possible websites as the anonymity set and distinguish closed-world setups, in which the candidate set is limited to the n websites in the training set, and open-world setups in which the training set is only a subset of $m > n$ possible websites.

2.3. Related Work

Prior work in the context of anonymity systems suggested a series of attacks and countermeasures that either threaten the security goals of a system or suggest new defenses. Besides work that directly relates to traffic analysis, we further find alternative approaches that extend the original problem and solution space of traffic analysis attacks. In the following, we discuss existing attacks and countermeasures and provide an overview of methods that add new perspectives to this context.

2.3.1. Traffic Analysis Attacks

Traffic analysis attacks aim for learning sensitive information from the meta data characteristics of encrypted transmissions. In the following, we overview passive and active attacks, document the development of website fingerprinting, and explain how routing attacks support traffic analysis (cf. Table 2.1).

Passive Attacks

Recent work suggests two types of metrics for *flow comparison* attacks, in which the adversary identifies similarities between input and output traffic. Correlation-based [LRWW04, SW06] attacks compute the similarity of monitored traces and identify relations between streams using the inter-arrival times of packets. Mutual information [ZFG+04, KAL+15] is a measure of the relationship between two streams and estimates similarity based on the entropy of observed pairs. Again, inter-arrival times are mentioned as traffic feature for this type of attack. We find further classes of passive attacks, e. g., Internet Exchange Point (IXP) sampling [MZ07] or disclosure attacks [DDT07] use statistical methods to compute the probability of transmissions being related. Most passive attacks assume a global adversary (cf. Table 2.1) but choose different types of evaluation setups that range from theoretical (○) to real-world (●) experiments. Also, the consideration of different application types (● considered, ○ not considered) is not consistent, leading to fundamental differences in the analyzed traffic features.

Passive attacks are relevant for the evaluation framework we introduce in Chapter 3. In particular, we benefit from the fact that passive attacks are limited to monitoring traffic. Therefore, we can pre-record an extensive database of different scenarios and apply passive attack techniques afterwards.

Active Attacks

Active attacks allow the adversary not only to monitor transmissions but also to interfere with traffic to improve the chances for a successful attack. One example of this are watermarking attacks [HB11, HB13, BPW13, WCJ07], in which the adversary injects 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 adversary. Other active attacks use coding techniques to add a specific pattern [YFG+07, LLY+09, SRW+10, LFJ+13] or exploit dependencies within and between transmission protocols [FLL+09, HBS13]. Similar to passive attacks, we find different approaches to evaluating active attacks with either theoretical or real-world experiments and differences in consideration of noise.

Table 2.1. Classification of Traffic Analysis Attacks.

Attack	Ref.	A/P	Adv.	Setup	Noise	App.
Flow Comp.	[LRWW04, SW06] [ZFG+04, KAL+15]	○	●	◐	◐	○
IXP Samples	[MZ07]	○	◐	●	●	●
Disclosure	[Dan03, KAP02] [MD04, DDT07]	○	●	○	○	○
Watermarking	[BPW13, WCJ07] [HB11, HB13]	●	◐	●	●	●
Coding	[YFG+07, LLY+09] [SRW+10, LFJ+13]	●	◐	●	●	○
Protocol	[FLL+09, HBS13]	●	◐	●	●	○
n-1	[SDS02, O'C05, DP04]	●	●	○	○	○

A/P active ●, passive attack ○
Adv. partial adversary ◐, global adversary ●
Setup evaluated in live Tor ●, reduced private network model ◐, theoretical model ○
Noise real noise ●, empirical noise ◐, statistical noise ○
App. considered ●, not considered ○

Active attacks are out of scope for the evaluation framework of Chapter 3, but motivate the implementation of alternative countermeasures in Chapter 4 that circumvents the threat of passive *and* active attacks.

Website Fingerprinting

In 1998, Cheng et al. [CA98] introduced a first traffic fingerprinting attack that uses a two-dimensional feature space to identify web pages by their total HTML size and the size of a page load. After this, iterations of follow-up work continuously improved the realism and success rates of existing attacks [CZJJ12, LL06, AG16, HWF09, PNZE11, WG13]. Up to this point, attacks successfully identify web pages from encrypted traffic, but are likely to fail in the presence of sophisticated security mechanisms.

The most prominent example of such additional protection is Tor, which has become an important target for the next generation of website fingerprinting. Wang et al. [WCN+14] introduce a k-Nearest-Neighbors classifier that uses a total of 3000 traffic features derived from multiple weighted

feature families; they achieve success rates in the range of 90 % to 95 %. Other attacks introduce k-Fingerprinting [HD16] or use Support Vector Machines (SVM) [PNZE11, CZJJ12] and the cumulative sum of packets (CUMUL) [PLZ+18] achieving 90 % to 93 % for a closed set of 100 candidate pages. All of these attacks are successful on obfuscated and encrypted Tor traffic but depend on a manual selection of features for the classification process. Rimmer et al. [RPJ+18] overcome this by introducing an automated website fingerprinting approach. Instead of consulting a comprehensive knowledge of the underlying protocols, they use automatic feature engineering [Wan15] and Deep Neural Networks [AG16] to achieve a classification success of up to 94 % for a 900-page data set.

Website fingerprinting attacks are relevant for Chapter 5, where we transfer state of the art attacks on Tor to the context of mobile networks. Besides fundamental differences in the attacker model, the specific protocols of mobile networks influence the set of available meta data features.

Routing Attacks

Routing attacks improve the starting position for a traffic analysis attack by forcing transmissions through areas under adversarial control. AS-level adversaries can manipulate the Tor routing [BMG+07] or perform BGP hijacks [SEV+16, NSZ+16] to blackhole traffic. While this allows for the observation of transmitted data, the hijack results in dropped connections that may reveal the adversarial activity. To overcome this, more sophisticated BGP interception attacks [BFZ07] force the adversarial AS to be at an intermediate point of the path. In this case, the connection is kept alive and allows the adversary to monitor the transmissions.

Routing attacks make traffic-analysis attacks more effective by increasing the probability of successfully de-anonymizing users in the network, or they help to learn the positions of critical nodes, e. g., onion services, on the network. In the context of this thesis, routing attacks emphasize the threat of large-scale adversaries, nevertheless, they are out of scope for our work.

Conclusion

Passive and active traffic analysis attacks are a severe threat as soon as the adversary monitors a significant portion of network connections. Routing attacks and current network characteristics like strong and sometimes colluding ASes, centralized infrastructures, or nation-state adversaries support the success of offensive work further. While we find this strong starting position for attacks, a high diversity of evaluation techniques makes it impossible to assess the actual threat of existing attacks. Website fingerprinting attacks depend on a less restrictive attacker model, as they only monitor the entry node of a connection. Nevertheless, the realism of scientific evaluation setups is still limited and cannot depict the incredible variety of websites on the Internet [JAA+14].

Deep learning advances both areas of research, traffic confirmation and website fingerprinting, to a new situation. While countermeasures against statistical attacks must be considered expensive but technically successful, countering automated feature engineering is much more challenging. We discuss examples for this as follows.

2.3.2. Traffic Obfuscation

High-latency systems [GT96, DL04, DDM03, Cot19, DG09] are less restricted by performance requirements and have more degrees of freedom for expensive traffic obfuscation. Nevertheless, they encounter the same anonymity trilemma [DMMK18] as low-latency systems [PHE+17] and must pick two out of the three important features *strong anonymity*, *low bandwidth overhead*, or *low latency overhead*. In the following, we discuss countermeasures that extend the original system design of Tor for protection against end-to-end confirmation or website fingerprinting attacks.

End-to-End Confirmation

The timing relations of transmissions are among the most important meta data features of prior attacks. Consequently, interfering with the number of packets sent or the timing between packets can help to disrupt the similarities

of related traces. Mixing approaches [Cha81] like stop-and-go mixes [KEB98] or alpha mixing [DSS06] adopt this concept and randomly hold back packets of a stream. While such random delays disrupt the timing relations during the transmission process, they also induce additional delays that contradict low-latency features. Another way of perturbation is the injection of dummy traffic, so-called chaffing [BL02, SW06], where extra packets are injected. Again, chaffing disrupts timing relations to challenge comparison metrics.

In the context of this work, we analyze at which expense mixing hinders the success of passive traffic analysis attacks (cf. Chapter 3) and test alternative countermeasures that circumvent adversaries rather than directly obfuscating traffic (cf. Chapter 4).

Website Fingerprinting

Early defenses against website fingerprinting are traffic morphing [WCM09], where the distribution of packet lengths from one website is transformed to adopt the characteristics of another website, or HTTP obfuscation [LZC+11] to manipulate TCP protocol features. Even though these specific countermeasures defend against older attacks [BLJL05, LL06, SSW+02], their specific obfuscation can be circumvented by recent attacks [CZJJ12, WCN+14, WG13]. Generic defenses refer to a more comprehensive concept of creating indistinguishable packet sequences for *all* websites. They use fixed rate transmissions with additional cover traffic [CNJ14, CNW+14] to adjust the traces of different websites, but induce an unbearable overhead while being inflexible with the injected packet rates. Goldberg et al. overcome these restrictions with a more efficient half-duplex countermeasures [WG17], nevertheless, some of the essential assumptions in their approach are highly susceptible to noise and make it harder to derive a security bound [Che17].

In the context of deep learning attacks, classical countermeasures do not defend against the automated feature generation process. Alternatively, adversarial examples [CW17] might deliver inputs with the goal to mislead the neural network. Unfortunately, we do not find any adversarial examples to counter website fingerprinting against Tor at the moment. As an alterna-

tive, Rimmer et al. [RPJ+18] discuss the half-duplex approach of WalkieTalkie [WG17] that might have a similar effect on deep learning attacks.

Conclusion

Existing obfuscation techniques against traffic confirmation and website fingerprinting attacks often focus on the transmission characteristics for providing a *direct* counter against statistical or machine learning techniques. Unfortunately, such interference with packets or the transmission channel always has a negative influence on the performance features of a system, which challenges their real-world deployment. In addition to the performance requirements, countermeasures must also consider the restrictions of the underlying protocols, e. g., attacks exploiting the transmission direction as one feature to distinguish connections cannot be countered by reversing the data flow between the client and the server. Especially deep learning attacks benefit from such restrictions, as the automatic feature engineering is capable of identifying a series of non-modifiable transmission characteristics.

2.3.3. Further Defensive Directions

While Traffic obfuscation is a direct counter to end-to-end confirmation and website fingerprinting attacks, other defensive approaches represent an alternative by circumventing or limiting the threat of an attack. In the following, we discuss path selection algorithms that help to improve the performance or security of Tor circuits. Furthermore, we provide an overview of censorship circumvention approaches that seek to avoid censorship authorities.

Path Selection

Tor selects the relays for a circuit according to the bandwidth they can offer, and nodes with better performance are preferred over smaller and often less stable ones. This does not only influence the performance we experience when using Tor, but it also has an impact on the anonymity set [BKMM14, JJD+15, EEFR16] in which we hide. Recent work suggested different strategies to improve Tor's circuit establishment. Better congestion

management [WBFG12, JGW+14] can help to improve the load balancing in Tor and increase the number of relay candidates for a circuit through a better distribution of traffic. Another critical factor is autonomous systems (AS) [AYM12, ES09, JWJ+13, BW16a], as an AS-level adversary is in a dominant position to perform traffic-analysis attacks.

Path selection algorithms are relevant in the context of geographical avoidance that circumvents the threat of nation-state adversaries (cf. Chapter 4). Similar to the mechanisms in an avoidance scheme, path selection algorithms use information about the network infrastructure or circuits for improving its security or performance.

Censorship Circumvention

We focus on two directions in the context of censorship circumvention. Decoy routing is a technique that combines obfuscation and the support of a proxy to access content that is otherwise censored. Pluggable transports are an obfuscation extension to Tor that help make standard Tor traffic look like something else, e. g., any other traffic that is not the target of monitoring and blocking.

Decoy Routing: Decoy Routing [KEJ+11, EJM+15, WSH14] circumvents censorship and blocking by routing critical traffic through servers outside the censored area. For this context, we assume to be located in a country where specific sites are prohibited, and all requests made to such contents are blocked or even reported. To overcome this situation, so-called decoy routers are used that provide accepted content outside the censored area, hence, sites that are not hosted in your country but tolerated by the censor. They act as said man-in-the-middle and forward requests to blocked sites as well as send the contents back to the client, all obfuscated through techniques that hide the actual payload of a transmission. Countermeasures like RAD (routing around decoys) [SGTH12] try to avoid the functionality of decoy routers by forcing routes on alternate paths that cannot traverse the decoy router. In general, decoy routing is another possibility to circumvent censorship, but it does not consider routing attacks and consequences through traffic analysis.

Pluggable Transports: Pluggable transports are another way of censorship circumvention as they provide access to Tor even in case conventional circuits and bridges are not an option because of blocking [BHKL13, WWY+12, DCRS13, Fif19, AW19]. Despite the full range of pluggable transport types, the general principle uses obfuscation to make Tor traffic look like some other, benign, protocol that is not the target of blocking or monitoring through the censor. In the context of geographical avoidance, the use of pluggable transports is complicated. They cannot guarantee secure routes, even though the obfuscation techniques make it much harder for an adversary to learn sensitive information from traffic meta data. That said, *it depends*. Random patterns in the obfuscation help to disrupt relations between traffic streams and detecting a connection to Tor becomes more difficult, but there is no protection against routing attacks.

Censorship circumvention helps to evade the influence of large authorities and delivers an essential input for the design of a geographical avoidance system (cf. Chapter 4).

Quantization of Anonymity

Different formal approaches seek to analyze the security capabilities of anonymity systems. Such lower or upper bounds help to understand the theoretical anonymity guarantees of a system in the presence of individual attacker models. AnoA [BKM+13] is a differential privacy based framework for analyzing common anonymity properties, e. g., sender anonymity, sender unlinkability, or relationship anonymity. The framework involves a modular specification of different adversarial capabilities for testing the above anonymity properties for single or multiple messages. The authors apply the theoretical quantization methods of AnoA to an adjusted version of Tor and analyze the impact of passive attacks. Like other formalization approaches [CL05, FJS07, FJS12, GH13, STRL01], AnoA does not consider the timing characteristics of network transmissions and leave timing attacks uncovered. In response, Backes et al. introduce the time-sensitive analysis framework TUC [BMM14] that extends prior systems with timing features.

2.3.4. Attacks on Mobile Networks

Chapter 5 switches the context of traffic analysis attacks from Tor to mobile networks and focuses on the feasibility of website fingerprinting on LTE layer two traffic. In contrast to conventional website fingerprinting attacks, the *radio* adversary does not monitor nodes in the network infrastructure but captures wireless transmissions within the cell of a commercial network. In the following, we focus on attacks that invade the privacy of LTE users.

Privacy Attacks on LTE

We find different classes of attacks on mobile networks, for example, International Mobile Subscriber Identity (IMSI) catchers are a common way to pinpoint users, but many of them require active radio capabilities. In the following, we focus on attacks that reveal sensitive user information using the abilities of a *passive* radio attacker.

Besides an early approach on website fingerprinting on LTE transmissions [RKHP19], we only find related work in the context of identification and localization attacks. 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 [KKHK12, SBA+16]. As discussed in Section 5.6.2, countermeasures like a frequent Temporary Mobile Subscriber Identity (TMSI) reallocation can only limit the period in which the adversary can de-anonymize the user [HBK18], but do not prevent the attack. In contrast to layer two website fingerprinting, paging takes place on a layer three control channel, i. e., we do not depend on the third layer TMSI, but use the Radio Network Temporary Identifier (RNTI) as *radio* session identifier. Similar reallocation attempts exist for the RNTI [Jov16b, Jov16a], nevertheless, such approaches lack randomness and do not prevent from matching the radio layer identity with the upper layer TMSI [RKHP19]. Assuming RNTI reallocation techniques with sufficient randomness, such countermeasures still cannot prevent against layer two fingerprinting.

2.4. Attacker Model

In the following, we define a general attacker model that i) covers traffic analysis attacks and ii) preliminary attacks for improving an attack's starting position. To this end, we first overview primary and secondary attack aims and further specify the adversary's technical capabilities.

2.4.1. Attack Aims

One central motivation for attacks against anonymity systems is learning sensitive information about the system's users and their actions. In the context of this work, we are interested in end-to-end confirmation [SW06, ZFG+04, DDT07, HB13, WCJ07] that allows for matching the endpoints (client and server) of a connection, and routing attacks [BMG+07, SEV+16, NSZ+16, BFZ07] that extend the monitoring coverage over the attacked network.

Primary: De-Anonymization

As introduced in Section 2.2.4, Tor offers anonymity by separating the endpoints of a connection, e. g., the user identity, and the accessed website. Attacks against Tor's anonymity aim for learning sensitive information that otherwise is expected to be protected. Such confidential information can include linked communications of a user to create a behavioral profile [SDM04] (long-term), or related transmissions at the endpoints of a connection allowing for the de-anonymization of a user (short-term). We consider attacks successful as soon as the security goals of Tor are violated, i. e., the adversary gains access to otherwise protected information.

Secondary: Preliminary Attacks

Preliminary attacks help to improve the starting position of an adversary. This includes gathering additional information about critical nodes in the network infrastructure like onion services [JJG+17], or manipulating the routing behavior [SEV+16, NSZ+16]. Preliminary attacks increase the probability of monitoring related connections and are relevant for non-global adversaries.

2.4.2. Adversarial Properties

We define two main influencing factors for the success of traffic analysis attacks. First, the technical capabilities determine the adversary's ability to analyze monitored traffic. Second, the operational abilities summarize aspects like the network coverage (number of nodes that can be monitored) and the protocol stack level of the attack. In the following, we define both influencing factors in the context of traffic analysis attacks and match them to the attacker properties initially introduced by Raymond [Ray01] and systematized by Edman and Yener [EY09].

Technical Specification

To de-anonymize a connection, the adversary compares traffic meta data from traces monitored at the entry and the exit relay of a Tor circuit. The available meta data includes transmission relations like the timing between pairs of packets or the number of packets over time. Further, it depends on the protocol stack layer of the monitored node, e. g., can reach up to the Internet layer or the transport layer. We distinguish the *capability* of an adversary to either conduct *passive* attacks, in which the capabilities are restricted to monitoring transmissions through compromised nodes, and *active* attacks that further allow manipulating traffic.

Operational Specification

The success of traffic analysis attacks increases with a higher *visibility*, i. e., when the adversary can monitor traffic from a large number of users. In the context of Tor, this resembles a situation with adversarial access to relays in the Tor network (by contributing as a volunteer relay operator), to layer three or four switches (network nodes that forward IP or TCP/UDP traffic), or monitored IXP. We consider large-scale adversaries that range from single AS-level [SFP16] coverage of approximately 40 % to colluding AS-level and nation-state adversaries that achieve up to 85 % coverage of the network [NSZ+16]. The adversarial *mobility* decides whether the partial visibility of the network is either *adaptive*, i. e., the adversary can change what

nodes are monitored, or *static*, where the set of monitored nodes stays the same. Furthermore, we distinguish an *internal* or *external* participation in the attacked network. More precisely, the adversary monitors traffic by participating in the network infrastructure or has access to the communication medium.

While routing attacks allow for actively improving the adversarial network coverage, specific characteristics of the Internet infrastructure influence the starting position of an attack further [KFR09]. Examples for this are the divide between western countries and the rest of the world that often forces traffic towards the US or the Netherlands, where a majority of content is hosted [EEFR18], or threats emerging from inter-domain routing behavior [SDB16]. Such Internet hegemony offers nation-state adversaries a powerful position to monitor even international traffic.

2.4.3. Specification of Attacks

In the remainder of this work, we focus on passive end-to-end confirmation for the comparison of state of the art attacks (cf. Chapter 3). In the context of alternative countermeasures, we consider the general threat of traffic analysis attacks conducted by nation-state adversaries (cf. Chapter 4). Finally, we analyze passive website fingerprinting and active traffic watermarking on mobile networks (cf. Chapter 5) that depend on a radio layer adversary. Based on the above general definition of attack aims and capabilities, we specify individual attacker models in the context of each chapter.

Es ist nichts so klein und wenig,
woran man sich nicht begeistern könnte.

— Friedrich Hölderlin

3

Attack and Countermeasure Performance

Contents

3.1. Introduction

The existence of successful de-anonymization attacks against Tor has a major impact because of its broad use. This potential impact motivates different classes of attacks attempting to reveal sensitive information about entities in the network [HB13, MKJ+11, BPW13]. Such academic approaches are an essential building block for improving Tor by more clearly defining potential threats. Nevertheless, we are uncertain about the real-world consequences of academic attack concepts, i. e., scientific attacks *can* pose a severe threat and affect millions of users, but are driven by a focus on novelty rather than realism. This leads to a fundamental challenge of estimating an attack's real-world impact.

Our focus in this work is on passive traffic analysis attacks. These attacks are a current concern to the Tor community, in which an adversary compromises user anonymity by correlating transmissions at the entry and exit of the circuit [MD05,SW06]. Recent work [CBP+14,KAL+15,BPW13,LFJ+13] has demonstrated that an autonomous system (AS)-level adversary can successfully conduct confirmation attacks, correlating the characteristics of transmitted data to identify connections within the network.

The weakness of correlation is aggravated by routing attacks and nation-state adversaries with capabilities to surveil substantial fractions of the network. Border Gateway Protocol (BGP) attacks like RAPTOR [SEV+16] can increase the efficacy of confirmation attacks by directing traffic through an adversarial AS. This allows adversaries to have a near-total view of the network, a threat model not addressed by Tor. Mitigating traffic confirmation attacks, in particular against a *global adversary*, remains an open research problem [The19e].

Tor threat research is lacking comparative evaluation methodologies. Instead, analyses have been very divergent, ranging from theoretical models to approximate simulation systems [JH12], to experiments on the live Tor network [WCJ07]. Theoretical models provide upper and lower bounds but are limited by the assumptions made. Simulated systems can incorporate more real-world characteristics and often analyze network characteristics at a realistic scale, but only approximate certain parameters like the dynamics

of an underlying network. The complexity of real-world network conditions makes it impossible to define holistic models that cover all potential cases, a fact that only allows for an estimation of effects on theoretical models and simulations. In contrast, experiments on the live Tor system demonstrate realistic conditions. However, especially in the context of traffic analysis attacks, work on the live network puts users at risk and is ethically discouraged [The19b]. Beyond the individual strengths of each of these methods, their *diversity* has led to a fundamental drawback: it is difficult to compare different attacks or understand their combined impact. This lack of comparability hinders the ability to understand current attack vectors and progress defensive research in response.

We introduce *DigesTor* to address this fundamental shortcoming. *DigesTor* is an evaluation framework that guarantees comparability for recent, current, and future passive traffic analysis attacks, combining the strengths of simulated and real-world evaluation. The framework runs a virtual private Tor network that generates traffic for typical scenarios on which arbitrary attacks can be evaluated. The system uses virtual machines with individual CPU cores for each node and transmissions of realistic traffic through the actual network stack. Intermediate links simulate realistic network conditions using traffic shaping with parameters from empirical measurements in the live Tor network. This experimental setup increases realism over artificial traffic generation in simulated environments [JH12], can provide realistic link models, and satisfies the ethical guidelines for Tor research.

DigesTor includes a suite of state of the art attack techniques that we evaluate using our framework. As a starting point for future work, this analysis provides a first performance comparison of existing attacks for their de-anonymization capabilities. Also, we demonstrate *DigesTor* by evaluating the use of delays as a potential countermeasure. In short, we make the following three main contributions with a focus on the first research question.

Research Question 1 *How can we overcome the scientific diversity in existing attacks to facilitate the development of new countermeasure techniques?*

- We release *DigesTor*, a comprehensive evaluation framework for passive traffic analysis attacks on Tor. This framework provides a basis to

enable a fair comparison of existing and future attacks, is made publicly available, and includes an extensive corpus of transmission traces.

- We demonstrate the usefulness of *DigesTor* to evaluate the performance of state of the art attack techniques. This leads to a first empirical overview of attack performance for different exemplary use cases and is a starting point for the development of future techniques.
- We use *DigesTor* to analyze low-latency mixing as a potential countermeasure to passive traffic analysis attacks. Results show that mixing, in fact, can counter confirmation attacks at a limited performance overhead only.

The above contributions result from a collaboration with Christina Pöpper.

3.2. *DigesTor* Framework

DigesTor is an open source analysis framework that provides comparability for the assessment of passive traffic analysis attacks. We offer a high-level overview and introduce its evaluation set in the following.

3.2.1. System Components

DigesTor provides two core features: a Traffic Analysis Framework and a Virtual Private Tor Network (cf. Figure 3.1). The *Traffic Analysis Framework* applies a set of attack techniques from related work to traces of our experimental network and outputs a performance assessment regarding the success of existing end-to-end confirmation attacks. The framework covers five comparison metrics, which estimate the similarity or distance between observations in the network, i. e., pairs of client and server traces.

The *Virtual Private Tor Network* is used to generate network traffic that corresponds to typical use case scenarios. The traces are the monitored traffic streams an adversary would gather in a confirmation attack and serve as an input to generic passive end-to-end confirmation attacks. We use a virtualized private network for two main reasons. First, isolating the setup protects users of the live Tor network and ensures we do not violate the

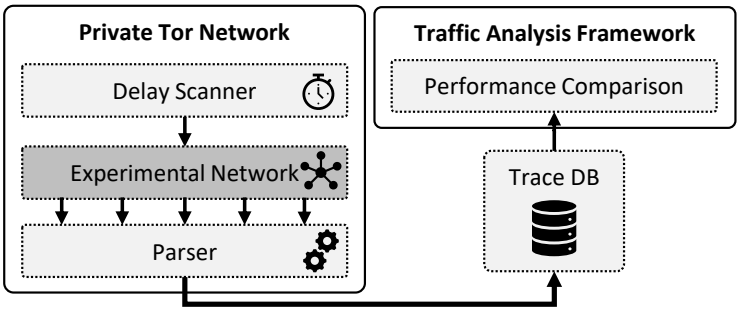


Figure 3.1. High-level overview of *DigesTor*. The private Tor network generates traffic for a centralized trace database; the traffic analysis framework applies passive attack techniques for a performance comparison. Results build the attack benchmarks.

existing ethical guidelines for Tor research [The19b]. Second, the technical characteristics of a virtual setup provide significant advantages compared to a simulation model. Using virtual machines for all nodes in the network, we utilize the actual protocol stack and transmit realistic application data. To improve the realism of our private setup, we use empirical link models to imitate transmission delays monitored in the live Tor network.

3.2.2. Traffic Analysis Framework

In the following, we detail the traffic analysis component of *DigesTor*. Recent work suggests two types of metrics for flow comparison attacks. Correlation-based [LRWW04, SW06] attacks compute the similarity in monitored traffic and identify relations between streams using the inter-arrival times, i. e., periods between packets. Mutual information [ZFG+04, KAL+15] is a measure of the dependence of two streams and estimates similarity based on the entropy of observed pairs. Again, inter-arrival times are mentioned as a traffic feature for this type of attack.

Attack Techniques.

From the current state of passive end-to-end confirmation attacks, we adopt the Pearson correlation coefficient (`P`) and Mutual Information (`MI`). We extend this by the Root-Mean-Square-Error (`RMSE`) as a measure of distance between two observations, and a scalar comparison (`SC`) of features, in which we compare the sum of a meta data vector. Moreover, we sample an optional preprocessing step with the combination of the principal component analysis and Pearson correlation (`PCA-P`).

Pearson Correlation Coefficient (P). Pearson's correlation coefficient ρ measures the extent of linear similarity between two vectors X and Y [SW06, LRWW04]. It is defined as:

$$\rho(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2 \cdot \sum_{i=1}^{n}(y_i - \mu_y)^2}}, \tag{3.1}$$

where n is the sample size and μ_x, μ_y are the means of the samples in X and Y, respectively.

In particular, Pearson's correlation coefficient can be used to identify value sets that behave similarly over time. Values of Equation 3.1 range between $+1$ and -1, where $+1$ indicates total positive correlation and -1 indicates complete negative correlation. A value of 0 represents the case where the input sets are uncorrelated. We expect the correlation coefficient to be higher for data sets from related client-server connections than for two separate connections.

Mutual Information (MI). The MI [ZFG+04] of two data sets is a measure of mutual dependence. In particular, MI uses the entropy of two random variables to predict one variable by observing the other. The MI between two data sets X and Y is defined as:

$$MI(X,Y) = \sum_{x,y} P_{XY}(x,y) \cdot log \frac{P_{XY}(x,y)}{P_X(x) \cdot P_Y(y)}, \tag{3.2}$$

where $P_{XY}(x,y)$ denotes the joint probability distribution. We compute the marginals $P_X(x)$ and $P_Y(y)$ from the joint distribution:

$$P_X(x) = \sum_y P_{XY}(x, y). \tag{3.3}$$

While a high MI indicates a tight dependency between the sets, an MI of 0 suggests that the values are independent. We use the MI to find related entry and exit traffic patterns for each of our measured features. Since the computation of the MI is based on probability distributions, we estimate the distributions by clustering the measurements in equal bins to determine the relative frequency distribution.

Root-Mean-Square-Error (RMSE) The RMSE measures the accumulated squared differences between the individual values of two data sets. It is an accuracy measure with scale-dependence [HK06] and is meant to identify errors between data sets of the same model. The RMSE between the entry trace X and the exit trace Y is defined as:

$$RMSE(X, Y) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}}, \tag{3.4}$$

where n is the number of observations corresponding to synchronized data samples.

Scalar Comparison. The attack technique of scalar comparisons (SC) is defined as a differentiation based on the difference of sums of individual features. In particular, we determine the minimum difference between cumulative client features in trace X and cumulative server features in trace Y to indicate matching connections:

$$SC(X, Y) = \left| \sum_{i=1}^{n} x_i - \sum_{i=1}^{m} y_i \right|, \tag{3.5}$$

where n is the length of the feature vector at the client and m is the length of the feature vector at the server.

Principle Component Analysis (PCA). A PCA takes a multi-dimensional observation matrix and computes its linear uncorrelated principal components to extract feature sets with maximal variance. This simplifies the features by constructing an orthogonal representation with minimal redun-

dancy. We briefly point out the necessary steps to show the effect on traffic matching. For an in-depth analysis of PCA, we refer to Jolliffe [Jol02].

First, we calculate a covariance matrix $\mathbf{\Sigma}$ of the observation matrix $\mathbf{C}$ containing all features:

$$\mathbf{\Sigma}(\mathbf{C}) = \begin{pmatrix} \mathrm{cov}(f_1, f_1) & \mathrm{cov}(f_1, f_2) & \cdots & \mathrm{cov}(f_1, f_5) \\ \mathrm{cov}(f_2, f_1) & \mathrm{cov}(f_2, f_2) & \cdots & \mathrm{cov}(f_2, f_5) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(f_5, f_1) & \mathrm{cov}(f_5, f_2) & \cdots & \mathrm{cov}(f_5, f_5) \end{pmatrix} \tag{3.6}$$

with

$$\mathrm{cov}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_x)(y_i - \mu_y), \tag{3.7}$$

where n is again the number of measurement points for each feature and μ_x, μ_y are the means of the respective feature. Next, we perform an eigendecomposition of matrix $\mathbf{\Sigma}$:

$$\mathbf{\Sigma}\mathbf{v} = \lambda \mathbf{v}, \tag{3.8}$$

with eigenvectors $\mathbf{v}$ and λ, from which we select the k greatest λ_i representing the features that exhibit maximal correlation. For the matching process, we use the reduced orthogonal feature sets of client and server candidates in a pairwise comparison through Pearson's correlation coefficient.

Attack Definition

In the following, we introduce the evaluation metrics that help us to measure the success of an attack, and specify the adversary model in context of *DigesTor*.

Evaluation Metrics. Eventually, we measure the success of an attack through the number of correctly guessed client/server connections, defined as *success rate*, and compare its improvement over random guessing, defined as ΔRG. The success rate describes the relative number of correct guesses in a setup, whereas the ΔRG indicates the strength of an attack. Furthermore, we use the area under the curve (AUC) for CDFs (cumulative distribution function) that summarize the results for combinations of multiple scenario

setups. The AUC is a measure of the robustness of a successful attack. A smaller AUC indicates *higher* success rates.

Attacker Model. We assume a global adversary with access to all entry guards and exit relays in the Tor network. This theoretical upper bound model is comparable to a nation-state adversary with approximately 85 % network coverage (cf. Section 2.4). From monitoring the traffic in relays, the adversary learns all unencrypted and meta data information up to the transport layer and uses the resulting traffic features to make de-anonymization attempts on the endpoints of connections. More precisely, the adversary compares traces recorded in the downlink direction from servers to clients and attempts pairwise matches between these endpoints. The technical capabilities of the adversary are limited to passive monitoring, i. e., we ignore the possibility of any active interference with the transmissions. Furthermore, we assume no computational or timely limitation is hindering the adversary to apply the above traffic analysis techniques.

The adversary follows the goal of maximizing the success rates for matching connection endpoints without being limited through any performance or stealth considerations. In contrast, the effects of countermeasures aim for restricting the success of the above comparison metrics and must stay within an acceptable range of performance impairments.

3.2.3. Helpers

Besides the core components of *DigesTor*, we utilize a parser for transforming raw traces of network traffic to aggregated meta data vectors. More precisely, we extract a set of five features f_i: (f_1 = `cnt`) packet counts, (f_2 = `iat`) inter-arrival-timing, (f_3 = `len`) packet length, (f_4 = `ttl`) time to live, and (f_5 = `wis`) TCP window size. This meta data can be read from the header information of a TCP/IP packet (`len`, `ttl`, `wis`) or derived from packet occurrences (`cnt`, `iat`).

Using a window-based aggregation [LRWW04, SW06], an average of all packets falling into one window is collected, e. g., for a measurement of 10 s and a window length of 0.1 s, we aggregate data in 100 equidistant windows.

This results in time vectors of meta data information $(f_{i,1}, f_{i,2}, \cdots, f_{i,n})$ with features f_i over n time windows.

This feature set is parsed for each connection and filtered in the downlink direction (data flow from server to client). The feature set is non-exhaustive but extends the standard features in the literature (packet counts, inter-arrival times) by three more characteristics (packet length, window size, time to live) whose relevance will be part of the experimental analysis in Section 3.4.

3.3. Experimental Setup

In our experiments, we perform a comparative performance evaluation of attack metrics and demonstrate *DigesTor* by analyzing mixing as a potential countermeasure against traffic analysis attacks. We introduce the experimental setup, define the analyzed use case scenarios, and discuss the influence of Tor's network infrastructure as follows.

3.3.1. Technical Specification

Our experimental network (cf. Figure 3.2) is defined by the different node types, i. e., clients, servers, and Tor relays, and by the topology that connects them.

Nodes. Entities in the network are configured to serve as i) clients that make requests through Tor, ii) servers that provide requested data, and iii) relay nodes that build the private Tor network. Each client connects to a predefined server and follows a use case scenario which includes either download requests via the *cURL* library or website browsing using the browser automation framework *Selenium* and a headless browser (developed as part of *Mozilla Firefox*). Requests are made through the SOCKS5 proxy at port 9050. They synchronize via NTP for all clients, i. e., experiments start and end at the same time. The server nodes provide file downloads over HTTP at port 80 and reverse proxy requests to a set of Alexa Top 50 websites at port 80 and 443. We use three relay nodes of which one is configured as the guard, one as middle and authority (we aim to limit the number of nodes

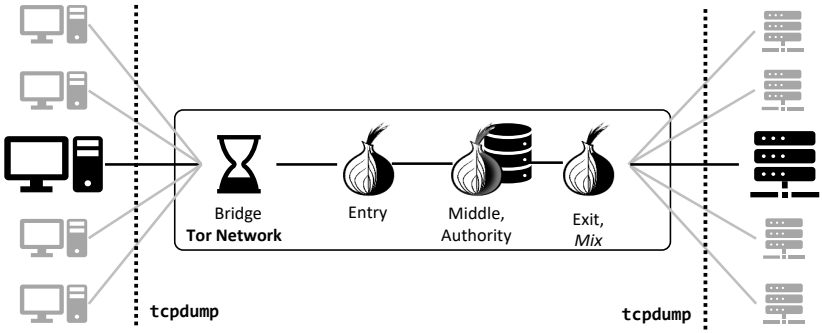


Figure 3.2. Private Tor network setup. Clients connect to servers through circuits of three relays. The bridge applies empirical traffic shaping for each client connection individually. Servers provide random binary files for downloads or proxy web requests.

in the network, therefore, the middle node functions as relay *and* authority), and one as exit relay. The relay, authority, and client nodes run Tor version `0.2.9.8` (December 2016).

Network. We use an empirical link model for the downlink connection of all clients. The link model adds per-packet delays drawn from measurements of arbitrary circuits in the live Tor network, which we individually specify for each connection. A bridge interface accomplishes traffic shaping, where each client connection samples from an individual delay distribution. For the network topology either a directed setup, using 1:1 connections between n clients and n servers, or a grouped configuration, using n:2 connections between n clients and two servers, is used. We fix the number of relays to three. The links between the relay nodes are not limited in capacity, whereas the server and client links are set to a maximum rate of 1 Mbit/s.

Hardware. The VMs run in a cloud space hosted in one central location, each node is assigned a distinct CPU core. The full setup can utilize up to 63 cores, 132 GB of RAM, and 504 GB of disk space. We capture the traffic of all client and server nodes using `tcpdump`. Raw network traces are gathered on one central file server for further processing and do not interfere with the performance of network nodes.

3.3.2. Scenarios

We test individual *topologies* of 2 to 30 clients to 2 to 30 servers in a *Directed* and of 2 to 30 clients to two servers in a *Grouped* setup. Furthermore, we distinguish three individual *application* models:

- **Static download.** The user requests a file from the server via *cURL* and permanently loads it during the entire duration of the measurement.
- **Random download.** Each user requests a file from the server via *cURL*, whereas on/off periods for the downloads are randomized for the entire duration of the measurement. Off periods are uniformly distributed between 2 s to 10 s, on periods are uniformly distributed between 2 s to 5 s.
- **Browsing.** From the Alexa Top 50 web pages, each client requests a random set of sites using a scripted headless browser. Between site requests, clients wait for a random period with a uniform distribution between 2 s to 5 s before the next request is sent.

We emphasize that the randomization of on/off periods can influence the results, as a higher variance in the duration of off periods helps to distinguish individual transmissions. Consequently, our results can only represent the parameter choices made above. We discuss the definition of more sophisticated use case scenarios in Section 3.5.

3.3.3. Comparison of Attack Metrics

In the following, we apply the Traffic Analysis Framework (combinations of features `cnt`, `iat`, `len`, `ttl`, `wis` and metrics `P`, `MI`, `RMSE`, `SC`, `PCA-P`) to all combinations (directed, grouped; static, random, browsing) and an increasing number of clients $n = 2$ to 30; each experiment is repeated for five random repetitions. We compute the general attack success (AS: how many connections were guessed correctly?), the improvement over random guessing (ΔRG: how much better was the attack compared to an uneducated guess?),

and the area under the curve (AUC: how convincing and robust was a result?) of the cumulative distribution function (CDF) of results.

3.3.4. Tor Network Infrastructure

While our experimental setup covers the *technical* comparison of attack metrics and traffic features, we are further interested in how Tor's network infrastructure influences the *operational* aspects of an attack. Therefore, we discuss the scalability of our setup and the relay selection process as a preliminary step to the performance comparison in Section 3.4.

Scalability

In the setups we demonstrate, clients run at a maximum rate of 1 Mbit/s. For the described *Grouped* and *Directed* scenarios, this translates into a throughput of 30 Mbit/s passing through each of the relays. This scale places these relays within the top 10 % of active Tor relays by bandwidth. Experiments with fewer active clients would approximate the traffic of less active relays, with approximately $\frac{2}{3}$ of relays transmitting at least 4 Mbit/s of traffic, the level of traffic we simulate in our smallest experiments. We do not model the number of active connections experienced by Tor relays. While we can expect a total of 500,000 active clients at any given point [JJ16], it is less clear how those clients are distributed across relays and bridges. However, the median relay will have less than 50 active clients regardless of the distribution. With up to 30 parallel connections our network setup achieves a similar relay workload.

Relay Popularity

Tor's network infrastructure is skewed towards the countries where we find the most Tor supporters, e. g., Germany (19 %), the US (19 %), and France (14 %) maintain more than half of the entire network. Furthermore, higher bandwidth relays are preferred in the circuit buildup procedure. An adversary can benefit from these characteristics and focus on frequently used

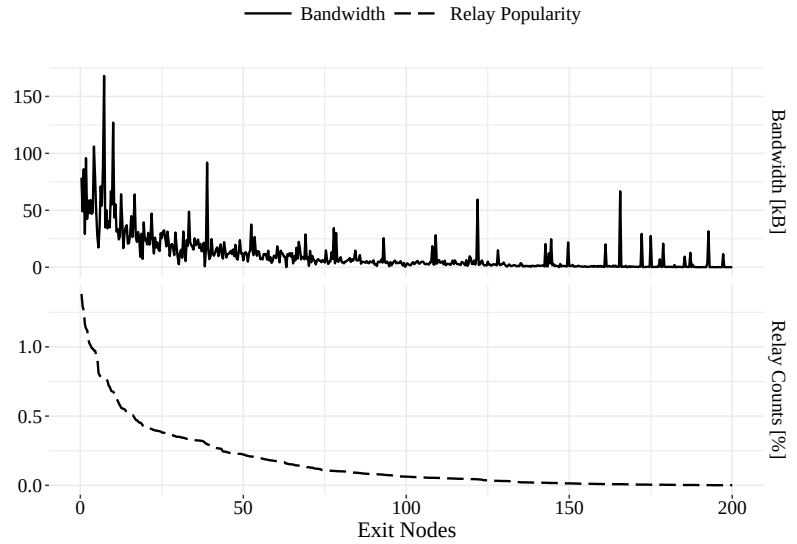


Figure 3.3. Statistics on Tor relay relevance. Distribution of exit relay popularity and advertised bandwidth, measured for a total of 100,000 Tor standard circuits. The bandwidth-focused selection of relays leads to a more frequent use of nodes with a high advertised bandwidth.

nodes, e. g., it is possible to cover 75 % of all *selected* exit relays by monitoring approximately 26 % of nodes (cf. Figure 3.3).

3.4. Evaluation

We use the above experimental setup of *DigesTor* for a first comparative analysis to (i) derive the best performing metric and feature combinations for each configuration, compare the characteristics of different (ii) topologies and application types, and (iii) analyze mixing as one possible countermeasure against traffic confirmation. Finally, we (iv) give an overview of the takeaway messages of our evaluation.

Table 3.1. Overview of Best Performing Metric and Feature Combinations.

Scenario	Metric	Feature	Δ RG	AUC	AS
Directed	P	ttl	35 %	0.72	0.49
Grouped	MI	iat	23 %	0.50	0.55
Random	RMSE	cnt	52 %	0.48	0.80
Static	MI	iat	16 %	0.65	0.46
Browsing	SC	iat	7 %	0.70	0.34
Global	MI	iat	23 %	0.61	0.52

Δ RG Relative improvement over random guessing
AUC Area under the curve for CDF of results
AS Relative attack success

3.4.1. Metrics and Features

As initial research question, we address the performance comparison of attack metrics and meta data features. Beginning with the overall *global* performance, we get a first impression of the impact of confirmation attacks in generic scenarios. We continue with an analysis of *individual* combinations of metrics and features for all scenarios.

Global Performance

In our first evaluation step, we identify the overall best-performing metrics and features for a combination of all scenario setups. Figure 3.4 summarizes the attack success, i. e., the relative number of successful connection identifications, for all traces in the *DigesTor* corpus. Each box represents the full performance range of a metric/feature, whereas we focus on the comparison median (horizontal bar) results. We see that Mutual information (MI) provides the best overall result (median=0.48) in a global comparison. This result summarizes the attack success for all combinations of MI with the given traffic features and applies for all scenarios introduced in Section 3.3.2. In the comparison of meta data features the time to live field (ttl) performs best (median=0.44).

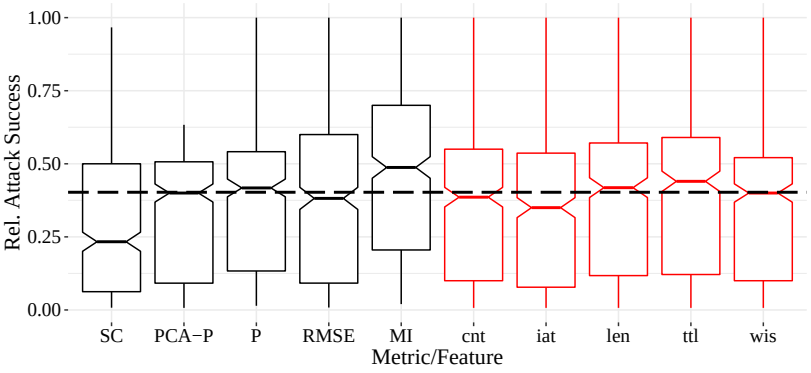
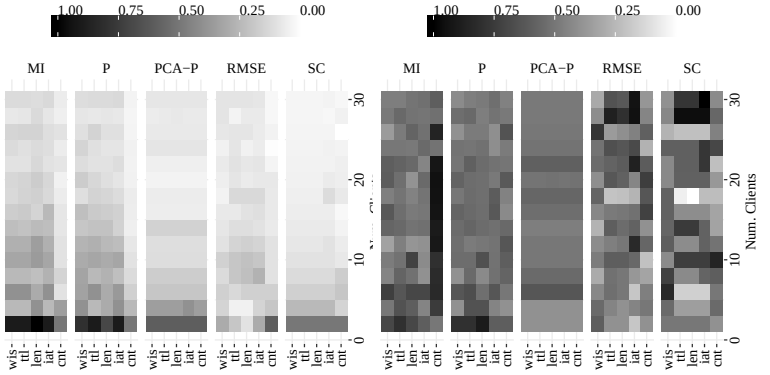


Figure 3.4. General attack success for topologies and applications combined with 2 clients to 30 clients. Summarizes the success of attacks, i. e., the relative number of successfully identified connections, (black) and traffic features (red) in comparison to random guessing (dashed line).

Individual Performance

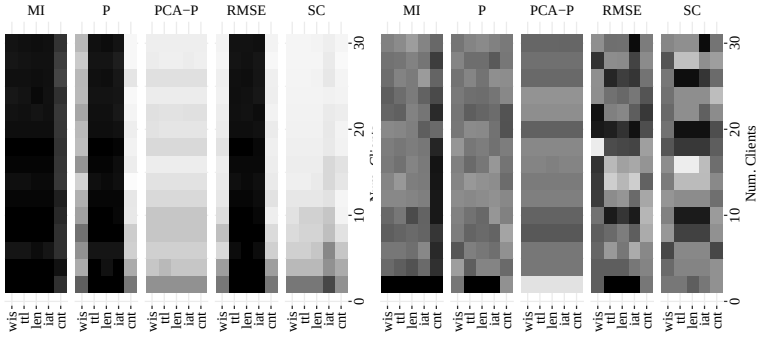
Figures 3.5, 3.6, and 3.7 highlight the performance of all specific combinations of metrics and features. Darker tiles in the heat map indicate a higher attack success at a specific experimental setup. Table 3.1 summarizes these results and provides an overview of the best performing metric and feature combinations for individual setups. We see that (`MI,iat`) performs best in a global comparison, i. e., it is the most robust combination while performing 23 % better than random guessing. Overall, `iat` is the most reliable meta data feature for most scenarios, whereas we see varying metrics for individual setups.

What Metric and Feature Combination Performs Best? Without any prior knowledge of the use case and number of concurrent transmissions, `MI`/`ttl` outperform an average random guessing attack. As soon as it is possible to adjust to a certain scenario, the targeted combination of a metric and feature helps to increase the improvement over random guessing.



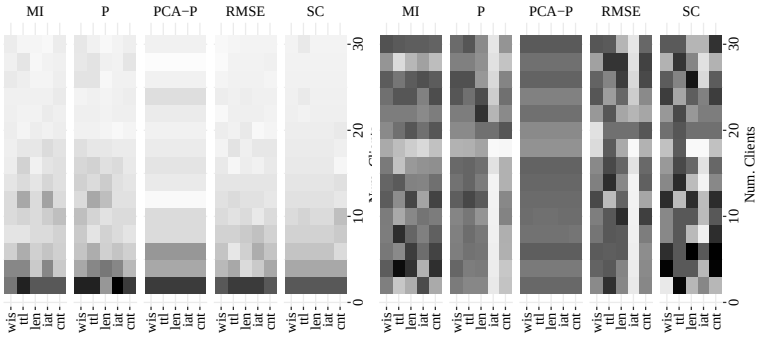
(a) Directed Static **(b)** Grouped Static

Figure 3.5. Average performance for static setup. The heatmap indicates the relative attack success, ranging from 0 → no success → lighter to 1 → high success → darker. Results show all attack metric and feature combinations.



(a) Directed Random **(b)** Grouped Random

Figure 3.6. Average performance for random setup.



(a) Directed Browsing **(b)** Grouped Browsing

Figure 3.7. Average performance for browsing setup.

3.4.2. Scenarios

Different topologies have two characteristics that influence the success of an attack. First, grouped setups, where n clients connect to only 2 individual servers, induce more noise through concurrent transmissions for traffic that is captured at the server. Such noise complicates the application of comparison metrics and destroys connection-individual parameters. One example for this is the attack success for a *random* download in the directed and grouped topology (cf. Figure 3.6). We see that it is possible to distinguish connections even for high user numbers in the directed setup (ΔRG=35 %), whereas we lose too much information in the grouped experiments (ΔRG=22 %). Second, the number of candidates for guessing a connection is limited to two serves in the grouped setup. Consequently, we experience more stable results for grouped topologies (AUC=0.5) than in directed setups (AUC=0.72) with overall more connection candidates.

What Scenarios Favor Attacks? Guessing on fewer candidates makes it easier to achieve favorable success rates for an attack. At the same time, it becomes harder to distinguish individual traffic characteristics through simple comparison metrics. Our results show that random downloads, where a high amount of data is sent in distinct patterns, provide the best improvement over an uneducated guess. In combination with a setup that reduces the

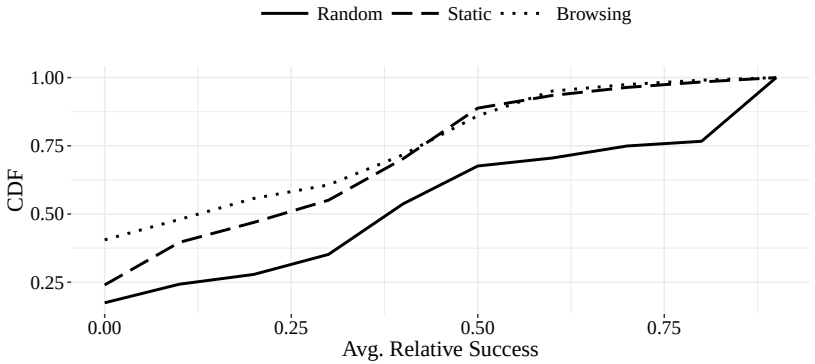


Figure 3.8. Attack success for three application types. The CDF indicates the best attack success for random downloads and shows only minor differences for static downloads or browsing scenarios.

noise of concurrent transmissions, this leads to a successful attack even for higher user numbers. The same does not apply to user-individual browsing, where traffic patterns are unique but the amount of data sent is insufficient for distinguishing connections reliably.

3.4.3. Countermeasures.

We can counter traffic analysis by perturbing traffic features during the transmission process. One example for this is mixing [ZFG+04], where intended delays for packets change the timing relations of a connection. As such countermeasures can decrease a system's performance, we analyze mixing concerning its protection capabilities and performance impairments.

Implementation.

We implement a mix within the Tor code and deploy it in the exit relay of our experimental setup. The mix delays TLS records within Tor before they are emitted for further transmission; it uses a defined delay duration (time held back) and rate (relative amount affected). TLS records are, within a

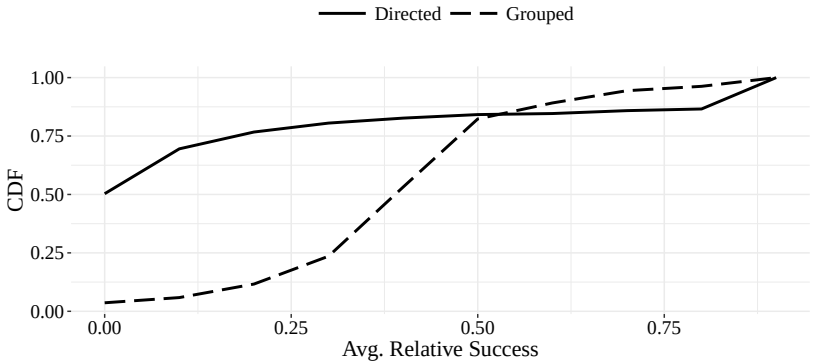


Figure 3.9. Attack success for two topologies. The CDF indicates a significantly better result for the grouped setup with a higher improvement over random guessing.

Tor relay, closest to the transport layer on which an adversary monitors connections. We, therefore, expect a maximum effect on traffic analysis attacks. In the following, we give an example for different mix delays (time added to sending of TLS records) and mix rates (a portion of records affected by mixing). The mix does not provide any differentiation of TLS records from different connections, e. g., mixing is applied to a fraction of *all* records in the relay.

Results.

At a static mix rate of 20 % (directed network topology, static download application), we achieve an AUC in the range of 0.9 to 0.95 for delay durations between 10 ms to 1 ms, which represents at least 20 % improvement over the unmixed attack success (AUC=0.72). At the same time, we see that varying mix rates do not influence the attack success significantly.

Moreover, we analyze the end-to-end delays for increasing mix delays at a fixed rate of 20 %. Results show slightly increased delays for mixed connections, while the performance impairments are still in an acceptable range.

Does Mixing Counter Attacks? Our results support the concept of mix-

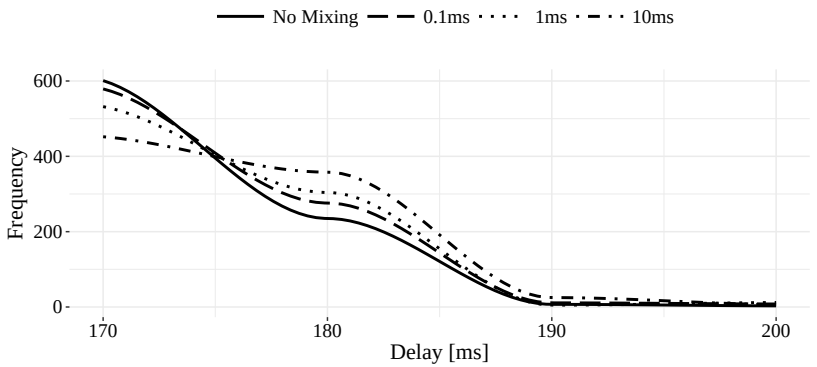


Figure 3.10. Distribution of end-to-end delays with applied mixing. Results show slightly increasing round trip times for mixed setups, where we tested a static mixing rate of 20 % and increasing mix delays.

ing, whereas the delays can only protect a subset of meta data features. The achieved obfuscation is sufficient for casual scenarios at an acceptable performance overhead, but at this price cannot guarantee *perfect* traffic analysis resistance.

3.4.4. Overview of Results

We summarize the results of our experimental evaluation as follows.

1. **Metrics and Features Combined.** For all topologies and applications we found a metric and feature combination that outperformed random guessing (Table 3.1). These combinations do *not* focus on a single traffic feature, hence, an isolated obfuscation of meta data features cannot protect against traffic analysis in general.

2. **Topologies and Applications.** Even though we found topologies and applications that hinder an attack, the attack framework outperformed random guessing attacks by 26 % on average (individual scenarios) and 23 % in generic scenarios.

3. **Affordable Countermeasures.** We use the comparative evaluation of *DigesTor* to demonstrate low-latency mixing as a countermeasure to traffic analysis attacks. Such effects can be achieved at minimal additional delays of 1 ms, which renders this solution an actual option for the live system.

3.5. Discussion

After demonstrating the experimental benefits of our traffic analysis framework, we now introduce how *DigesTor* can be used to support future research and what limitations the system faces at the moment. Furthermore, we discuss the ethical guidelines for this work and the potential of mixing as a countermeasure.

3.5.1. Goals of *DigesTor*

The goal of our evaluation framework is to accelerate the deployment of new defenses. To achieve this, we must provide a set of conditions which appropriately represent Tor's infrastructure, but also operate at sufficient scale to approximate the parameters of the real network.

How to use *DigesTor*?

The results of this work provide a first comparative overview of attack metrics and meta data features. Our work supports future research as follows.

- **Trace Corpus.** Our trace corpus represents standard topologies and application types and can be used to evaluate generic passive attacks without harming users of the live network. Furthermore, this once more supports the comparability of results.

- **Attacks.** The traffic analysis framework already provides a representative set of metrics and can be extended further by new attack metrics and meta data features. This allows comparing new approaches with the success of existing work.

- **Defenses.** Following the example of mixing as a countermeasure, future defensive research can use the performance comparison to assess the effects of novel countermeasures.

Limitations.

For the use case scenarios, we approximate real user behavior by simple models, e. g., through randomized web requests to a restricted set of sites or random download patterns. This does not represent the user behavior that defines the traffic patterns in a real-world scenario. In end-to-end confirmation attacks, a matching between client and server traces is the primary interest. Adding user models to the experimental setup in a future revision of *DigesTor* helps to create more realistic scenarios, but is not crucial for the technical evaluation of attacks.

3.5.2. Achieving Realism

We found that there are numerous factors which impact the ability to claim that a test network is representative of conditions in the real Tor network. Of particular importance is a replication of the number, diversity, and bandwidth distribution of connections experienced by different nodes, along with appropriate replication of the views available to various classes of adversaries.

We can address some of these issues by simply doing better than what is generally achieved in the live network since overall size and bounds are known. For example, our bandwidth reaches the 90th percentile of nodes, allowing us to claim general applicability for that metric. Other factors which are based upon diversity rather than scale, and in particular the variety of bandwidth distributions flows and circuits will experience, prove to be one of the hardest factors for a test network to replicate. While we address this issue through multiple different canonical traffic patterns, additional instrumentation of Tor to extract a better understanding of this distribution would allow for more efficient evaluations.

3.5.3. Ethics

In compliance with the Tor Ethical Research Guidelines [The19b], we designed this work in a way that does not harm users of the live network. We emphasize that especially the experimental evaluation of traffic analysis attacks can cause damage to real-world users and should always be conducted in a controlled environment. In turn, this applies to the analysis of countermeasure implementations whose security yet has to be proven.

3.5.4. Mix Countermeasure

Our TLS mix concept is implemented at exit nodes and can support a slow rollout over the existing network. Mixing of TLS records means there cannot be mixed and unmixed connections at the same time in one relay, reducing the unmixed bandwidth for the sake of increased security. However, not all nodes in the network must provide mixing, as a small fraction is sufficient to introduce uncertainty for an adversary across many active circuits. Along with the dynamic adaption of mix parameters, this makes the mix concept flexible: instead of using fixed setups, mix parameters can be coupled with monitoring the current network status and load.

3.6. Conclusion

Aiming for the *comparison* of existing traffic analysis attacks requires an initial insight into the current landscape of attacks on Tor. Prior work emphasizes the threat of passive traffic analysis but fails to provide a comparable experimental basis for the evaluation of each attack. *DigesTor* is an appeal to comparability in security research on Tor. The attack landscape of current research offers various classes of offensive work that *might or might not* pose a threat to the live Tor network. With *DigesTor* we share two core features: We generated a first traffic analysis corpus of this kind that we share to support the comparability of future research. The second core feature is the Traffic Analysis Framework, which applies a set of recent attack techniques for comparative performance analysis. To demonstrate the benefits of *DigesTor*, we analyze mixing as a potential countermeasure against passive

traffic analysis attacks. Our results indicate that mixing hinders the success of otherwise successful confirmation attacks.

Drei Uhr, das ist immer zu spät oder zu früh für alles,
was man machen will.

— Jean Paul Sartre

4
Geographical Avoidance

Contents

4.1. Introduction

Tor's anonymity protects the actions of a broad ethical spectrum ranging from benign to criminal use cases. Both directions serve as motivation for censorship authorities [WL12] as well as law enforcement agencies [Sec19] to hinder the use of Tor and to monitor what is going on in the "dark parts" of the Internet.

Users can circumvent blocked Tor access in different ways [FLH+15], but they never know if someone analyzes their traffic [MD05, GH12, LRWW04]. Low-cost countermeasures do not sufficiently protect meta data [DCRS12] and obfuscating traffic against correlation leads to per-packet delays [KP18] or additional traffic [DP04]. However, we gain trust in a connection by *avoiding* paths through critical countries. Such circumvention becomes even more important since we know that, e. g., monitoring a circuit's middle relay is already sufficient to identify onion services [JJG+17]. Sophisticated path selection [AYM12, BW16a] is a starting point for this approach, but systems tend to focus on performance features [WBFG12] rather than geographical characteristics.

DeTor [LHL17], proposed by Li et al. in 2017, makes an attempt to provide *provable geographical avoidance* of untrusted countries. Provable avoidance means that it is impossible for an established Tor circuit to have traversed a forbidden area. This does not only apply to the avoidance of *relays* located in a specific country, but also considers the *Internet routing* between the client and server. DeTor uses an approach comparable to the principle of distance bounding: instead of depending on hardware solutions [ABF+08] or the modification of routing protocols [NWN+11], it uses the Round-Trip Time (RTT) of a connection and compares it to a theoretical lower bound for reaching the forbidden area [GZCF06,KBJK+06,PS01,WBF+11,WSS07]. The lower bound is estimated using the geographical locations of relays in the circuit and utilizes the fact that transmissions through the Internet can never be faster than approximately $^2/_3$ of the speed of light [LHL17].

Unfortunately, several fundamental design flaws hinder DeTor from providing a convincing solution for geographical avoidance. (*i*) DeTor does not consider the diverse network infrastructure of Tor and the underlying network,

e. g., it applies one static decision threshold for *all* circuits. Tor's skewed distribution of relays leads to various circuit lengths that cannot offer the same performance features for all users. Applying the same threshold even for varying connection characteristics leads to overly restrictive avoidance decisions. Furthermore, (*ii*) DeTor makes false assumptions on the available ground truth information. In particular, it assumes symmetric routes, miscalculates the distance within the lower bound detection mechanism, and ultimately accepts connections traversing forbidden areas. This contradicts the "provable" security guarantee for geographical avoidance. DeTor accepts external GeoIP information without any further verification and overlooks the chances of using false locations as the foundation for a decision. Finally, (*iii*) DeTor was designed without considering the constraints of real-world deployment. By sending timing probes through the *entire* circuit, the system reveals the connection endpoints even before we can be sure about the security of this circuit. This opens up new attack vectors instead of protecting against potential threats. We argue that all of these flaws are unnecessary and introduce strict security and performance issues that render the system unusable for an actual deployment.

In our work, we approach the problem of geographical avoidance systematically and begin with a definition of its *challenges*. We introduce three classes of challenges that we identify as the general pitfalls of geographical avoidance, namely, the demanding characteristics of Tor's (*i*) *network diversity*, the lack of trusted (*ii*) *ground truth* information, and the requirements of a real-world (*iii*) *deployment*. Tackling these challenges, we propose a new timing-based avoidance system that overcomes design flaws of existing systems. We back up the theory of these challenges with a preliminary assessment of Tor's network infrastructure and the transmission characteristics of the underlying network. Our results show that the skewed distribution of Tor relays that we measured empirically not only leads to different levels of anonymity for users but also affects the essential end-to-end timing of messages sent through the network. Ignoring this diversity means to oversimplify the decision process with consequences for either Tor's security or performance. We find that accepting external GeoIP information as ground

truth for relay positions is error-prone and can impact geographical avoidance decisions. False locations would imply propagation speeds that exceed the speed of light and, with that, are provably wrong from a physical perspective. We verify the GeoIP information and identify false locations by applying this physical proof to improve the information through trilateration [JSM+18].

The assessment of challenges is our foundation to propose technical solutions and design a new, empirical avoidance concept. *Empirical avoidance* has two main benefits. First, it allows considering hop-individual transmission characteristics rather than one static threshold for different connections. Consequently, we can apply avoidance decisions concerning the various performance characteristics of Tor and step back from the collateral damage of overly restrictive decisions. Second, we derive the hop-individual timing estimates from distributed measurements of several reference points. This distributed approach adds another level of security, as it allows to represent single connections through empirical data that cannot be manipulated by an adversary prolonging messages [vH05]. In a first evaluation step, we analyze the performance of our novel avoidance concept and compare it to existing approaches.

In a second step, we introduce the prototype implementation *TrilateraTor* that is the first also to satisfy the requirements of a real-world *deployment*. *TrilateraTor* introduces a novel measurement technique that derives a circuit's end-to-end timing directly from the `NTor` handshake in Tor's circuit establishment procedure. As the establishment of several ready-to-use circuits is part of Tor's start up procedure, the use of *TrilateraTor* neither induces any delays through preliminary probing nor information leaks. Along with additional verification steps for untrusted ground truth information and the less restrictive empirical avoidance concept, *TrilateraTor* provides real answers to the challenges of geographical avoidance. We analyze the performance of our prototype implementation in another empirical simulation study, discuss the operational steps of a practical deployment, and provide a detailed security analysis. Our contributions focus on our second research question.

Research Question 2 *Do alternative countermeasures find a better compromise in the security and performance trade off for a long-term defense?*

- **Challenges of Geographical Avoidance.** We assess the problem of geographical avoidance in Tor and identify three classes of challenges. These classes address the *diversity* of Tor's infrastructure and the underlying network, the lack of *ground truth* information, and the constraints arising from the real-world *deployment* of an avoidance system.

- **Preliminary Measurements.** We conduct an empirical evaluation of Tor's infrastructure to confirm the relevance of the above challenges. Our results show that a skewed relay distribution cannot provide the *same anonymity for all* and can limit the success of an avoidance system. Furthermore, we identify an unverified assumption that hinders DeTor [LHL17] from providing *provable* avoidance.

- **Experimental Evaluation**. Starting from the assessment of the given network infrastructure, we introduce solutions for the set of challenges and compare their performances in an empirical simulation study. In a second step, we propose, implement, and evaluate *TrilateraTor*, our approach to take the constraints of a real-world deployment into account.

The above contributions result from a collaboration with Kai Jansen, David Rupprecht, Thorsten Holz, and Christina Pöpper. In particular, Kai Jansen contributed to the trilateration based localization of relays. David Rupprecht contributed to the distributed network measurements.

4.2. Background

Before we define fundamental challenges for avoidance and provide possible solutions for system deployment, we introduce some background on the context of geographical avoidance. This includes the technical background of trilateration, the attacker model relevant in this context, and a brief overview of the functionality of DeTor.

4.2.1. Trilateration

As a means to geographically localize Tor relays, we make use of trilateration. This technique is based upon measured distances to multiple known reference points. This general approach is used in, e. g., satellite navigation systems (such as GPS) or to determine the location of mobile phones in radio cells and utilizes time or signal strength differences between reference points [JSM+18]. In the context of geographical avoidance, we use the round-trip times from one node to multiple reference servers to derive hop-individual time references.

The underlying theoretical model can be summarized as follows. The unknown location of a relay $\overrightarrow{R}$ is denoted by (x, y), i. e., a position is defined by its latitude and longitude coordinates and neglects altitude information for the sake of simplification. As references, we use RTT measurements from n other nodes $\overrightarrow{S}_1, \overrightarrow{S}_2, \ldots, \overrightarrow{S}_n$ to $\overrightarrow{R}$. As a result, we obtain n RTTs $t_1, t_2, \ldots, t_n$ between known references and the respective relay. These timings are related to geographic distances considering a typical transmission speed v of up to $^2/_3$ of the speed of light. Having three or more geographic distances allows us to pinpoint the target by intersecting circles with radii corresponding to the estimated distances.

Due to noise in RTT measurements, the resulting circles do not intersect in a distinct point but rather mark a target area. To find the most likely position $\overrightarrow{R}$, we use a weighted root-mean-square error approach. This correction technique optimizes the result towards the minimal error for all reference measurements:

$$\arg\min_{\overrightarrow{R}} \sqrt{\frac{\sum_{i=1}^{n} \left[\left(dist(\overrightarrow{R}, \overrightarrow{S}_i) - t_i \cdot v \right) \cdot \omega_i \right]^2}{n}}, \tag{4.1}$$

where ω_i is a normalized weighting factor based on the distance to the reference. In particular, smaller RTTs are expected to be less affected by noise and consequently have higher weight in the error minimization process. The output of Equation 4.1 is the most likely relay position with minimal error.

4.2.2. Attacker Model

We assume a nation-state adversary of one single country and assume, for the sake of simplicity, this adversary to only act within the borders of this country. The adversary follows the goal of gathering sensitive information about users or connections, e. g., for the generation of specific profiles or evaluation of behavioral patterns both delivering crucial information for follow-up attacks. To this end, the adversary conducts active or passive traffic analysis attacks on traffic monitored in one or more nodes falling into the reach of the adversarial country. While traffic analysis helps to learn sensitive information, the adversary can further hold back transmissions to manipulate the timing characteristics of ICMP and TCP packets, e. g., to influence the localization of relays (cf. Section 4.3.2) or the computation of circuit relations (cf. Section 4.5.3). Routing attacks [SEV+16] are another way to force traffic through adversarial areas.

The adversary aims for learning sensitive information about users and does so by analyzing traffic routed through network devices under adversarial control. As a countermeasure to this, geographical avoidance tries to identify connections through the untrusted area to offer an assessment of the circuit's security and alternative connections.

4.2.3. DeTor

In 2017, Li et al. [LHL17] proposed DeTor as a system to provide *provable geographical avoidance* in Tor. The core principle is comparing the measured RTT of a Tor circuit with a lower bound threshold that includes the trip to the forbidden area. If the measured RTT does *not* exceed the threshold, the respective forbidden area could not have been reached. In other words, the additional distance, and hence time, required to traverse the forbidden area is higher than the measured RTT would allow. This concept was originally introduced in the context of Alibi Routing [LLV+15], where single hops were checked and later extended to three-hop connections to fit the needs of Tor.

When estimating the lower bound, DeTor first calculates the minimal geographical distance D_{min} required for routing through the forbidden area and, second, relates it to a transmission speed of $^2/_3$c, which is an estimation of

the maximal speed of Internet connections. Considering an established Tor circuit, DeTor calculates the following threshold:

$$R_{min} = \frac{3}{2c} \cdot min \begin{cases} 2 \cdot D_{min}(c, F, e, m, x, s) \\ 2 \cdot D_{min}(c, e, F, m, x, s) \\ 2 \cdot D_{min}(c, e, m, F, x, s) \\ 2 \cdot D_{min}(c, e, m, x, F, s) \end{cases}, \tag{4.2}$$

where c, s are client and server, e, m, x are entry, middle, and exit relays of Tor, and F is the forbidden area. To obtain geographical positions, DeTor performs a Geo IP lookup with the respective IP address of relays.

When deciding whether a Tor circuit avoided a forbidden area, a binary decision on the measured RTT R_{e2e} is performed against the calculated threshold R_{min}:

$$avoided = \begin{cases} 1, & (1+\delta) \cdot R_{e2e} < R_{min} \\ 0, & (1+\delta) \cdot R_{e2e} \geq R_{min} \end{cases}, \tag{4.3}$$

with δ being a static overhead parameter in the range between $[0, 1]$ designated to compensate network inconsistencies and measurement noise. Whenever a measured RTT R_{e2e} is shorter than the DeTor threshold R_{min}, the circuit is proven to avoid a forbidden area.

4.3. Challenges

We begin our work with a systematic evaluation of the challenges of geographical avoidance, i. e., we identify fundamental influencing factors that define the performance *and* security of an avoidance system. We introduce three classes of challenges (cf. Table 4.1), namely (*i*) *network diversity* that leads to heterogeneous transmission characteristics, (*ii*) a lack of *ground truth* information that complicates avoidance decisions, and (*iii*) the restrictions of a realistic *deployment*. In the following, we provide a detailed introduction of these three classes of challenges and complement our theoretical claims with the results of a preliminary measurement study that addresses the characteristics of Tor and the underlying network. Throughout this work, the set

Table 4.1. Challenges of Geographical Avoidance.

Class	Challenge	Solution/Design Goal	Section
Network Diversity	*Relay Distribution*	Prevent Collateral Damage	§4.3.1
	Connection Lengths		§4.3.1
	Connection Failures		§4.3.1
Ground Truth	*Relay Locations*	ICMP Reference, Update	§4.3.2
	Asymmetry	Single Extension	§4.3.2
	Transmission Characteristics	Individual Estimates	§4.3.2
Deployment	*Performance*	Evaluation	§4.6.4
	Information Sources	Circuit Est. Timing	§4.5.3
		Distributed Measurements	§4.6.1
	Security	Security Analysis	§4.6.3

of challenges will guide our design of a new avoidance concept and later also dictate the requirements that a prototype implementation must satisfy.

4.3.1. Network Diversity

Tor's circuit establishment procedure and the transmission characteristics of the underlying network directly influence the end-to-end timing of transmissions. Diverse network infrastructures can be one crucial influencing factor for such varying conditions that have significant consequences for timing-based avoidance systems. We identify the following avoidance challenges related to network diversity.

1. **Skewed Relay Distribution.** The worldwide distribution of Tor relays is skewed towards countries with a higher number of Tor supporters. *The biased relay distribution can induce performance impairments when an avoidance decision excludes a high number of nodes.*

2. **Connection Lengths.** The network infrastructure might enforce certain routes for a connection, e. g., in cases where the path between

Table 4.2. Consensus Statistics.

		DE	FR	NL	EU RU	GB	SE	UA	NA US	CA
Relays	[%]	19	13	8	5	3	2	2	18	2
Bandwidth	[%]	22	23	13	2	3	2	NA	11	2

countries is forced to travel through a trans-Atlantic cable. *Along with the skewed distribution of relays this influences the length of a connection, which also affects its timing characteristics.*

3. **Connection Failures.** Permanent and temporary partitions in the network infrastructure affect the availability of routes and different circuits. *Such partitions force traffic through specific routes and can hinder the avoidance of a forbidden area.*

All these characteristics address the complex infrastructure of Tor. An avoidance system must incorporate such varying transmission characteristics and provide a flexible decision mechanism that reduces the negative effects of incorrect decisions. A wrong decision can lead to accepting critical connections that might harm the user, or they introduce collateral damage in cases where conservative security is preferred over performance.

Skewed Relay Distribution

We analyze the characteristics of one consensus file and derive the distribution of relays and their performance (cf. Table 4.3 for an overview of all experiments). The majority of Tor relays runs in Europe, where multiple countries are located within a comparably small area (cf. Table 4.2). This influences the choice of relays and renders avoiding specific countries within Europe more challenging. We see that 72 % of all relays are operated in the `EU`[1] and 21 % run in `NA`; the remaining 7 % are distributed over all other continents. The same applies to the bandwidth offered, i. e., `EU` provides 81.5 %

[1] `NA` - North America, `EU` - Europe, `AS` - Asia, `SA` - South America, `OC` - Oceania, `DE` - Germany, `FR` - France, `UA` - Ukraine, `NL` - Netherlands, `GB` - United Kingdom, `SE` - Sweden, `US` - USA, `CA` - Canada, `RU` - Russia, `IN` - India, `SG` - Singapore, `BR` - Brazil

Table 4.3. Overview of Experiments and Parameter Setups.

Protocol	Target	Servers	Nodes	Duration	Num. Results	Purpose	Section
ICMP	Relays	16	6042	20 h	1,837,761	Violations	§4.3.2
						Trilateration	§4.3.2
						Propagation Speed	§4.3.2
		8	6042	20 h	27,274	Reference 1	§4.6.2
					62,643	Reference 2	§4.6.2
TCP	Weighted Circuits	8	Random	7 d	135,924	Length	§4.3.1
	Artificial Circuits	8	150	14 d	360,395	Hop Estimates	§4.4.1
			150		360,395	Simulation	§4.4.2
			150		134,370	Reference 1	§4.6.2
					223,070	Reference 2	§4.6.2
Ntor	Artificial Circuits	8	1945	4 d	104,889	Handshake Time	§4.5.4
			3724			Circuit Failure	§4.3.1
			893				

Protocol Network protocol used in measurements. *ICMP* messages sent as standard ping, *TCP* messages sent through Tor circuits using a reply server, *Handshakes* are offsets between initial and final `NTor` handshakes.
Target Where probes were sent to. *Relays* are single relay nodes from the consensus; *weighted circuits* are Tor standard circuits where we do not interfere with the relay selection; *art. circuits* are artificial circuits we build from selected relays using the control port.
Servers Number of servers we use for conducting measurements. *ICMP* measurements originate from all servers, as they do not depend on a reply server; *TCP* measurements depend on a reply server and we split the set of servers into senders and receivers
Nodes Number of nodes addressed in a measurement; we use a filtered consensus where all relays provide the `Stable` and `Running` flags. We use this filtered consensus to analyze the distribution of relays (Section 4.3.1). *ICMP* measurements send pings directly to these nodes; *TCP* measurements use Tor circuits, hence, the nodes summarize the relays used to build circuits. Weighted circuits are built from whatever Tor selects, artificial circuits are built from permutations of $m \times n \times l$ entries, middles, exits.
Duration Time elapsed between first and last measurement in a batch of measurements. Might include several repetitions of the same measurement.
Num. Results Total number of individual results.
Purpose What we use the results for.
Section Parts of this paper that utilize these measurements.

of the overall bandwidth, `NA` has 17 %, and all other continents provide no more than the remaining 2 %. As the Tor relay selection is weighted towards higher bandwidth nodes, we find the most prominent choices for relays in Europe.

The distribution of relays influences the overall length of circuits, which determines the transmission times between clients and servers. Furthermore, a higher density of relays *and* countries makes it harder to distinguish between different countries. This leads us to an evaluation of the expected *connection lengths*.

Design Goal: Overly restrictive avoidance decisions cause collateral damage in regions with a high density of relays and countries. We must provide a decision mechanism that does not exclude too many relay choices.

Table 4.4. Overview of Empirical Circuit lengths.

		EU-EU	EU-NA	NA-EU	EU-AS	NA-NA
Median	[km]	4384	11,117	12,394	12,897	19,210
Minimum	[km]	318	8425	6411	10,329	16,907
Maximum	[km]	40,630	45,436	44,807	44,094	51,092

Connection Lengths

The length of a circuit depends on the client/server location and the distribution of relays involved; the overall distance traveled in a circuit influences the RTT of a transmission. We define the length of a circuit as the distances $(client, entry) + (entry, middle) + (middle, exit) + (exit, server)$ and use approximate direct connections between all nodes from client to server of Tor standard circuits.

The shortest circuits are built within Europe and the longest circuits in North America (cf. Table 4.4 for reference, combinations of continents describe the client and server locations). A closer look at the relay locations for all `(NA,NA)` circuits reveals that none of the entry relays was located in `NA`, only 14 % of circuits had a middle relay in `NA`, 27 % had an exit in `NA`, and only 4 % of all circuits went through a middle *and* exit in `NA`. We must assume that even though we established a connection that was limited to `NA`, the circuit traversed the Atlantic twice, which results in a high average circuit length.

Design Goal: Varying circuit lengths lead to individual timing characteristics. The decision threshold of an avoidance system should consider individual characteristics for precise decisions on different connections.

Connection Failures

Partitions in the network infrastructure influence the circuit establishment procedure on the application layer and the selection of routes on the network layer. This influences the transmission path of a circuit and eventually its end-to-end timing features. As temporary and permanent connection issues can be caused by a wide range of reasons that are nontransparent for an

avoidance system, we limit our analyses to a summary of monitored circuit establishment failures.

In our measurements, overall 11 % of circuit establishments failed (12,500 out of 105,889 circuits). We use the consensus archives to check whether a relay was unavailable during the circuit establishment and distinguish two cases: A relay might *not* be documented in the consensus and we consider it as completely unavailable, or the relay occurs in the list of relays and we can check its flags. Our results reveal 20 % of relays that caused the connection failure to have the `Stable` flag set (router is suitable for long-lived circuits), whereas all failure relays are flagged as `Running` (router is currently usable). We find 9 % of failing relays to be entry guards and 29 % to be flagged as an exit. On average, failing relays provide 38.8 Mbit/s of advertised bandwidth (on average a relay provides 63 Mbit/s) and are 3122 km away from the preceding node in the circuit (cf. Table 4.4 as reference for total circuit lengths). The overall rate of circuit failures is non-negligible and further influences the circuit buildup procedure.

Design Goal: Connectivity issues and partitions amplify the effects of Tor's skewed relay distribution. Decision thresholds must be flexible to respect diverse performance features.

4.3.2. Ground Truth Information

Transmissions through Tor and the underlying network infrastructure are not transparent. Therefore, we lack trusted ground truth information about precise relay locations, all hops of the transmission path, or performance features given at the time of transmission. Nevertheless, we depend on a specific set of information to provide a profound avoidance decision. The lack of ground truth information introduces the following challenges.

1. **Relay Locations.** We have no reliable information about the actual positions of relays. GeoIP databases claim to provide accurate information on a country level, nevertheless, such databases are an untrusted source of information [Tor19a]. Manipulated or false entries that suggest an incorrect country code are a security threat. *Reference mea-*

surements help to verify GeoIP information and provide an additional source of information to identify false country codes.

2. **Asymmetric Paths.** Routing between the client and server is not performed on symmetric paths, but routes can differ on the way forth and back. *Assuming symmetric paths induces an error in the application of an avoidance decision.*

3. **Transmission Characteristics.** Dynamic routing might change the paths of a circuit between individual transmission sessions. Furthermore, varying network conditions can influence transmissions through the effects of congestion. *Assuming static characteristics introduces inaccuracies in the avoidance system.*

Even though we cannot gain full transparency in the transmissions, preliminary measurements and verification steps help to achieve more trust in the available data. In the following, we introduce a verification mechanism for GeoIP location information, identify the security threat of assuming symmetric paths, and estimate the dynamics of varying transmission characteristics.

Relay Locations

Prior avoidance systems use lower bounds to decide whether it is possible that a circuit traverses a forbidden area and for this, the locations of relays must be known. The consensus does not provide coordinates for relays, so the best way to estimate their position is an IP address lookup in a GeoIP database. Unfortunately, such databases provide untrusted information that might lead to false locations [GZPH16,HF+11,SZ11], e. g., we find up to 29 % of disagreement for city-level router locations [GSH+17]. We conduct reference measurements, similar to the approach of Weinberg et al. [WCSG18], to *verify* GeoIP locations and identify potential errors.

We measure the ICMP round-trip time between different remote servers and all relays of one consensus (cf. Table 4.3 for an overview of all experiments). In a first evaluation step, we compare the transmission time with the great circle distances between servers and relays to approximate the transmission speed in each measurement (cf. Figure 4.1). We use this speed estimate

to identify provably false GeoIP locations, i. e., locations that lead to propagation speeds faster than the speed of light. Such a violation occurs in cases where the GeoIP location documents a position that is further away from our reference point (remote servers) than indicated. Consequently, the measured time is too short to travel the entire distance to the recorded position. As we use multiple worldwide server instances, we receive reference measurements from opposing points and identify false information as soon as at least one server indicates a speed of light violation.

From all tested relays, we find approximately 6 % relays (330 out of 6042) to exceed the maximum allowed propagation speed and, consequently, to be represented through false GeoIP information. Using trilateration, as introduced in Section 4.2, we utilize the reference measurements from our server instances to update the position of obviously false relay locations. Besides the improved location, we update the country code of 3 % (194) of the relays.

Solution: We use ICMP reference measurements to verify the general correctness of untrusted GeoIP locations and identify obvious false locations that violate the speed of light. Trilateration allows us to improve the location data by removing provably false information.

Asymmetric Paths

We acknowledge the general approach of the recently proposed DeTor [LHL17] system, but find—besides further security and performance issues—a critical overestimation in its lower bound decisions. DeTor bases its mechanism on *symmetric* routes, which is not a valid assumption as has been discussed and demonstrated by Sun et al. [SEV+16]. We use this as an example of the consequences of a false asymmetry assumption. In particular, DeTor calculates its decision threshold based on a detour to the forbidden area on *both* directions of a round-trip. This is a critical misconception introduced when the authors transitioned their technique from one-way connections [LLV+15] to Tor circuits. A negligent doubling of the necessary distance overestimates

the required time. To fix the symmetric routes apparent in DeTor's time estimation (cf. Equation 4.2), we consider asymmetric routes and obtain:

$$R_{min} = \frac{3}{2c} \cdot min \begin{cases} D_{min}(c, F, e, m, x, s) \\ D_{min}(c, e, F, m, x, s) \\ D_{min}(c, e, m, F, x, s) \\ D_{min}(c, e, m, x, F, s) \end{cases} + D(c, e, m, x, s). \tag{4.4}$$

The amount of overestimation done by DeTor can be quantified as:

$$R_{error} = \frac{3}{2c} \cdot [D(A, F, B) - D(A, B)], \tag{4.5}$$

where A and B are the hops with an extension to reach F. DeTor overestimates the decision threshold by R_{error}, which represents the range of false decisions. The greater the distance to F, the higher the overestimation done by DeTor. This constitutes a critical security flaw as connections are falsely labeled secure, creating the illusion of protection from being monitored, and putting users to risk. DeTor uses an uncertainty parameter δ that can be used as a factor to adjust the measured RTT (cf. Equation 4.3), nevertheless, this does not fix the system-intrinsic overestimation.

Solution: We consider only one forbidden area extension for the entire connection, i. e., assume asymmetric paths.

Transmission Characteristics

Varying transmission characteristics influence the end-to-end timing of a connection, e. g., congestion in relays or routers prolongs the transmission times and can lead to false avoidance decisions. Consequently, the timing characteristics of a circuit depending on the distances between hops and the amount of routing that takes place in between. We utilize the ICMP reference measurements to review real-world timing characteristics and derive the experienced propagation speeds.

Three "clouds" of points (cf. Figure 4.1) summarize the transmission distances from remote servers to all relays in the consensus and indicate sparse areas like oceans and continents with only a few relays. We apply a nonlinear

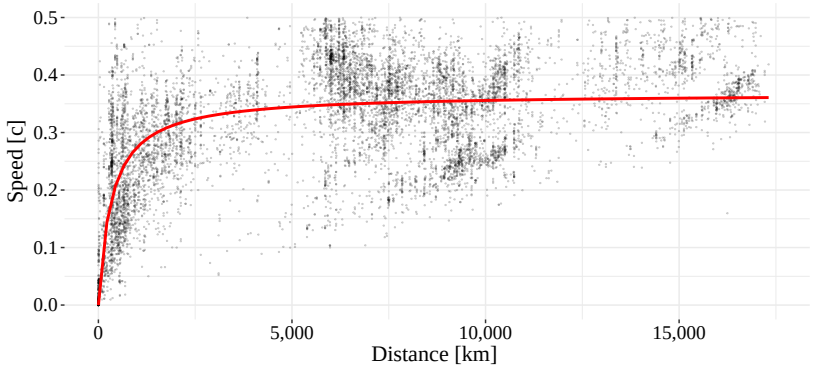


Figure 4.1. Empirical analysis of propagation speeds. We measure the propagation speeds versus traveled distances (points) of ICMP measurements from 16 reference servers to 6042 Tor relays; nonlinear least squares fit (NLS) of the relation between transmission distance and propagation speed (red line).

least squares fit (NLS) to these measurement results and receive the propagation speed as a function of the transmission lengths. The fitting function indicates a dynamic propagation speed rather than a fixed threshold, e. g., we find varying transmission characteristics for different circuit and hop distances. Typical transmission speeds are in the range of 0.22c to 0.67c [KBJK+06], whereas we see a maximum speed of $0.381c$ and a mean of $0.342c$ in the NLS fit of all ICMP measurements.

Solution: We use hop-individual timing estimates for all possible pairs of relays and step back from a static speed assumption for all connections.

4.3.3. Deployment

Deploying an avoidance system in the real-world means that we need reasonable sources of information for an accept or reject decision. Furthermore, we must maintain the original security and performance features of Tor, as otherwise new attack vectors open up. We define the following requirements for a realistic deployment.

1. **Gather Required Information.** All information required to perform a profound avoidance decision must be made available for Tor users. *For a realistic deployment we need a reasonable source of this information and generate trust in its content.*

2. **Security.** Gathering information for the avoidance decision must maintain the original security features of Tor. *The system must avoid actions that reveal any sensitive information about users or the network.*

3. **Performance.** Additional security through geographical avoidance might justify minor performance impairments, nevertheless, *it remains a design goal to maintain the original performance.*

A real-world deployment leads to additional requirements for the features of an avoidance system, e. g., they define the amount of information we can (or cannot) incorporate in the decision process, and they also dictate the security and performance features that must be satisfied. Even though they are the conditions for the practical deployment of a system, the *deployment* challenges are still independent of the general concept of an avoidance system. In other words, it is possible to propose a general avoidance concept that answers the challenges of missing *ground truth* information and follows the design goals associated with *network diversity* in a first step. As soon as the general avoidance concepts can satisfy these challenges, we can approach the subsequent step of deriving a prototype implementation that also serves all real-world conditions.

We organize our evaluation procedure according to this two-step workflow. In a first evaluation step, we introduce an empirical avoidance concept and rate its detection capabilities and potential collateral damage in comparison with recent proposals in this context. This initial assessment provides an overview of how different concepts can manage the challenges. In a second step, we extend the experimental setup by real-world constraints and introduce a prototype implementation that utilizes the empirical avoidance concept of Section 4.4.

4.4. Simulation Study

The assessment of challenges (Section 4.3) is our starting point to evaluate building blocks for a realistic avoidance system. In the following, we introduce an empirical avoidance concept and its system model and compare it with recent work in this context.

4.4.1. Empirical Avoidance Decisions

From the preliminary measurements we learned that Tor not only provides a skewed distribution of relays (Section 4.3.1), but also that varying transmission characteristics (Section 4.3.2) and circuit lengths (Section 4.3.1) have a fundamental influence on the end-to-end timings of circuits. Consequently, we lose information when applying a *static* threshold in the avoidance decision. In the following, we propose an alternative approach to estimate the timing characteristics of each hop individually.

Relay Hop Time Estimation

Our goal is to obtain a realistic estimation of transmission times between individual hops. We do so by extracting dependencies of circuits that share the same hops. In particular, we analyze RTT measurements of Tor circuits that we gather from remote probing servers. We build these circuits from permutations of entry, middle, and exit relays such that they share pairwise identical hops. This redundancy of circuit segments allows us to estimate the timing distribution that each hop contributes to the overall circuit's RTT.

We aim to create a map of RTT relations between all possible combinations of Tor relays:

$$optimize \begin{pmatrix} RTT(c \rightarrow \mathbf{e_1} \rightarrow m_1 \rightarrow x_1 \rightarrow s) \\ RTT(c \rightarrow \mathbf{e_2} \rightarrow m_1 \rightarrow x_1 \rightarrow s) \\ \dots \\ RTT(c \rightarrow e_1 \rightarrow \mathbf{m_1} \rightarrow x_1 \rightarrow s) \\ RTT(c \rightarrow e_1 \rightarrow \mathbf{m_2} \rightarrow x_1 \rightarrow s) \\ \dots \\ RTT(c \rightarrow e_1 \rightarrow m_1 \rightarrow \mathbf{x_1} \rightarrow s) \\ RTT(c \rightarrow e_1 \rightarrow m_1 \rightarrow \mathbf{x_2} \rightarrow s) \\ \dots \end{pmatrix}, \tag{4.6}$$

where e_m, m_n, x_l are combinations of relays and c, s are the remote servers we measure circuits from. In the above notation, the hops partially overlap, which allows us to define equal segments throughout all measurements. The dependency between measurements allows us to assign portions of the total RTT to individual hops. Notably, the measurements take the Tor and other network overhead into account, resulting in hop time estimations already including realistic overhead metrics.

We define an objective function, which minimizes the error for all combinations of measurements, as shown in Equation 4.7:

$$\min_x f(x) = \|\mathbf{A}x - b\|, x \geq 0, \tag{4.7}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a design matrix we arrange from our measurements, and $b \in \mathbb{R}^m$ is the vector of observations [KSD13], i. e., the measured RTTs. The design matrix is arranged as follows:

$$\begin{array}{c} \\ m_1 \\ m_2 \\ \vdots \\ m_x \end{array} \begin{array}{c} \begin{array}{cccc} n_1 \rightarrow n_2 & n_1 \rightarrow n_3 & \dots & n_y \rightarrow n_z \end{array} \\ \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \dots & 0 \end{pmatrix} \end{array} \tag{4.8}$$

Here, the rows contain individual measurements m and the columns represent *all* pairs of nodes n, i. e., hops between relays that occurred in the measurements. A 1-entry denotes that the measured circuit contained this specific hop, whereas a 0-entry is assigned to all other hops. In total, a maximum of four 1-entries can exist in each row, as this represents the number of hops from the client to the server.

Equation 4.7 represents a non-negative least-squares (NNLS) problem, which is a constrained version of a least squares problem. The sheer size of the problem, i. e., several thousand measurements and tens of thousands of relay combinations, exceeds the processable complexity by magnitudes. Nevertheless, applying an optimized solver [KSD13, MFLS17] and the fact that we are dealing with a very sparse design matrix allows us to handle large-scale problems. We implement such a solver to calculate the timing distribution of all hops minimizing the squared error. As a result, we receive a lookup table that provides pairwise estimates for all relays in the consensus.

Forbidden Area Decision

The hop estimations are our basis to calculate the time it would take to send data through a forbidden area. In particular, we measure the RTT R_{e2e} for a newly built circuit and identify the involved relays. From these relays, we compute the decision threshold R_{est} that summarizes the expected transmission time for the current circuit. Our approach follows the concept of DeTor (cf. Equation 4.2), but uses the empirical estimates instead of translating great circle distances into a lower bound transmission time. We compute the decision threshold R_{est}, the shortest possible extension ext_F to the forbidden area, the hop estimates to send data from the client to the server (excluding the hops involved in the extension), and the estimates for the way back from the server to the client. First, we compute the shortest possible extension:

$$ext_F = min\left\{\frac{D(A,F,B)}{avg[S(A,F),S(F,B)]}\right\}, \tag{4.9}$$

where $D(A,F,B)$ denotes the great circle distance D from a node A over the forbidden area to the next hop B. As we cannot know the exact propagation

speed for the extension to the forbidden area F and nodes A, B, we approximate the extension using the average empirical speed $avg[S(A,F), S(F,B)]$ of all RTT measurements between the respective countries that summarizes the propagation speeds to $S(A,F)$ and from $S(F,B)$ the forbidden area. If for example the shortest extension takes place between an entry relay in `NL` and a middle relay in `FR` with `UK` as forbidden area, then we use the average propagation speeds of `NL`→`UK` and `UK`→`FR` and apply it to the extension distance. Accordingly, we receive an empirical result for the extension time to F on the shortest possible trip for a circuit. We use the approximate extension time to now define the decision threshold:

$$R_{est} = ext_F + est(c, s \setminus ext_F) + est(e, s), \tag{4.10}$$

where ext_F is the shortest possible extension (cf. Equation 4.9), $est(c, s \setminus ext_F)$ are the estimates of all hops except those involved in the extension, and $est(s, e)$ summarizes all estimates for hops on the way back from the server to the client. In other words, we take a detour to the forbidden area on the trip from the client to the server and have two nodes of the circuit involved in this extension (cf. Figure 4.2). These two nodes represent the fastest possible option to reach the forbidden area, whereas all other nodes use direct connections. Consequently, we make lookups on the hop estimates for all pairs of nodes *not* involved in the extension to receive the transmission time from the client to the server. On the way back, we follow the asymmetry assumption and now use the estimates for lookups of *all* hops in the circuit. We receive the transmission time from server to client and can add up all components to the decision threshold R_{est}.

In the decision process, we relate the measured time R_{e2e} to our estimated time R_{est}, which we define as our *time ratio* Δ:

$$\Delta = \frac{R_{est}}{R_{e2e}}. \tag{4.11}$$

The reject/accept decision can now be performed directly against this time ratio. A time ratio of 1 marks the equality of our estimated threshold and the round-trip measurement. A lower Δ is calculated when the measured RTT

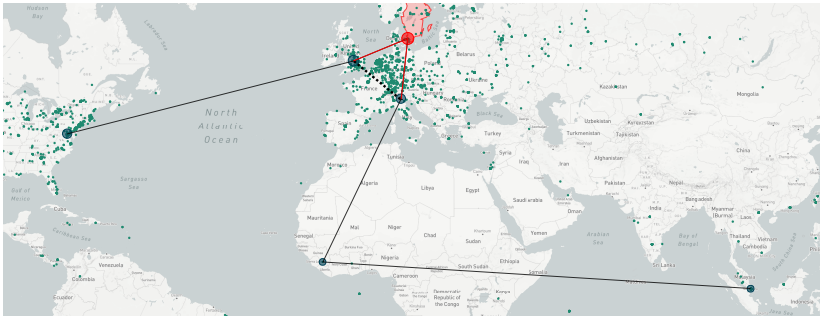


Figure 4.2. Exemplary extension of circuit with client in `AS` and server in `NA`. Relays in `AF` (entry) and `EU` (middle, exit). In this example, Sweden is the forbidden area with the shortest extension between the middle and exit. We use the extended circuit (red) on the way from client to server and the standard path (dashed) on the way back.

exceeds the estimation and indicates insecure circuits. On the other hand, a higher Δ results from measurements faster than the estimate. To account for a trade-off between security and performance, we can shift the decision threshold to either end. This allows to establish higher security guarantees or to keep more circuits for the sake of performance. Furthermore, we follow the lower bound threshold of $^2/_3c$ for use cases where provable avoidance is preferred over an empirical decision.

System Components

We build our empirical avoidance concept from the above decision mechanism and apply the design goals and solutions introduced with the assessment of challenges in Section 4.3. Our avoidance concept consists of two organizational units, i. e., on the network side we conduct distributed measurements as an information input for the avoidance decision. This includes the ICMP reference measurements for verifying relay locations through trilateration (Section 4.3.2), and the TCP measurements for the computation of pairwise hop estimates (cf. Equation 4.6). On the client side, we conduct the circuit measurements where repeated TCP probes through an established circuit reveal the RTT R_{e2e} of the connection. We compare this measured

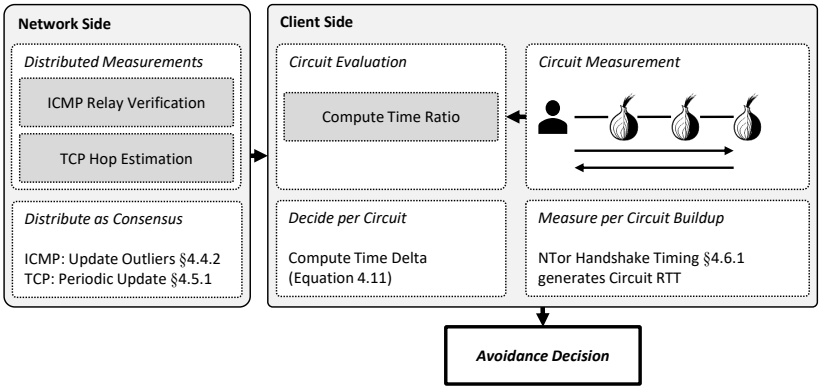


Figure 4.3. High level overview of the empirical avoidance concept. Network-side components provide offline information, client-side computes the time ratio from the measured time R_{e2e} and the threshold R_{est}.

RTT with the empirical threshold R_{est}, which leads to the time ratio between the measured and predicted transmission time (cf. Equation 4.11). The final avoidance decision uses this time ratio to rate the current circuit and, eventually, reject or accept its usage. All network side measurements are conducted offline; we discuss realistic ways to realize this following the example of Tor's consensus along with the proposal of the prototype (Section 4.5). Client-side measurements must be conducted before an established circuit transmits user data.

4.4.2. Experimental Evaluation

We compare our empirical avoidance concept with two other approaches that we distinguish by their consideration of the design challenges in the classes of *network diversity* and *ground truth* (a detailed evaluation of the requirements of a real-world *deployment* follows in § 4.5). We address the original version of DeTor that assumes symmetric paths and static transmission characteristics (cf. Equation 4.3); we refer to this as *symmetric* avoidance concept. Furthermore, we compare this to an updated version of DeTor, referred to as *asymmetric* concept, which functions in the exact same way but assumes

asymmetric routes to correct the logical flaw of DeTor (cf. Equation 4.4). Both static concepts use unverified GeoIP information. Finally, we introduce a novel *empirical* concept that uses hop-individual estimates and verified GeoIP locations. For the sake of comparability, we apply the decision mechanisms to *full* circuits from client to server that were measured by active TCP probing. We are interested in the detection capabilities of two static (symmetric, asymmetric) and one novel empirical avoidance concept. Our evaluation first focuses on the general performance concerning the number of rejected circuits and the avoided advertised bandwidth.

Measurement Setup

Our experiments are based upon empirical RTT measurements from the live Tor network, i. e., we use actual transmission characteristics for the computation of hop estimates and use the RTTs of full circuits to simulate avoidance decisions for all three concepts.

Test Set. We perform RTT measurements from eight server instances (`CA`, `NL`, `US`, `IN`, `SG`, `GB`, `DE`, `BR`) that send 20 TCP ping messages of 100 B length through an established Tor circuit. After each message, we wait 1 s until the next 100 B are sent to avoid any interaction. In case a reply was not received within the timeout limit of 2 min, we assume a failed connection. From 1670 entries, 2712 middles, and 735 exits, we build a total of 70,081 individual circuits and perform 275,509 measurements; the selection of relays is randomized and biased towards higher bandwidth nodes that provide 100 Mbit/s advertised bandwidth on average (132 Mbit/s in entries, 22 Mbit/s in middles, 148 Mbit/s in exits).

To ensure that our artificial circuits resemble similar transmission characteristics as weighted standard circuits, we build 135,924 additional weighted circuits using the `NEWNYM` command from the same remote server instances and compare their characteristics to those of the artificial circuits. The results of a Kolmogorov-Smirnov (KS) test of the probability distributions of both circuit lengths show that artificial `(NA,EU)`, `(NA,NA)` circuits tend to be shorter than the measured Tor standard circuits, while we find a higher similarity for the other combinations `(EU,EU)`, `(EU,NA)`, `(EU,AS)`, `(NA,AS)`.

Table 4.5. Loss [%] of Available Circuits and Bandwidth.

	Circuits			Bandwidth		
	Symm.	**Asymm.**	**Emp.**	**Symm.**	**Asymm.**	**Emp.**
DE	90	90	71	86	86	74
US	61	73	60	60	73	63
FR	94	94	76	92	92	76
UA	89	91	49	85	87	43
RU	88	86	56	84	85	51
NL	93	93	79	89	89	80
GB	94	94	73	92	92	65
SE	92	93	60	89	89	56
CA	66	79	–	66	78	–
Average	**85**	**88**	**65**	**82**	**86**	**64**

Simulation Methodology. We use the RTT measurements of all artificial circuits and compare the detection mechanisms of the three avoidance concepts. For each circuit, we iterate the top nine relay-providing countries (`DE`,`US`,`FR`,`UA`,`RU`,`NL`,`GB`,`SE`,`CA`) as hypothetical forbidden areas, using the following simulation methodology:

1. For all circuits, we identify the shortest possible extension to the current forbidden country F, compute the extension time, and identify its position in the circuit.

2. For the empirical approach, we perform a lookup on the estimated RTT for each hop in the current circuit and approximate the transmission time for the extension hop to the forbidden area F. Using this information, we compute the RTT threshold R_{est} (cf. Equation 4.10) and the time ratio Δ (cf. Equation 4.11) of the circuit.

3. For the symmetric and asymmetric decision, we follow the detection mechanism proposed in DeTor and compute the time consumption of each hop using the great circle distance between relays and a static speed of $^2/_3$c. We estimate the RTT threshold for a circuit following the definitions of Equation 4.2 for the symmetric approach and Equa-

tion 4.4 for the asymmetric approach. Again, we derive the time ratio Δ for the circuit.

4. We apply a decision threshold of $\Delta \geq 1$ to accept a circuit and handle all other time ratios as a reject decision.

4.4.3. Results

To evaluate our results, we analyze the relative number of circuits an avoidance concept rejects for a forbidden area F. Furthermore, we estimate the loss in advertised bandwidth that results from the avoidance decision.

Detection Capabilities

The reject and accept rates of a system indicate the restrictions in the choice of circuits when avoiding a specific geographical area. Table 4.5 (top) summarizes the reject rates, i. e., the relative number of circuits that were rejected because the measured RTT exceeded the respective threshold. When comparing the symmetric and asymmetric approaches, we see only minor differences for forbidden countries within Europe (`DE,FR,UA,NL,GB,SE`), but a significantly increased reject rate for the asymmetric approach for `US` and `CA`. This is caused by the higher extension distance to North America, i. e., remote forbidden areas emphasize the overestimation of DeTor's symmetric approach (cf. Equation 4.5). In comparison, with the hop-individual decision we reject overall approximately 23 % fewer circuits, as a result of the individual consideration of hop RTTs to be less conservative with the comparison threshold.

Performance Impairments

Being too conservative with the reject decision can cause severe performance impairments, especially in cases where large user groups decide to circumvent a certain area. The empirical approach manages to reject fewer circuits and can maintain on average 216 Mbit/s more (advertised) bandwidth per circuit. Table 4.5 (bottom) summarizes the relative bandwidth loss in a worst-case scenario, in which 100 % of users avoid a certain country. Example: As

we know from the usage statistics [The19d], approximately 45 % of Tor's advertised bandwidth is consumed on a daily basis. If we take the 13 % average daily users of the United States as an example [Met19] and assume `UA` as the forbidden region, this translates into an overall load factor of 51 % for the individual decision (56 % for the symmetric, 56 % for the asymmetric decision). Even though our results predict a worst-case scenario, it is likely that a majority of users is motivated to avoid the *same* country due to censorship activities. Losing bandwidth in the range of 85 % brings us close to an overloaded situation and is unacceptable.

Collateral Damage

Conservative reject decisions not only result in performance impairments for a user but can also cause collateral damage to the entire network. While highly sensitive use cases should maintain a restrictive lower bound threshold, less difficult cases allow for a trade off between detection capability and performance. We can adjust the security of the individual implementation by reducing the initial decision threshold of 1 for lower time ratios. This increases the chances of routing through the forbidden area but helps to reduce the reject rates drastically. We have a close look at the potential of using alternative decision threshold in the prototype evaluation (Section 4.5).

Summary. Utilizing an empirical decision allows incorporating individual timings. This reduces the error of a static lower bound threshold that can only represent best-case propagation speeds and neglects the varying transmission characteristics of real-world connections. Using an empirical threshold allows to trade off performance and security while reducing the collateral damage through overly restrictive decisions. Our results indicate that such collateral damage has an enormous impact on Tor's performance that affects all users and, therefore, cannot be an acceptable trade for security.

4.5. Prototype *TrilateraTor*

TrilateraTor is the prototype implementation of an empirical avoidance system that takes all three classes of challenges (Section 4.3) into account. In

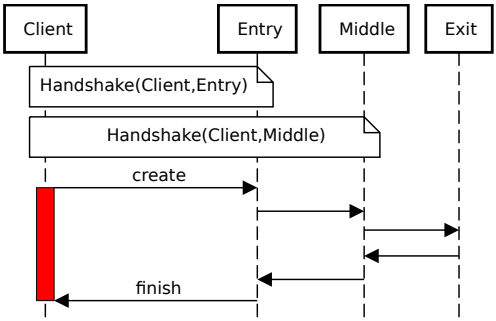


Figure 4.4. Summary of the `NTor` handshake protocol. When building a new circuit, the client performs three handshakes with the entry, middle, and exit relay of the circuit. We are interested in the offset between the `create` and `finish` messages (red bar) that summarizes the RTT for a message to be transmitted from client to exit and back.

particular, we extend the empirical avoidance concept (Section 4.4) by features that satisfy the conditions of a practical *deployment* scenario. In the following, we detail the system model, improve the security of the circuit RTT measurement technique, and provide an experimental analysis of the prototype's performance. Finally, we discuss the possible ways to realize the deployment of *TrilateraTor*.

4.5.1. System Model

TrilateraTor's geographical avoidance consists of the same organizational units as the previously introduced empirical concept (cf. Figure 4.3). The network-side measurements are conducted in case of changes in the consensus (update ICMP verification) or on a periodical basis (TCP hop estimates). We discuss options to share the information of the `Distributed Measurements` among Tor users in Section 4.6.1 and suggest an infrastructure with Tor's bandwidth authority. The first adaption to a real-world deployment takes place on the client side, where we apply a novel measurement technique to recover the end-to-end RTT R_{e2e} of a circuit. In contrast to the generic empirical concept, where we sent TCP ping probes to the entire connection,

we now limit our measurements to the Tor nodes in the circuit, i. e., the connection from the client to the exit relay. Again, we compare the measured RTT R_{e2e} to the predicted time R_{est} and derive the time ratio Δ. The time ratio suggests an avoidance decision following the desired trade off between performance and security in which we can shift the decision point towards higher (more security) or lower (more performance) thresholds.

4.5.2. Avoidance Decision

In the evaluation of avoidance concepts we were able to use hop estimates for full circuits, i. e., our RTT measurements provided us with estimates that also cover hops between the client and entry/exit and server. In a realistic setup, such hops are highly individual and cannot be covered. Furthermore, we switch from actively sending TCP ping probes through an established circuit (introduced by DeTor) to measure the time offset in the key establishment for organizational and security benefits, as we will present throughout this section. These changes lead us to an updated decision threshold R_{est} that consists of the shortest possible extension to the forbidden area ext_F, an approximation of the transmission time between client and entry $app_{c,e}$, and the pairwise hop estimates for remaining hops in the circuit. We first define the $c \rightarrow e$ approximation:

$$app_{c,e} = \frac{D(c,e)}{avg(S(c,e))}, \tag{4.12}$$

where $D(c,e)$ is the great circle distance from the client to the entry, and $avg(S(c,e))$ is the average measured propagation speed from the client's country to the entry's country. Using this approximation of the first hop, we now define two cases for the definition of the predicted transmission time R_{est}:

$$R_{est} = \begin{cases} 2 \cdot app_{c,e} + ext_F + est(e, x, e \setminus ext_F) & , c \notin ext_F \\ ext_F + est(c, x, c) + app_{c,e} & , c \in ext_F \end{cases},$$

where we distinguish an extension to F that happens without including the client ($c \notin ext_F$), or an extension that takes place between the client and the

entry node ($c \in ext_F$). In the first case, we approximate the hop from the client to the entry twice for both directions of the connection and estimate the remaining hops and the extension as usual. In the second case, the shortest possible extension is between the client and the entry, and we only need to approximate this hop for the way back from the exit to the client. Just as in the empirical concept, we estimate only hops that are not involved in the extension ($est(e, x, e \setminus ext_F)$); the estimation $est(c, x, c)$ includes all hops from client c to exit x and back to c. For the final avoidance decision, we compare the predicted time R_{est} with the measured RTT between the client and the exit R_{e2e} and reject or accept a circuit according to the time ratio and desired decision threshold. We derive the circuit timing from a novel measurement technique that we introduce as follows.

4.5.3. Circuit Establishment Timing

On each new circuit establishment, Tor performs three cryptographic handshakes with the entry, middle, and exit relay of a connection (cf. Figure 4.4). Each of these handshakes traverses parts of the circuit and delivers the end-to-end timing information R_{e2e}. We measure the handshake timings in the `NTor` [Tor19b] implementation, which provides Tor's cryptographic primitives since version `0.2.4.x`. More precisely, we measure the offsets (red bar) between the `create` and the final `finish` messages. The client repeats the handshake procedure for each relay in the circuit and finally delivers the total transmission time between the client and the exit. In contrast to active TCP probing, the Tor client performs the required cryptographic handshakes at each circuit establishment, i. e., we can derive all relevant information without any active interference. We will see later how this benefits the usability of the avoidance system and overcomes one existing security issue of DeTor.

4.5.4. Experiments

In our prototype evaluation, we first address the timing characteristics of RTTs derived from the circuit establishment procedure and compare them to the characteristics we observed for TCP pings (Section 4.3). Furthermore,

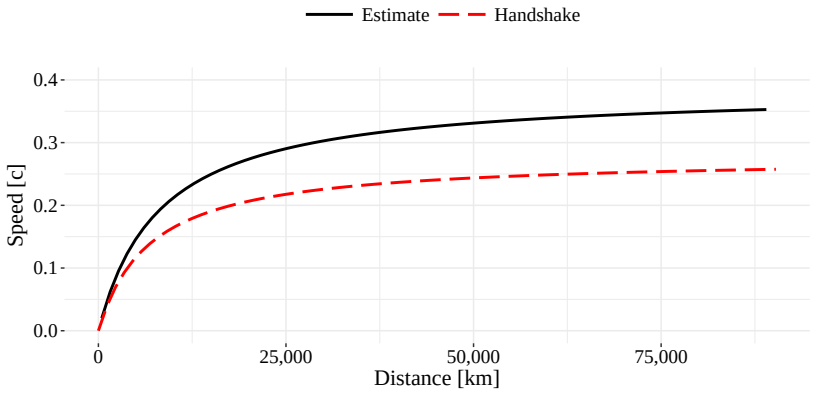


Figure 4.5. Comparison of propagation speeds in estimates (black) and `NTor` handshakes (red). Results summarize the spectrum of measured times for full circuits (areas) and the NLS propagation speed fit (lines).

we analyze the performance of *TrilateraTor* in comparison to the theoretical avoidance concepts of Section 4.4.

Experimental Setup

We use eight worldwide server instances (`CA,NL,US,US,IN,SG,GB,DE`) and measure a total of 16,500 individual relay combinations (1945 entries, 3724 middles, 893 exits) for the exit handshake offsets. For each measurement, we draw 100 top bandwidth relays from the first consensus of the day and form random circuits from this; measurements are repeated every 10 min within three days. We document the handshake timings for all successful circuit establishments along with the relay at which the buildup procedure failed in case of an unfinished circuit establishment. Besides the handshake measurements, we repeat the ICMP reference measurements (Section 4.3.2) and TCP ping measurements (Section 4.4.1) to provide recent information for the verification of relay positions and the estimation of hop relations.

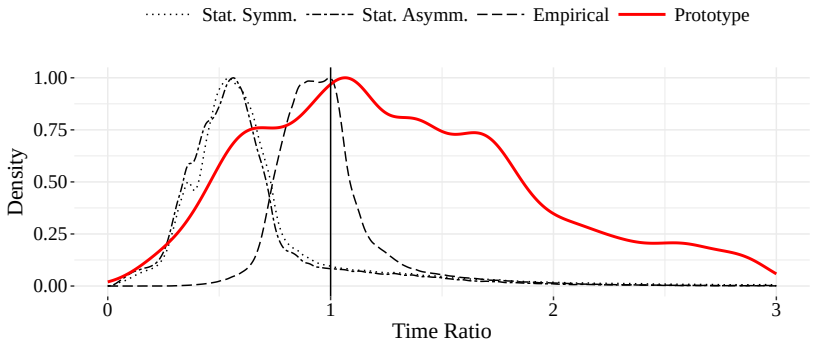


Figure 4.6. Comparison of time ratios of different avoidance concepts. We find higher time ratios for the prototype of *TrilateraTor*, indicating higher acceptance rates for tested circuits. Theoretical concepts (Stat. Symm., Stat. Asymm., Empirical) fall below this rate by 0.598 on average.

Timing Characteristics

We analyze the robustness of timings from the key agreement procedure, as in comparison to repeated TCP Ping measurements, a much smaller data basis for the decision is given. To do so, we measure the median deviation of handshake times between identical hops. Our results indicate that exit handshakes differ by 7 % between measurements, which results in an average variance of 32 ms.

Furthermore, we compare the propagation speed of handshake timings with the TCP ping hop estimates (cf. Figure 4.5). The cryptographic computations of the handshake procedure induce an additional overhead that leads to an overall reduced propagation speed in comparison to the hop estimates. As this computational overhead is not related to the transmission characteristics of a connection, we speed up the handshake measurements by the difference of $0.031c$ between the estimates and the handshakes.

Performance

We now compare the performance of *TrilateraTor* with the theoretical concepts of Section 4.4 and analyze the spectrum of time ratios that results from avoiding the top nine relay providing countries (cf. Figure 4.6). We see that the static avoidance concepts lead to smaller time ratios, which supports the finding that a fixed speed assumption leads to overly restrictive decisions. In comparison, the empirical approach achieves a median time ratio of 0.946 and with that is close to a balanced distribution of decisions.

Our prototype implementation improves this result further, e. g., we achieve a median time ratio of 1.297 that allows accepting a high number of circuits. At this point it is important to once more emphasize the structural differences between the theoretical concepts and *TrilateraTor*: While we can analyze *full* circuits in the theoretical concepts, the prototype is limited to connections from the client to the exit. Nevertheless, the comparison of time ratios still delivers an essential perspective on the differences between the theoretical concepts and practical implementation.

4.6. Discussion

For successfully deploying *TrilateraTor*, we depend on reliable sources for all information that we consider in the avoidance decision. Furthermore, we must maintain Tor's original level of security and limit the performance impairments that the avoidance feature induces. In the following, we discuss the organizational aspects of deployment, analyze the security features of *TrilateraTor*, and estimate potential performance impairments.

4.6.1. Information Sources

We depend on three different sources of information for an avoidance decision in *TrilateraTor*: Distributed ICMP and TCP measurements improve the trust in GeoIP information and deliver the empirical timing information for individual hops between relays. Handshake measurements allow us to derive a circuit's RTT without any active probing.

ICMP and TCP Measurements

The strength of the ICMP and TCP measurements lies in the fact that we use multiple worldwide server instances that either reliably identify false relay locations in the case of ICMP, or generate representative empirical estimates of the timing characteristics between hops (TCP). While we will see later (Section 4.6.3) how this adds another layer of security, we are now interested in ways to organize these distributed measurements in case of deployment.

We assume fixed relay locations, i. e., ICMP reference measurements only require updates for changes in the consensus. Throughout 2017, a fluctuation of approximately 17 % occurred when existing relays disappeared, or new relays appeared in the consensus. For an average number of 7283 relays in the consensus, this translates to approximately 1238 nodes that require updates (in the worst-case, only new relays need to be verified). The situation is different for TCP measurements, e. g., we do not only need to cover fluctuations in the consensus but must also consider varying transmission characteristics (Section 4.3.2). Therefore, periodic updates help to improve the data basis for the pairwise hop estimations. For both the ICMP and TCP information, we advocate a consensus-centric infrastructure that allows users to access all relevant information.

The overhead through distributed measurements is negligible in comparison to Tor's daily usage and the provided capacities. We can assume approximately 2.8 Mio. daily Tor users, and an average consumed bandwidth of 121.5 Gbit/s. Under the assumption that an average user builds at least three circuits (this is a minimum estimate, numbers should be much higher), all experimental circuits represent approximately 4×10^{-4} % of Tor's daily circuits. To send 500 messages (as an upper bound for the number of probes sent) with a length of 100 B, we require 758 kbit/s per day, which is only 6.24×10^{-7} % of the daily bandwidth consumption in Tor.

4.6.2. Reproducibility

Our experimental setups can only represent snapshots of Tor's network infrastructure and describe the period in which empirical data was gathered. Differences might arise from varying network conditions (congestion, outages,

Table 4.6. Variance in Timings Measurements [ms].

Type	Iteration	Mean	Median	SD	Duration	#Results
TCP	1	287	288	158	5 days	223,070
	2	359	335	180	7 days	134,370
	3	327	295	185	8 days	275,509
ICMP	1	99	67	98	1 day	27,274
	2	56	18	77	1 day	62,643
	3	136	128	102	2 days	1,837,761

attacks), the selection of measurement points (server locations), the possible forbidden areas, etc. Nevertheless, our selection of experimental components represents worldwide server positions, top bandwidth relays provide the majority of Tor's performance capacities, and the number of conducted measurements delivers a profound data basis. Table 4.6 summarizes the characteristics of our repeated measurements. In both sets, the first two iterations were conducted within one month, whereas the third iteration serves as a reference from six months later.

Results show a high standard deviation (**SD**) within all measurement sets, but are in a comparable range between iterations (28 ms delta between TCP results; 26 ms for ICMP). The variance of results once more confirms the findings of our challenges assessment, i. e., transmission characteristics depend on the infrastructure and health of the network and change, accordingly. Nevertheless, results are sufficiently comparable even through longer measurement periods.

4.6.3. Security

Any behavior that leaks information can open new attack vectors and, consequently, harm users that depend on additional protection mechanisms. Furthermore, overly restrictive decisions reduce one of Tor's core security features, the anonymity set size, and even facilitate traffic-analysis attacks. In the following, we discuss security implications that could arise from deploying an avoidance system.

Fingerprinting

An adversary that actively monitors the circuit establishment procedure might recognize deviations from expected patterns and derive fingerprinting information from this. The ability to fingerprint actions of the avoidance system can reveal the endpoints of a connection, help to derive sensitive information from unexpected user behavior or to reduce the anonymity set.

Revealing Connection Endpoints. Measuring the timing characteristics of a circuit through TCP pings requires sending messages along the full transmission path between the client and the server. Under the assumption of a strong AS- or state-level adversary, such messages can reveal the endpoints of a connection, as the RTT measurements also include the destination of the connection. *TrilateraTor does not leak such information, as it utilizes the crypto handshake of the circuit buildup procedure.*

Unexpected Behavior. Active TCP ping measurements add unexpected traffic to the standard transmission patterns of a user. An adversary can monitor batches of TCP probes sent out by the avoidance system and derive additional information from this. Such information includes the presence of an avoidance system and might help to predict the choice of relays. *TrilateraTor attaches to already existing functions of Tor and does not depend on active probing, i. e., it maintains the original circuit buildup behavior.*

Reducing the Anonymity Set. Rejecting a majority of circuits helps an adversary to predict the remaining set of relays that are suitable candidates to circumvent a forbidden area. As a consequence, traffic-analysis attacks become more likely, and the measurement overhead is reduced—both factors would otherwise only enable potent adversaries to succeed. *TrilateraTor manages to reduce the number of rejected circuits and, furthermore, allows to apply a context-sensitive trade off between security and performance.*

Measurement Manipulation

A powerful, nation-state adversary can manipulate [ZNR07] the distributed measurements (ICMP, TCP ping) of an avoidance system by holding back probes. This results in an overall increased transmission time that would manipulate the relay verification and computation of hop estimates. *Trilat-*

eraTor inherently limits the impact of such attacks. All network side measurements are conducted from multiple reference points, i. e., the scenario is comparable to verifiable trilateration as proposed by Čapkun et al. [vH05]. Prolonging one distance to a reference would require the shortening of at least one other distance to a different reference. However, this would need accelerating packets beyond typical Internet transmission speeds mitigating the manipulation success while leaving conspicuous attack fingerprints. *TrilateraTor protects against measurement manipulations that otherwise would affect a timing-based avoidance decision.*

4.6.4. Performance

Two influencing factors have the potential to impair Tor's original level of performance. First, timing-based avoidance systems depend on RTT measurements for a tested circuit. Prior work introduced active TCP probing where a client sends messages through the established circuit and measures the offset until the response was received. This approach forces users to wait until the measurement procedure is finished and hinders from directly using a (safe) circuit. We overcome this by using the circuit establishment handshake as an information source, i. e., we induce no additional waiting time. Second, restrictive avoidance decisions limit Tor's available resources. Our worst-case evaluation (Section 4.4 and Table 4.5) proves that *TrilateraTor* manages to reduce this source of collateral damage by preserving approximately 22 % more of Tor's advertised bandwidth. This is an important result, as a congested infrastructure also affects users that do not make use of *TrilateraTor*. From an individual perspective, users must always accept slight performance impairments through geographical avoidance, as rejecting the most prominent relay choices often leads to weaker circuits.

4.7. Conclusion

In this work, we assessed the challenges of geographical avoidance for data transmissions and used it as a foundation to introduce a novel empirical avoidance concept. To this end, our concept considers hop-individual trans-

mission characteristics instead of static thresholds for individual connections, limiting the collateral damage through overly restrictive avoidance decisions. In a two-fold experimental study, we first compared the performance of our empirical avoidance concepts to existing work and managed to outperform other approaches by rejecting 22 % fewer circuits and maintaining on average 216 Mbit/s more advertised bandwidth. In a second step, we introduced the prototype implementation *TrilateraTor* that considers the requirements of a real-world deployment in addition to the challenges of Tor's diverse network infrastructure and untrusted ground truth information. *TrilateraTor* is the first to provide *practical* geographical avoidance and overcomes fundamental security issues of prior systems.

Der Kampf gegen Gipfel vermag ein Menschenherz auszufüllen. Wir müssen uns Sisyphos als einen glücklichen Menschen vorstellen.

— Albert Camus

5
LTE Website Fingerprinting

Contents

5.1. 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 ranging from denial-of-service through jamming [LRCN13, LJL+16, ASS15, Jov13], over downgrade attacks that enforce a more insecure communication standard [SBA+16, Jov16b, MO17], to identification and localization attacks that reveal the presence of a user within a radio cell [SBA+16]. 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). Rupprecht et al. [RKHP19] 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 [WCJ07, WCJ05] or passively [LRWW04], is often in a position to recover sensitive information about a user. Privacy leaks by traffic fingerprinting attacks first emerged when Cheng et al. [CA98] 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 [HWF09, Hin02], models of the user behavior [PLZ+18], and modern machine-learning algorithms [Wan15, RPJ+18] helped to improve the success of fingerprinting attacks in more challenging scenarios. Systems with additional security features, e. g., the Tor anonymity network [The19c], limit the threat of traffic analysis. Nevertheless, there is a large body of powerful attacks that also suc-

ceed in the context of Tor [JAA+14, JJG+17, MZ07, MD05, DDT07]. 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 [JAA+14]. 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 meta data 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, we make the following three contributions to answer the third research question.

Research Question 3 *Can we transfer well-known attack techniques to mobile networks to emphasize the threat of traffic analysis?*

- **Performance Baseline.** Using meta-analysis as a starting point, we conduct a *website fingerprinting* attack in a controlled lab environ-

ment 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.

The above contributions result from a collaboration with David Rupprecht, Thorsten Holz, and Christina Pöpper. In particular, David Rupprecht conducted the lab and commercial LTE measurements.

5.2. Preliminaries

BBefore 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.

5.2.1. 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 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 Radio Network Temporary Identifier (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 signaled 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 meta data input for traffic fingerprinting.

5.2.2. 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 [RPJ+18] where the adversary gathers labeled traffic traces of candidate websites for a training set to later test an unlabeled 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. [CA98] 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 [SSW+02, LL06, HWF09]. 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 [WCN+14] or Support Vector Machines (SVM) [PLZ+18] with success rates around 90 % on obfuscated Tor traffic. Current attacks extend this by automated feature engineering [Wan15] and deep learning [RPJ+18] with high success rates in large data sets.

Fingerprinting Attacks. We define two separate fingerprinting attacks that exploit the meta data 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).

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 IMSI or 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 [srs18] implemented on a Software Defined Radio (SDR) in combination with an analysis framework (sniffer) like Airscope [Sof19], imdea OWL [BW16b, Bui17], or other commercial systems [San12]. 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, meta data 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.

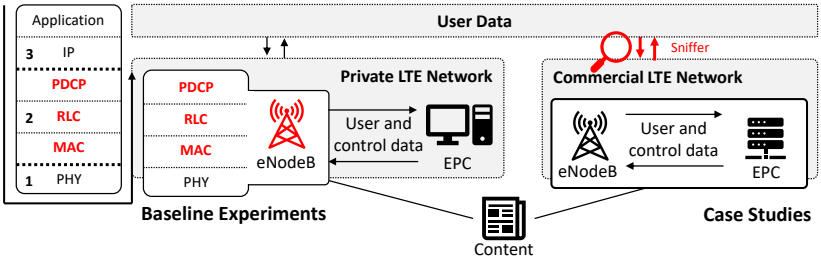


Figure 5.1. Comparison of experimental setups. The lab setup serves the baseline experiments of Section 5.4; the commercial LTE network serves the case studies of Section 5.5. In both cases, user data traverses the LTE protocol stack and we can access and decode data up to the PDCP layer. In the private network, we access this data in the eNodeB; in the case studies we use a sniffing tool.

5.3. 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. In the following, we introduce the components of this private network and discuss how this controlled environment differs from the commercial network we use for the empirical case studies of Section 5.5.

5.3.1. 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 5.1). 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 [Sof19]. We next define the technical characteristics of our network components and introduce the meta data features that we derive from the traces.

We derive transmission meta data from the sub-layers of LTE layer two traffic, which limits the available information for the classification problem of the fingerprinting attacks. In the following, we define the technical characteristics of our network components and introduce the meta data features that we can derive from traces, respectively.

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 5.1.

- **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.

Table 5.1. Specification of Experimental Devices.

Device	Resolution	Chipset	OS	Browser	Network	Release
Moto G4	1080x1920	Qualcomm MSM8952 Snapdragon 617	6.0.1	Chrome	GSM/CDMA HPS/LTE	05/2016
P9 Lite	1080x1920	HiSilicon Kirin 650	7.0	Chrome	GSM/CDMA HSPA/LTE	10/2013
Nexus 5	1080x1920	Qualcomm MSM8974 Snapdragon 800	5.1	Chrome	GSM/CDMA HSPA/LTE	10/2013
iPhone 6s	750x1334	Apple A9	12.0	Safari	GSM/CDMA/HSPA EVDO/LTE	09/2015

- **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 meta data. We use the RNTI to distinguish different traces and refer to DCI information to understand the resource allocation. Furthermore, we derive meta data 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 these *raw* transmission information, we generate an aggregated representation in which we summarize packets in time-based windows (cf. Figure 5.2). We apply a window of 500 ms length and aggregate the packet occurrences in each window. Following this approach, we compress of the original raw trace to 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.

5.3.2. Recording Procedure

For the experiments of Section 5.4 and the website fingerprinting case study of Section 5.5, we follow a general recording procedure to gather the data sets of different scenarios. In the following, we introduce this recording procedure and highlight aspects that differ for the lab and the commercial network.

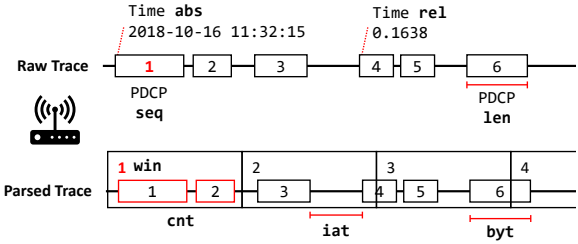


Figure 5.2. Aggregation of raw traces into condensed representation. We separate time data into windows of same length and gather occurrences of packets within these windows.

Table 5.2. Overview of Website Candidates in Experiments.

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

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 5.2). 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 (cf. Figure 5.2).

5.3.3. Parameters

We compare the success of different attack setups (①–②, cf. Table 5.3) 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 5.1). 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?*

We use the above parameters to create a performance baseline of website fingerprinting in a controlled network environment. This performance assessment is a reference for the real-world case studies of Section 5.5 and serves as a starting point for future work on LTE layer two fingerprinting attacks.

5.3.4. Attack

Website fingerprinting on LTE is, as previously introduced in context of Tor, a classification problem for an open or closed world setup in which the adversary gathers labeled traces for a training set and later conducts the attack on the unlabeled traffic of a user.

Classification Problem. We focus on three machine-learning techniques that we find in recent WF approaches (k-Nearest-Neighbor (k-NN) [WCN+14], Support Vector Machines (SVM) [PLZ+18], Neural Network (NN)). 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:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.1}$$

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.

5.4. 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. The results of our private network experiments serve as a performance baseline and estimate the attack success in a controlled environment,

Table 5.3. Overview of Experimental Setups and Results.

Experiment		Setup				F_1	
ID	**Description**	**Devices**	**Parameter**	**Classifier**	**Cached**	**Uplink**	**Downlink**
①	Classifier	Nexus 5	-	**k-NN**	Uncached	**0.949**	**0.945**
				SVM		0.928	0.928
		P9 lite		NN		0.922	0.919
				k-NN	Cached	**0.860**	**0.815**
		Moto G4		SVM		0.822	0.776
				NN		0.842	0.806
②	Recording Effort	Nexus 5	**5-20 Traces**	k-NN	Uncached	**0.849**	**0.844**
		P9 lite, Moto G4	20-60 Traces			0.921	0.933
③	Hardware/Software	iPhone 6s	iOS	k-NN	Cached	0.686	0.706
		Moto G4	Android			0.751	0.726
		P9 lite A, P9 lite B	Same device	k-NN	Cached	0.835	0.805
④	Time	P9 lite	7 month Comb.	k-NN	Uncached	0.871	0.827
			7 month Offset			0.037	0.026
			12 day Comb.			0.819	0.790
			12 day 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

comparable to a best-case scenario. In the following, we provide an overview of all baseline experiments and the experimental results, respectively.

5.4.1. Experiments

Table 5.3 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; for a global performance comparison, we mix the data sets of three different smartphones. After identifying the best performing classifier, 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 through hardware and software ③, the impact of time ④ and 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 (cf. Table 5.2) for a closed-world attack.

Experiment ①: Classifier Comparison

In our first experiment, we compare the performance of three classifiers (k-Nearest Neighbor (k-NN), Support Vector Machines (SVM), Neural Network (NN)) 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. We form random training and test sets from the entire set of 9000 traces (consisting of 60 traces for 50 websites with three devices; 9000 traces cached and 9000 traces uncached).

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.*

Experiment ②: Recording Effort

In the preceding experiment, we used a total of 60 traces per website, which leads to a robust classification success but also introduces a high measurement overhead. If we can achieve a sufficient success rate with fewer traces, we can limit the measurement overhead and, consequently, also reduce the restrictions of the attack scenario. In other words, the adversary can limit the recording overhead and requires less time to conduct a successful attack. 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. Even though we achieve a higher attack success when using more traces, the improvement does not justify the measurement overhead. *We continue the following experiments with data sets of 20 traces per website.*

Experiment 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 varying 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 for the reference Android traffic $F_1 = 0.738$. 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 can help to distinguish multiple devices, we do not find any significant difference between both data sets recorded in parallel.

Experiment ④: Time

Due to changing website contents, we consider time as an important influencing factor for the success of an attack. In particular, such changes impact traffic characteristics and can hinder an attack that uses data sets recorded over a longer period or traces that are not sufficiently *fresh*. 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 progressing impact of time. Our long-term data set consists of recordings from March and September 2018, whereas we record the second data set in 12 consecutive days of September 2018.

In both cases we test scenarios in which the adversary conducts the attack with an *offset*, 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 summarize a scenario in which the adversary monitors the training set on one day and performs the attack on recordings from a later point in time. This offset decreases the attack success, as we are limited to an average of $F_1 = 0.711$ for twelve days and see a failing attack with an average of $F_1 = 0.032$ for seven months. This effect can be explained with the content

changes that have a more significant impact for scenarios in which the training set can only consist of one specific recording session.

Experiment ⑤: Obfuscation

As obfuscation systems often function on the application layer, we cannot be sure whether these effects reach down to the traffic of layer two. To measure such obfuscation effects, we setup `Orbot` [Gua19] as Tor proxy that sends and receives all traffic through a Tor circuit.

Results. Our results show that, when using this additional layer of obfuscation, we experience a lower classification success of $F_1 = 0.532$. This drop in performance can be explained with the transmission characteristics of Tor traffic. First, the 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 (cf. Figure 5.3). 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.

Experiment 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.

In our experiments, we test whether the attack is possible with a data set consisting of WiFi traffic and assess the attack success with training data from WiFi traces and test data from LTE traces. We run two simultaneous experiments in which one device connects to the standard LTE network, and

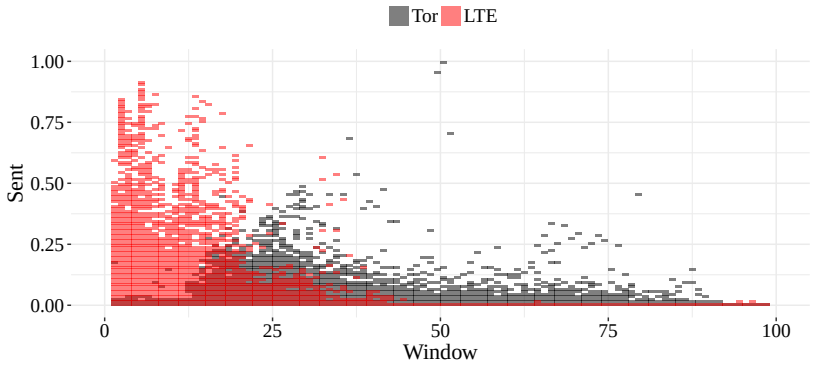


Figure 5.3. Comparison of Tor and LTE transmission characteristics. Sent data versus transmission duration in comparison for Tor and LTE traffic. We find a wider distribution of Tor transmissions that indicate volatile transmission characteristics for Tor.

the other device fetches websites via a WiFi router. As the recordings run at the same time, we intend to limit effects through varying website contents or network conditions.

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 meta data 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.

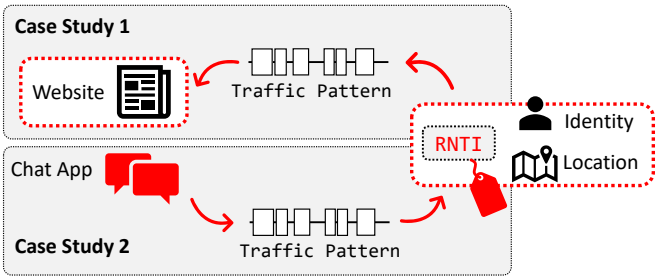


Figure 5.4. Commercial network case studies. The results of a targeted user identification/localization (CS2) can amplify the impact of the otherwise untargeted website fingerprinting (CS1) by delivering the identity of a specific user.

5.4.2. Summary

We achieve high success rates for a closed-world setup with 50 websites and can assume that LTE layer two traffic provides sufficient meta data 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.

5.5. Real-World Experiments

Figure 5.4 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.

5.5.1. 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 [Sof19] 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 [KKHK12, BW16b].

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 5.2.2) 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 5.4) 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.

5.5.2. 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. As explained above, we record traffic using a downlink analysis tool [Sof19], that provides us with the DCI scheduling information, allows us to distinguish multiple transmissions through the RNTI, and derive traffic meta data from the decoded PDCP sub-layer. To recognize the traces of our specific experimental UE, we use the Qualcomm debug interface and derive the RNTI using a TMSI lookup through SCAT [fgs18].

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. We follow the recording procedure of our baseline experiments (cf. Section 5.4) and record traces for the same candidate set of 50 websites using one smartphone (LG Nexus 5). To entirely focus on the evaluation of live network effects, we use an uncached browser setup and train the k-NN classifier on 20 downlink traces.

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 9.845 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) [KKHK12]. 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. Consequently, 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 comparably 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 LTE layer two fingerprinting and creates a substantial impact.

5.5.3. CS2: Traffic Watermarking

In the second case study, we demonstrate the feasibility of an 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 [RKHP19]. 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 at the same time, 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. As our attacker model (cf. Section 5.2.2) assumes a single adversary that can cover one radio cell only, the presence of the targeted user within the monitored cell must be predicted. We discuss the impact of this attack in the presence of a large-scale adversary in Section 5.6.1. In the following, we further specify the attack procedure and provide a proof-of-work.

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 (cf. Figure 5.5), 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 5.6.2). 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 [Sof19] 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 messages consisting of 100 characters ("AAAA...") using the WhatsApp web interface on a separate computer. We create a distinct pattern by timing

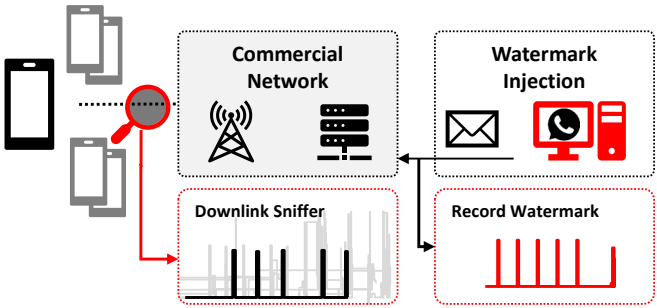


Figure 5.5. Concept of active fingerprinting. Adversary injects watermark using WhatsApp web interface; watermark messages get transmitted through LTE network and adversary monitors outgoing pattern (red pattern). User receives messages; downlink sniffing shows all traffic (gray lines) including injected watermark (black line).

each message with a 0.5 s delay between each transmission and send a total of five messages per watermark. As the raw 100 B messages are not the final message length but are a target to the encryption and encapsulation overhead of the WhatsApp communication protocol, we simultaneously monitor the outgoing traffic of the WhatsApp computer to receive the *final* byte pattern of our 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. Figure 5.6 shows an exemplary scenario in which we recorded all active transmissions (gray background lines) and the injected pattern (highlighted in red) of the commercial LTE cell. 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

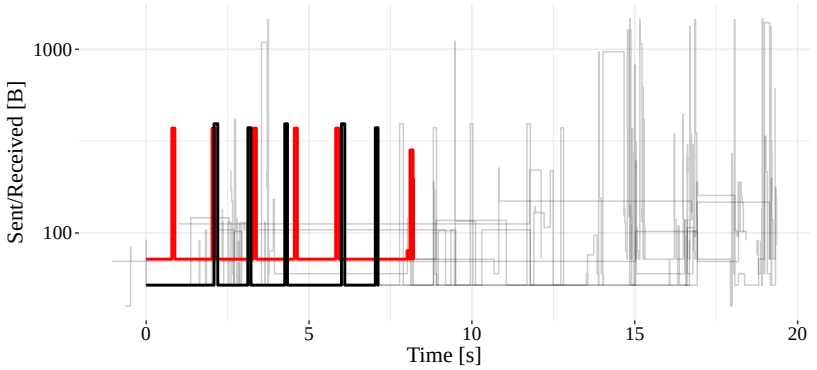


Figure 5.6. Example of monitored traffic for CS2. Downlink traces for all active users in the cell; red line indicates injected watermark, black line indicates recovered watermark. 17 active users present during this specific recording.

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 13.2 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 [TIM+13]. In comparison to paging attacks, which cover an entire tracking area, the injection of layer two watermarks allows for more precise localization.

5.6. Discussion

Our experiments provide a performance baseline for state of the art fingerprinting attacks and prove their feasibility in a real-world commercial network. 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.

5.6.1. Large-Scale Adversaries

The attacker model of Section 5.2.2 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 5.5.

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 [Cro19]. 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 meta data 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, i. e., he can extend the world size [JAA+14] to increase the overall success of the attack. Consequently, data-hungry classification approaches, e. g., deep learning [RPJ+18], 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*.

5.6.2. 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 [SBA+16, KKHK12]. A frequent and randomized reallocation of all temporary identifiers [SBA+16, HBK18], e. g., the TMSI, can help to mitigate the threat of paging attacks. As we exploit meta data 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 Identifier (GUTI) reallocation, the RNTI reallocation policy is *not* part of the specification [3GP09a]. Jover [Jov16b, Jov16a] 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) [3GP09b].

Countermeasures against website fingerprinting aim to obfuscate traffic characteristics that otherwise reveal the similarities between monitored website traces [WG17, WCN+14, WCM09]. While we see that the general application layer obfuscation of Tor has a significant effect on the attack (cf. Section 5.4), 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 [KKHK12] 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 [SBA+16], but also WebSockets or embedded JavaScript offer the required functionality.

5.6.3. 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 [HEC+19]. 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.*

5.6.4. 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 [JAA+14] 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 [RPJ+18]. 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. [RPJ+18] 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). Parallelization can reduce the required time to generate a sufficient amount of recordings, nevertheless, this requires multiple smartphones and data traffic in case of a real-world experiment. While we must assume that deep learning, in general, would be possible with LTE traces, we cannot satisfy the data set requirements to achieve a reasonable attack performance.

5.7. 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.

Part II.

Protection of User Data

No matter what other people or even critics will say, you have to follow your own direction which not necessarily has to be a straight line to success; sometimes it will be a curly, dramatic curve you have to go, but that's the only way to leave a little landmark of brave respect to others and to the dimensions of your own capability.

— Edgar Froese

6

Revocation of Online Data

Contents

6.1. Motivation

Anonymity systems depend on strict security requirements for protecting sensitive user data *and* user identities. As we learned in the first part of this thesis, large-scale adversaries and the inevitable performance requirements of the real world challenge the design of secure and fast anonymity systems. By switching to the context of protecting *user data* instead of *identities*, we lose some of the security requirements involving anonymity and gain more degrees of freedom for the establishment of *digital forgetting*.

As an integral part of our daily lives, we use online services as source of information, communication platform, cloud storage, and numerous more applications. By accessing such services, we also share a certain amount of personal information with the service providers and leave long traces of online footprints. Keeping track of everything we share becomes increasingly difficult [KHKU18] as time goes by and the amount of data continues to grow. The 2016 European GDPR contains the right to erasure [GDP19] and equips users with the right to delete their online data. Although this right changes the power relations between users and large service providers like Google or Facebook, we identify two apparent limitations. First, users need to make *proactive* decisions when they want to establish their rights. Second, the right to erasure only retroactively affects information that has already been shared with a service provider. *Revocation systems* change the way we experience the enforceability of our rights. They offer the technical means for assigning lifetimes to online data upon its upload and allow for proactive handling of information.

Prior work on revocation systems agrees on the general concept of controlling uploaded data through an encryption key [GKLL09, ZSXF10, CDCFK11]. More precisely, the system encrypts a file before its upload and shares the key material with the desired group of users. After a specific lifetime, the key expires and the uploaded object cannot be recovered anymore. Although we still share sensitive information with large service providers, it is now only an inaccessible ciphertext. Different technical and usability aspects challenge the design of a realistic revocation system. First, sharing the key material requires an infrastructure for storing the key. In contrast

to a centralized key server, an external and distributed infrastructure is not involved in the sharing-and-deleting process and has no interest in interfering with the security of the revocation system. However, we must expect a large number of encrypted files and users, i. e., we depend on scalable solutions. Second, revocation systems are meant to support users in handling their online data [CSMK15]. This means limiting the user's burden, as otherwise, we achieve only a minor increment in comparison to the proactive options we already have due to the right to erasure. A critical example of this is the inconvenience of planning the time we want files to remain available.

In this chapter, we document the technical background of exemplary key distribution infrastructure, discuss essential aspects of user perspectives, and overview related work in this context.

6.2. Background

Following the general concept of a distributed revocation system, we depend on an external infrastructure for handling the key material of encrypted data. In the following, we introduce the technical background of the Domain Name System (DNS) infrastructure as one example of independent key storage. Furthermore, we overview the results of prior empirical work addressing the perspectives of users.

6.2.1. Domain Name System

The DNS is a decentralized system for the localization of resources on the Internet. DNS resolvers associate readable domain names with the numerical IP addresses of a target and help to forward requests to the desired destination. In its decentralized infrastructure, different types of DNS resolvers exchange and delegate requests to provide comprehensive coverage of addresses and locations. Different types of servers in the infrastructure help to organize requests, e. g., authoritative resolvers cover their own designated spaces and can be requested for specific lookups.

The DNS infrastructure helps to create ephemeral storage for the key material of encrypted data. More precisely, we can use entries in the DNS

resolvers to encode the bits of encryption keys and make them publicly available to users. In the following, we provide a brief introduction of client-side DNS resolvers and explain how the caching mechanism helps to encode keys.

Resolvers

DNS resolvers are the client side of the DNS infrastructure and can be requested for the IP address information of a domain name. To resolve a request, they sequence queries to other resolvers until the resolution of a domain name is possible. Three types of requests manage the forwarding of requests to other resolvers: Recursive queries begin at a single server that forwards the request through multiple servers until a full resolution is possible, i. e., they include queries to additional resolvers. In contrast, iterative requests begin at one resolver that replies with either the full resolution or the destination of another resolver that might have the answer to the initial query. More precisely, the name server does not fetch the full resolution but only suggests other servers with additional information. Non-recursive queries can result in partial information, as the request does not proceed to further servers.

The type of query defines the caching behavior of a resolver, i. e., only *recursive* requests trigger a resolver to cache the resulting information. We benefit from this behavior when encoding the bits of an encryption key in multiple DNS resolvers.

Caching

Caching entries helps to reduce the overhead of repeated recursive requests to a series of DNS resolvers. In case of a recursive query, the resolver investigates the requested information and saves the resolved IP address in a cache entry in addition to replying to the client. Such cache entries have a predefined Time to Live (TTL) that defines the period in which the cache entry remains available. The resolver can answer subsequent requests for the same domain name from the cache entry.

When a cached entry is present in a resolver, a simple lookup on the stored entry makes requests to other resolvers obsolete. Although this reduces the

number of queries sent through the network, it also means that fresh information is not propagated through multiple nodes anymore and the response information exists only in the local cache. The assigned TTL of a cached entry helps to find a reasonable balance between efficient local lookups and the need to request and propagate uncached information through multiple nodes. Such lifetimes vary from seconds to several weeks.

6.2.2. User Perspective

Even though we already find a series of revocation systems in the literature [NDCT07, RD12, Per05b], such approaches do not necessarily share the same priorities with users. A fundamental question in this context is whether users see the need to delete what once was uploaded. In 2018, Khan et al. [KHKU18] conducted several user studies to analyze whether users were interested in removing or unsharing data uploaded to Google Drive or Dropbox. Their results show that users often forgot that specific files still existed in their cloud storages and they decided to delete data in approximately 83 % of cases. Three years before, Clark et al. [CSMK15] experienced similar results when the majority of users decided to delete private photos they once shared using cloud services. The authors use their findings to motivate the concept of *retrospective cloud data management* that offers options to organize files long after their initial upload.

Even though the above studies focus on cloud services, Mondal et al. [MMG+16] experienced different behavior in the context of Twitter. In their studies, users were not always surprised about their previous uploads and often already took care of their older contributions. While users deleted approximately 28 % of their older posts, the authors identified technical issues in the delete options of Twitter: Although the system allows users to delete their posts, it does not take care of related posts like replies and maintains references to removed content. Again, the results of this user study support the assumption that we need mechanisms for long-term handling of online data. Nevertheless, the cases are not entirely clear for all files, and we find the will to retain old content while other data should be deleted entirely [BCK+13].

Mondal et al. argue for an inactivity-based approach that withdraws data at insufficient interaction or missing exposure.

6.3. Related Work

Prior work in the context of revocation systems suggests different technological and operational concepts for handling online data. In the following, we discuss related work on centralized approaches that depend on trusted third parties, and decentralized approaches comparable to *Neuralyzer*. Furthermore, we overview attribute-based encryption as an alternative research direction.

6.3.1. Trusted Party Revocation

The automatic deletion of data is not a new idea, and researchers worked on this concept for quite a while. The first set of works involves trusted parties. Perlman proposed Ephemerizer, a trusted service that ensures the timely expiration of emails [Per05b, Per05a]. Nair et al. [NDCT07] extend the original idea of Ephemerizer with an identity-based crypto system. Systems such as X-Pire! [BBD+11] or the revocable backup system [BL96] also rely on a trusted key management server. Pöpper et al. [PBČC10] presented an approach utilizing porter devices for secure storage of long-term keys including explicit key deletion and forward-secret protocols, even under device compromise. Reimann et al. [RD12] proposed a revocation system that allows for long-term expiration dates of several months after publication. The above systems have a trusted party in common, i. e., they depend on a reliable and secure service point as central component of their infrastructure. This stands in contrast to the *decentralized* infrastructure intended for *Neuralyzer* that is fully independent of the aims of the user and the adversary.

6.3.2. Decentralized Revocation

Vanish [GKLL09] was the first system not relying on a centralized, trusted system for the later revocation of data. Instead, it utilizes a decentralized architecture based on Peer-to-Peer (P2P) Distributed Hash Tables (DHT).

Unfortunately, Vanish is susceptible to Sybil attacks [WHH+10] that can compromise the system by continuously crawling the DHT and saving each stored value before it expires. To overcome the vulnerabilities of Vanish, Zeng et al. [ZSXF10] presented SafeVanish, which extends the length range of Vanish's key shares to substantially increase the attack cost, while it does also some improvement on the Shamir's Secret Sharing implemented in Vanish. In a survey about the vulnerabilities of self-destructing data systems, Geambasu et al. [GKK+11] implemented a framework for testing key-storage mechanisms based on different infrastructures and presented countermeasures to data-harvesting attacks. Casteluccia et al. [CDCFK11] presented EphPub, a system that utilizes the DNS infrastructure and is capable of providing longer lifetimes for objects. Our significant differences to EphPub is that we (*i*) designed a way for refreshing cache entries to prolong the lifetime of an object and (*ii*) applied access heuristics that manage the revocation of objects depending on the access behavior.

6.3.3. Attribute-Based Encryption

Another group of approaches is the Cipher-Text-Policy Attribute-Based Encryption (CP-ABE) schemes first introduced by Bethencourt [BSW07]. With CP-ABE, attribute-based encryption enforces access control policies, as users can only decrypt sensitive data if they provide a defined set of credentials. These credentials relate to group or location attributes, e. g., specific enterprise departments, defined in a policy definition. Different from previous attribute-based encryption systems such as [EC12], attributes are used for a description of user credentials while access policies are defined by the encrypting party instead of storing this information in user keys. Hur et al. [HN11] presented a system providing the enforcement of access control policies even with the revocation of attributes and thus access privileges. Balani and Ruj [BR14] took this concept to the cloud, outsourcing the decryption to a proxy server unable to retrieve information from the computations. Even though these approaches provide encryption-based handling of access policies, they still cannot offer the dynamic revocation of files. That is the underlying

crypto-infrastructure may be ported to a system such as *Neuralyzer* while handling the key information must be performed in a decentralized manner.

6.4. Attacker Model

For the revocation of online data, we consider the rather specific model of a *retrospective* adversary. In contrast to traffic analysis attacks on anonymity systems, the retrospective adversary attempts to recover data only after it reached the expiration date and was revoked. We find two main reasons for this definition. First, we argue that data uploaded to the Internet is intended to be publicly available to a target group of users and should become unavailable only if the user wants to delete it. Second, protecting online data *while* it remains accessible describes an alternative problem statement in which protection systems aim to, e. g., track users who accessed and shared uploaded data. In the context of this work, we focus on the challenge to revoke access *after* a reasonable object lifetime.

Within the concept of retrospective privacy, we define a *local* adversary capable of adding custom messages to the system, e. g., by injecting new, altering existing, or replaying previous messages. However, we assume that the adversary does not monitor the communication channels of the sender and receiver at all times and, consequently, cannot use any targeted actions to recover an object after its revocation. In other words, the adversary cannot monitor all traffic that occurs in connection with the revocation system. In addition to the discontinuous monitoring and altering of system-related communication, the adversary may have access to the internal memory of sender and receiver *after* the lifetime of an object.

In contrast to the above local attacks, a *large-scale* adversary can monitor the traffic of multiple nodes in the DNS infrastructure and increases the knowledge about parts of the key portrayal. While this increases the starting information for, e. g., a brute force attack on the encryption key, it also induces an immense measurement and evaluation overhead.

Wer ein Warum hat, dem ist kein Wie zu schwer.

— Friedrich Nietzsche

7
User-Driven Revocation

Contents

7.1. Introduction

Social media and cloud storage services have changed the information culture of our society. In the era of Web 2.0 people willingly leave lasting digital traces of their lives while decisions on uploading such information are short-termed. In contrast to analog data, these traces remain available as long as the service providers decide to. As a consequence, data such as uploaded documents, communication contents, personal profiles, and posts remain accessible even years after their initial relevance ceased [BBB+13, GA05, Ros07, MJB11]. While the decision to upload personal information to the Internet can be made by each user individually, the control of published data passes to the service provider. Users depend on a responsible privacy policy while they lose the transparency of the storage and provision process in most cases. Besides, many data scandals and insights into their archiving practices [Huf19, The19a, Mas19, Kre19, Mar19] damaged the confidence in the corresponding services. Unfortunately, these tendencies conflict with the users' right to be forgotten [Eur19, Con10].

As there is no solution to regaining control over once uploaded data in external storages, a possible remedy to this problem is proactive user-driven access control. For instance, the timed revocation of data equips users with control over personal information by revoking the access to data at a specific time after its publication, even if external service providers maintain files. Solutions such as Ephemerizer [Per05b], Vanish [GKLL09], and EphPub [CDCFK11] allow users to define a prefixed time when the data will be deleted. All these solutions rely on the very same concept: they encrypt the data and prevent access after the predetermined expiration date by destroying the decryption key. In these solutions, the decryption key is often spread within an existing infrastructure, for example, on distributed hash tables. As long as the key bits can be accessed, the published data remains available. The security goal of all timed revocation schemes is *retrospective privacy*, i. e., the schemes guarantee the revocation of access rights *after* the expiration time.

However, all the previously-mentioned schemes suffer from the same limitation: users should have *prior knowledge of the correct time* when to delete

their shared data. Unfortunately, this is not always feasible [BCK+13], and users' privacy preferences are also likely to change over time [AT13, AT17]. Pictures of the last day trip, for example, remain interesting only until the next event takes place. In reality, knowing beforehand when the image should disappear is a complicated task that can cause additional overhead to individuals, while it remains most of the times without the desired results; the picture may pass before all of her friends have seen it, or it stays available for so long that it has a negative impact.

We address this problem by introducing the concept of *flexible expiration times*, which follows the security goals of *retrospective privacy*. Our concept builds on prior work and continues to use similar infrastructures, at the same time it does not require to predefine an expiration date and therefore removes the respective load from users. Instead, a suitable revocation model triggers the deletion of data, e. g., it expires after interest in the data drops or excessive access forces an untimely revocation. No matter which revocation model a user selects, the shared data will disappear after a period without the user's requirement of choosing this time.

Although the concept of flexible expiration times sounds simple, its actual design and implementation are not straightforward. First, we want an infrastructure that is independent of the various use case scenarios of uploading data, i. e., has no interest in interfering with our privacy goals. Second, we need a distributed infrastructure that allows spreading information over multiple nodes, i. e., to increase the information "surface". Third, we depend on a decay characteristic that destroys information over time to create ephemeral storage but allows us to update entries to extend the lifetime of data if needed.

In this paper, we investigate the problem space of user-driven expiration times and propose *Neuralyzer*, a timed revocation scheme that allows dynamic access control of users' publicly available data. *Neuralyzer* extends existing approaches by applying flexible expiration times while providing retrospective privacy. More specifically, our prototype uses the caching mechanisms of the DNS, similar to the EphPub system [CDCFK11]. To this end, it uses encryption to protect the data and then splits and distributes the parts

of the decryption key over various DNS entries. The key is accessible to anyone who knows which entries have been used for the encoding of the key bits. At the same time, data access leads to the automatic extension of the lifetime of the key bits in the cache of the DNS servers. In essence, the key is valid as long as it is stored in the cache and vanishes once the cache entries are empty. To assess our results, we evaluate the performance of the designed framework regarding data lifetime for different access scenarios, e. g., drop of interest, excessive access, and manual revocation. Based on the results of a simulation study as well as a prototype implementation, we show that our approach provides dynamic access revocation to published data. Overall, we believe that *Neuralyzer* can be an essential building block to protect users from the long-term exposure of their online data. In summary, we make the following main contributions concerning the last research question.

Research Question 4 *How can we offer users tools to control their online data in the presence of large service providers?*

- We identify the limitations of current schemes for the timed revocation of data and introduce the concept of *flexible expiration times* for online data.

- We propose a protocol to revoke the public access to data that should be forgotten based on three different access heuristics: (*i*) drop of interest, (*ii*) excessive access, and (*iii*) manual revocation.

- We assess the feasibility of our approach by implementing and evaluating a working prototype. Our experimental results demonstrate that our prototype is able to successfully destroy data with flexible expiration times.

The above contributions result from a collaboration with Apostolis Zarras, Markus Dürmuth, and Christina Pöpper. In particular, Apostolis Zarras contributed to the prototype implementation of *Neuralyzer* and its core decision mechanisms.

7.2. Design Goals

In this section, we first state the problem we are addressing and introduce the term of retrospective privacy. Furthermore, we describe three different access heuristics as motivation for the concept of flexible expiration times. Finally, we present the threat model used throughout this paper.

7.2.1. Problem Statement

The security goal of our approach is to prevent access to shared data after its expiration time, summarized with the term of *retrospective privacy*. This is achieved under the application of different access heuristics which time the revocation of an object.

Retrospective Privacy. With the publication of information on the Internet, all physical control of data passes to the respective service provider. Timed revocation schemes encrypt valuable information and revoke the access to an individual encryption key once an object should become inaccessible. If access to an expired object is successfully prevented, then *retrospective privacy* is fulfilled. More precisely, the concept allows adversarial actions *after* the access revocation and does not consider attacks during the lifetime of an object.

Access Heuristics. The data should be accessible only for a limited period. Hence, the proposed protocol must revoke access rights after that time. Predefined expiration times are the only revocation technique that has so far found attention for the proposal of technical solutions [Per05b, GKLL09, CDCFK11]. These approaches are, however, independent of the access heuristic. We argue that predefined expiration times have drawbacks regarding appropriateness and user-friendliness (the users may not know the expiration time nor may want to decide on it beforehand) and thus more dynamic revocations schemes are desirable. We can achieve dynamic revocation based on the following types of heuristics that take into account the number of accesses over time.

- *Drop of interest:* In case of fading interest in the uploaded information, we can assume disappearing relevance. Therefore, the system should

detect drops in interest and revoke the accessibility of information to protect its future privacy. In particular, uploads of personal information may be of short-term interest, as such posts are frequently updated and often relate to recent events: A user uploads a picture from an event recently attended, however, does not want to be accessible forever, but only for a period where interest in the event is still present. Through applying the above heuristic, the picture will remain alive by enduring requests, though once the interest drops it will become inaccessible.

- *Excessive access:* Users should be able to revoke the access to publicly available data in case of excessive access. Therefore, a protocol should have the capacity to revoke the access in case of high demand automatically. For instance, an advertising campaign provides free vouchers that should be limited in number. By applying the excessive access heuristic, a maximum number of accesses to shared data can be constrained after which the data ceases to exist.

- *Manual revocation:* Manual revocation of objects is an essential fallback method if an applied heuristic does not cover proper deletion. With the capability of manually revoking data, any applied fixed or dynamic lifetime of an object can expire on demand.

We claim that the described list of access heuristics contains examples of desirable behavior. While technical approaches for all heuristics and evaluations of their applicability are beneficial, in this work, we chose to focus on the first one, i. e., drop of interest, and its technical realization. We later also extend our investigations to the other access heuristics where applicable in the description of our solution and evaluation. We note that providing technical means to address all desirable access heuristics in parallel may not always be possible.

7.3. High-Level Idea

In our proposed model, we define three crucial entities of *Neuralyzer*: (*i*) the ephemeral storage, (*ii*) the sender, and (*iii*) the receiver (cf. Figure 7.1).

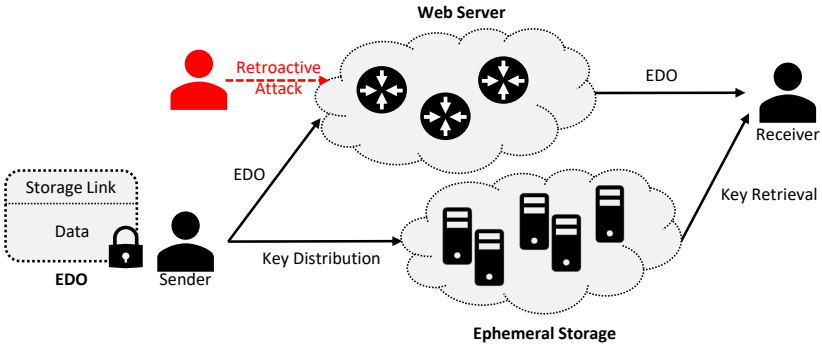


Figure 7.1. *Neuralyzer* system model. Encrypted data (EDO) uploaded to web server of service; access key distributed in ephemeral storage. Receiver recovers data using encryption key; retrospective adversary attempts to recover data.

Ephemeral Storage. The core part of our system is a mechanism to store the decryption key. The key should remain accessible only for a valid period and then disappear without leaving traces that allow the adversary to backtrack the original key. To this end, we define the *ephemeral storage* as a medium which assures that data is accessible only for a valid period and then disappears permanently. As the decryption key is essential for the accessibility of an object during its lifetime, the candidate infrastructure for ephemeral storage must provide consequent up-times.

Sender. The sender must be aware that, as long the ephemeral key exists, everyone with access to it can decrypt the data. The uploaded data to a server that the user does not own, e. g., Google Drive, Dropbox, etc., allows to any receiver as well as the server itself to retrieve, decode, and store it permanently. Previous works made it harder for the server to use the collected data [DCSTW12, PPP+13] accurately, but these approaches are outside the scope of this work. Following our threat model, we assume that the server does not proactively collect this data. If this assumption is not fulfilled, out-of-band channels can be set up and used to distribute the information required for key recovery. With that in mind, the sender compiles the data to a data structure called Ephemeral Data Object (EDO) which contains the

encrypted data and a link to the ephemeral storage that includes information for the construction of the decryption key, and then uploads it to a server from where it can be retrieved.

Receiver. A receiver who retrieves an EDO can decode it and decrypt its data before the key expires. To do so, the key must first be recovered from the ephemeral storage and then be reconstructed locally. Additionally, in our proposed model and following the *drop of interest* access heuristic, the receiver contributes to the viability of the EDO. Thus, it is crucial that the ephemeral storage provides mechanisms for extending the lifetime of the stored key based on the receivers' access. This way, the data will be accessible as long as there is sufficient interest in it and disappear afterwards.

7.4. Scheme Description

In this section, we provide the design details of our approach. We propose the utilization of the DNS resolvers' caching mechanisms as an instantiation for the ephemeral storage. In the following, we first introduce the concept of *ephemeral bits* and then describe the details of the protocol design.

7.4.1. Ephemeral Bits

We refer to standard state of the art cryptography to encrypt the payload of an EDO and share the key information in the ephemeral storage. The size of the key defines the object's resistance to a brute force attack but also increases the required space in the distributed data structure as follows. In our approach, we represent the encryption key bit-wise and spread these bits over the nodes of the ephemeral storage, i. e., larger keys increase the protection capabilities of the encryption while consuming a higher number of entries in the key storage. As we refer to an infrastructure that maintains the key material only for a *limited* time, we refer to key bits as *ephemeral bits*.

In our proposed scheme, we use the caching mechanisms of DNS resolvers to encode the ephemeral bits. Such DNS resolvers provide a caching mechanism that keeps requested entries for a predefined TTL [Moc83] and flushes the

Table 7.1. Notations of Experimental Setups.

Explanation	Notation
Sender	S
Receiver	R
Length of DNS portrayal	N
Length of decryption key	I
Key threshold	x
TTL thresholds	t_1, t_2
Key of length I	$\|K\|_I : k_i \in K$
Recovered key of length I	$\|K'\|_I : k_i \in K'$
Cache entries for key	$\|C\|_{I \times N} : c_{i,n} \in C$
Domains of cache entries	$\|D\|_{I \times N} : d_{i,n} \in D$
TTL values of cache entries	$\|TTL\|_{I \times N} : ttl_{i,n} \in TTL$

entry after this time expired. We make use of this TTL and encode key bits in cached and uncached DNS entries, e. g., depending on the predefined timeout the key material destroys itself.

7.4.2. Protocol Description

Our approach allows data to vanish once the interest for it drops. This way, if a retrospective adversary attempts to access the data after that point, it cannot be recovered anymore. To do so, we introduce the data structure EDO that protects the publicly accessible data by encrypting it, encapsulating the encrypted content, and ensuring its disappearance when reaching the expiration time. The data contained in the EDO becomes useless after the expiration of the key material, even if an adversary retrospectively obtains a valid copy of the EDO. Overall, the lifetime of the EDO is divided into three phases: (*i*) construction, (*ii*) access, and (*iii*) revocation. For convenience, Table 7.1 summarizes the notation we use throughout this section.

Construction Phase

In this phase, a sender S uploads an EDO to a web server (target application), which will become inaccessible after the interest for it drops. For each new EDO, our algorithm executes the following steps:

C1. First, S generates a random key. The size of the key defines the security of the encrypted data, and therefore we recommend at least a 128-bit key. Then, S uses this key to encrypt the data using the AES algorithm.

C2. Next, S converts the key bits into ephemeral bits. Specifically, each ephemeral bit refers to a list of DNS entries, which we call *DNS portrayal*. Each DNS entry represents a precompiled domain name and a DNS resolver (cf. Figure 7.2). Precompiled domains can be generated either by crawling the web or by selecting random IP addresses and performing reverse DNS lookups (cf. Section 7.4.3 for details). Note that the selected domains must not be preloaded in the cache of the associated DNS resolvers. With that in mind, our algorithm can assign the value `1` in a DNS entry by performing a *recursive* DNS request setting the cache entry.

For all bits k_i of the key, we proceed as follows: We do not take any action if $k_i = \texttt{0}$, apart of a non-recursive DNS query verifying that the respective DNS entry is indeed not set. For every $k_i = \texttt{1}$, we execute recursive DNS queries for the respective domain names defined in the EDO. As a result, values of the DNS portrayal for each key bit `1` are set to active (loaded in the cache), while the DNS portrayal for each key bit `0` remains unset (corresponding domain names are not in the cache).

Note that we do not initially assign the value `1` to *all* entries in the DNS portrayal if $k_i = \texttt{1}$, but only to a certain number of them. We want to avoid that all DNS resolvers happen to clear their caches at the same time, which would prevent key updates during the access phase (detailed further below). Thresholds define the length of the DNS portrayal (N) and the number of entries required for successful

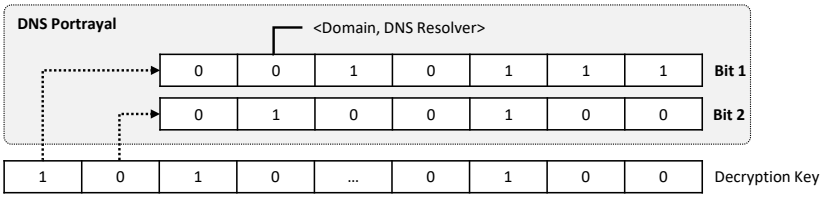


Figure 7.2. Key distribution of decryption key bits in DNS portrayal. One key bit is represented through defined number of entries in DNS portrayal, each entry refers to the domain and open DNS resolver.

representation of an ephemeral bit (x). We discuss the threshold values in Sections 7.4.3 and 7.5.

C3. In this final step, our algorithm compiles the EDO. An EDO contains the encrypted content and the list of domains D for each key bit k_i, represented by N cache entries $c_{i,n}$. As we will present in the access phase, this information is sufficient for successfully decrypting the EDO during its lifetime.

Access Phase

Once the sender compiled the EDO, it can be distributed to third-party servers. A receiver R can retrieve the EDO and access its content as follows:

A1. First, R retrieves the encrypted content and extracts the list of domains D for each k_i from the EDO.

A2. To assign the correct values to each ephemeral key bit, our algorithm performs *non-recursive* queries to the DNS resolvers for their corresponding domain names. If the resolver contains the domain in its cache, our algorithm assigns the value `1` to this DNS entry, otherwise the value `0`. To minimize the errors that may occur from externally modified entries, we use an empirically calculated threshold x. If the sum of the returned values exceeds this threshold, the algorithm sets the corresponding ephemeral key bit (cf. Equation 7.1). Such errors

may be induced by random recursive DNS requests leading to $0 \rightarrow 1$ switches (due to DNS queries from users external to our scheme during the execution of the protocol), or failures at the DNS cache resulting in $1 \rightarrow 0$ switches (the DNS server has emptied its cache, for example, due to a reboot).

A3. In this step, our algorithm extends the lifetime of EDO. To do so, it updates a random DNS entry of each DNS portrayal that represents an ephemeral key with the value `1` by executing a recursive request. However, we want to have variation in the remaining TTL values on the entries of each portrayal. This minimizes the danger of having resolvers that simultaneously empty their caches or at similar times. We achieve this by performing a recursive DNS request when the median or minimum TTL per DNS-portrayal is less than a preselected threshold. For instance, we can update the DNS entries if we see that the median TTL is less than $\text{TTL}_{max}/2$, or if the DNS entry with the minimum remaining time to empty its cache is less than $\text{TTL}_{max}/10$, where TTL_{max} is the maximum value assigned to the TTL by the DNS servers. It is worth to mention here that both conditions work in parallel and we update a DNS entry whenever one of these conditions, or both of them, is satisfied. In essence, without refreshing the cache entries, an object would expire as soon as a significant amount of 1-bit representations has switched from 1 to 0. Overall, to prolong the initial lifetime limit, each receiver performs a cache refreshment after a successful reconstruction of the EDO for all key bits $k_i = 1$ (cf. Equation 7.2). The `refresh` operation is only executed in case the median TTL $\texttt{median}(ttl_{i,n})$ or the TTL of a single value $ttl_{i,n}$ for a key bit k_i fall below thresholds t_1 or t_2. In this case, a random $c_{i,n}$ currently 0 is set to 1 by a recursive DNS request to the respective domain $d_{i,n}$. Note that the threshold values as well as the metrics (median/minimum TTL) can be adapted to a specific deployment scenario and are not necessarily bound to the above definition.

A4. In the final step of this phase, the ephemeral key has been successfully reconstructed. Then, the receiver uses this key to decrypt and access the encrypted data.

$$\texttt{recover} \begin{cases} \sum_{n=0}^{N-1} c_{i,n} \geq x : 1 \\ \sum_{n=0}^{N-1} c_{i,n} < x : 0 \end{cases}, \tag{7.1}$$

$$\texttt{refresh} \begin{cases} (\texttt{median}(ttl_{i,n}) < t_1) \vee (\exists n : ttl_{i,n} < t_2) : 1 \\ (\texttt{median}(ttl_{i,n}) \geq t_1) \wedge (\forall n : ttl_{i,n} \geq t_2) : 0 \end{cases}, \tag{7.2}$$

Revocation Phase

This is the last phase in the lifetime of an EDO. We expect that, after some time, there will be a drop of interest for a published EDO. Consequently, the number of accesses decreases as well and the number of `refresh` operations (step *A3*) falls below the minimum required amount. This results in cleared caches of the DNS entries, which leads to bit flips of cache entries and a false reconstruction of the key. Consequently, the encrypted data remains on the external web servers but can no longer be decrypted by any user.

Alternative Access Heuristics

While we introduced the key mechanisms of *Neuralyzer* for the example of a drop in interest, we now discuss how alternative access heuristics can be realized.

Excessive Access. In this case, we want to revoke the access of an EDO if the interest for it exceeds a particular upper bound of allowed accesses. For this reason, we need to count the number of accesses to the EDO. As DNS resolvers do not have an access counter that would be visible or accessible to regular users, we can enrich the EDO with a probabilistic self-destruction mechanism. More precisely, for every access to the EDO, we generate a random number. The decision to destroy the decryption key is based on whether the generated random number is larger than a specific bound defined by the highest number of acceptable accesses, e. g., for a allowed accesses,

the bound would be $1 - 1/a$ if the random numbers are selected from $[0, 1]$. If the result exceeds the defined bound, we manually destroy the ephemeral key by performing recursive DNS requests for all key bits. This causes all the ephemeral bits to take the value `1`. In other words, this approach is like a dice with N sides, and if we throw the proper side the ephemeral key is destroyed. Covering an excessive access model requires the user to define an upper bound for the number of requests, which is added to the EDO.

Manual Revocation. Additionally, our approach supports the revocation of an EDO at a time its creator decides to. This can be done by performing recursive DNS requests to all entries of the portrayal. Even though this raises the threat of a denial-of-service attack, in which the adversary destroys the key on purpose, we emphasize that this scenario is not covered in the retrospective attacker model.

7.4.3. Instantiation of the Scheme

Central to the application of the proposed scheme are the list of domain names used in the construction of the EDO and the length and threshold of the DNS portrayal. We detail both in the following. Furthermore, we reason about the error correction capabilities and possible scalability issues of the proposed scheme.

List of Precompiled Domains

We collect the domain names of our prototype automatically by using reverse DNS lookups. This method is based on the DNS infrastructure and allows the resolution of an IP address to its designated domain name, also known as forward DNS resolution. To generate a list of domains, we perform reverse DNS lookups to a range of IP addresses. In this procedure, we exclude addresses that reserved for special purposes by the Internet Engineering Task Force (IETF) and the Internet Assigned Numbers Authority (IANA). Before employing the domains, we use non-recursive DNS lookups to ensure that the domains are not currently cached in the DNS resolvers.

An alternative way to generate such a list is by crawling the Internet [RD12], using heuristics to reach less likely used websites. In both cases, our list of

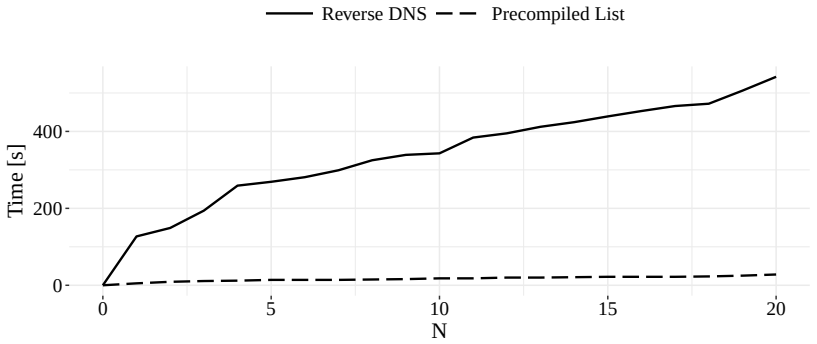


Figure 7.3. Comparison of construction times. The time to construct an EDO through reverse DNS lookups increases for longer DNS portrayals.

domain names should contain rarely used sites to reduce the chance of interference from legitimate DNS lookups. However, the randomized reverse DNS lookup approach above has the advantage that different users are more likely to select different domains so that their key storages do not interfere with each other.

Figure 7.3 illustrates an example of EDO generation with and without a precompiled list of domains for different DNS portrayal sizes N. The time consumption increases with a higher number of lookups. Measurements of 1024 random lookups, e. g., as required for $N = 8$, indicate an average duration of 0.31 s per successful operation. In this setup, encountered no collisions of domains. We note that the sender S can compile domain lists in advance to prevent bottlenecks in the publication of data.

Size of DNS Portrayal

The length N and the threshold x of the DNS portrayal determine the number of errors that our scheme can handle, and both parameters influence the maximum lifetime of an object. We consider any unexpected modification of the cache of a DNS resolver as a possible error. Examples for this are (i) a DNS resolver that empties its cache before the defined TTL expires (bit flips

to 0) or (*ii*) an "accidental" recursive DNS request to our selected domain (bit flips to 1). To find representative error numbers, we monitor how DNS entries behave in the real world. Our measurements of error frequencies represent a set of 1853 DNS resolvers that showed reliable service throughout several months. We monitor the cached, and uncached entries of 1000 randomly selected domains and collect all errors that occurred during the entire lifetime of an entry. Results show that (*i*) with an error probability of $e_{1\to0} = 0.7\,\%$ a resolver empties its cache before the scheduled TTL expires and (*ii*) $e_{0\to1} = 0.1\,\%$ of accidental recursive requests were performed on the observed set of domains during our measurements. Although the exact error rates may vary over time, these numbers provide reasonable estimates for our simulation.

Both types of errors, i. e., bit flips $e_{1\to0}$ and $e_{0\to1}$, should be handled by our scheme. Based on this insight, we can derive a minimal parameter setup $N = 3$ and $x = 2$ that enables correcting errors of type (*i*) and (*ii*) while providing the capability of refreshing cache entries. During the initialization, i. e., when the key bits are first stored in the portrayal, the number of bits set to 1 must be larger than x; in later investigations, we use $\lceil \frac{N}{2} \rceil$ 1s for the initialization.

Beyond the minimal setup, an increase of the portrayal length N allows for increasing the scheme's correction capability, whereas an increasing x threshold leads to a higher overhead in the presentation of bits (cf. Equation 7.4).

The initial lifetime (before lifetime extensions due to new accesses to the object) are given by the TTLs. Typical TTL values we observed for 1000 random lookups are characterized as follows: median of 86,400 s = 24 h, minimum 72 s, maximum 604,800 s = 14 d, standard deviation 110,704 s ≈ 30 h. Due to variations in the TTLs as well as due to errors, we initially do not set only one but multiple entries in each portrayal to 1.

Error Correction

Errors can destroy an encryption key before the heuristic would have triggered the revocation of an object. The use of a correction scheme should compensate such errors as long as an object is legitimately available. We consider the regular expiration of an entry as type-*(iii)* error for this pur-

pose: as long as the applied heuristics do not trigger the destruction of the key, all changes in the key representation should not lead to the expiration of an object.

As the expiration of an entry is a necessary event, the error probabilities for $e_{1\to0}$ and $e_{0\to1}$ errors are distinct. Previous revocation systems utilized different correction schemes: EphPub [CDCFK11] uses Reed-Solomon codes and Vanish [GKLL09] uses Shamir's Secret Sharing. The following paragraphs compare the performance of both codes with the key portrayal applied in our scheme.

Error Correcting Codes. The Reed-Solomon codes are optimal for correcting *burst* errors, i. e., bit flips that occur in a row in an encoded word. In our scheme, the majority of type-*(iii)* errors occur randomly in 50 % of the cache entries and—since DNS servers and domain names are picked and distributed using a random selection—are uniformly distributed rather than bursty. An alternative to burst-optimal Reed-Solomon codes and Golay codes [BFvT02]. The extended binary Golay code can correct uniformly distributed errors and would be more suitable for the occurrences of errors in our scheme. Different to standard error correction schemes, the parameters of our portrayal can be adapted to the type of errors occurring, i. e., the asymmetric distribution of $e_{1\to0}$ and $e_{0\to1}$ errors. Given this characteristic, it is possible to achieve high correction rates while applying a smaller overhead to the key representation. Concerning the applied overhead and correction capabilities, the portrayal outperforms standard error correction schemes. Thus, the portrayal is selected for providing robust lifetimes and the possibility of refreshing the key representation of an object.

Shamir's Secret Sharing. Secret sharing schemes distribute portions of a secret message over several users where a threshold s defines the number of shares required for reconstructing the original secret. When applied to an encryption key, the key can only be recovered when at least s shares are available at access time. Threshold values close to the number of shares lead to fast revocation and high security while smaller s values lead to a more robust system that can survive a higher number of errors. Overall, secret-sharing can provide theoretical security for revocation schemes that

do not rely on the refreshing of entries: as soon as a bit error occurs in one share it becomes invalid and cannot be used for reconstructing the key. As our scheme requires the refreshing of bits, it must be robust to bit errors occurring through the expiration of cached entries.

7.4.4. Scalability

As *Neuralyzer* does not use any centralized component, we cannot track domains that are already in use for the key portrayals of existing objects. Consequently, it is not transparent whether an uncached entry for a domain is currently unused or represents a 0-entry or erroneous 1-entry in another portrayal. The probability of overlapping with an existing portrayal depends on the number of active domains, e. g., for 510 million domain names [Int19] and a minimal portrayal with $N = 3$ and $x = 2$, overall 1,328,125 parallel users can share data via our model. The probability of overlapping entries at the initialization is approximately 1 % for 28,000 parallel users, as 50 % of bits in a portrayal are `0`-bits that are prone to overlap with entries that are already used.

We note that high numbers of parallel users can increase the original probability for $e_{0\to1}$ errors in addition to the error rate induced by accidentally performed recursive requests. To increase the robustness of our scheme for scenarios of massive parallel use, an additional encoding can be applied. For instance, in the portrayal of the encryption key, each bit can be encoded by a mixed tuple. A possible implementation could use a Manchester encoding, where `0`-bits are encoded by `01` while `1`-bits are encoded by `10`. This allows to detect all `0` key bits that are already in use under the expense of doubled overhead.

7.5. Simulation Study

We conduct a simulation study to explore the parameter space for three performance parameters for the access heuristics of dropping interest and excessive access.

Table 7.2. Overview of Parameter Ranges.

Simulation		Empirical	
Parameter	**Space**	**Parameter**	**Space**
I	128	$e_{1\to0}$	$[0, 0.007]$
N	$[4, 5, ..., 10]$	$e_{0\to1}$	$[0, 0.001]$
x	$\{1, 2\}$	TTL_{min}	$1,200$
t	$43,200$	TTL_{max}	$604,800$

7.5.1. Simulation Setup

Table 7.2 summarizes the setup parameters of our experiments. They include the simulation parameters (portrayal length N, key recovery thresholds x and t, fixed key length I) that we adjust in our simulation model, and the empirical parameters (bit flip probabilities $e_{0\to1}$ and $e_{1\to0}$, TTL values TTL_{min} and TTL_{max}). We learn the occurrences of bit flip errors from measurements on the set of possible DNS resolvers (cf. Section 7.4.3) and refer to the distribution of TTL values obtained in the work of Casteluccia et al. [CDCFK11].

In a simulation run, we monitor the lifetime of an object beginning with its initialization and ending with its revocation through the destruction of the encryption key. We execute the simulation step-wise with each step representing one second of object lifetime and simulate the interest in an object with user requests following an exponential (excessive access) or shifted normal distribution (dropping interest). In other words, we use a probability distribution function to estimate the number of user accesses in each simulation step. These access numbers trigger the refreshing of key information and lead to a revocation of the object through expiring lifetimes of DNS cache entries.

7.5.2. Performance Parameters

We measure the results of a simulation run using the following performance parameters:

- *Sensitivity of expiration*: The sensitivity of expiration refers to the point of time when the access to an object gets revoked. While systems

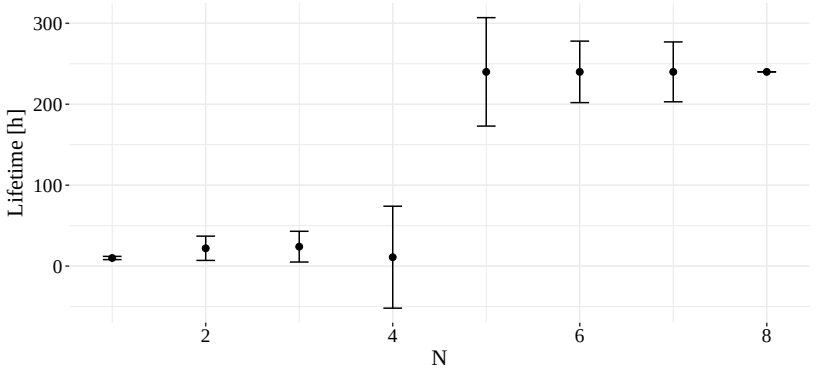


Figure 7.4. Distribution of object lifetimes. Increasing portrayal sizes N (points) help to prolong the lifetime of an object and limit the standard deviation for random repetitions (error bars).

with high sensitivity react to the underlying metrics at an early point in time, more robust parameter setups extend the lifetime of an object.

- *Reliability of results*: The lifetime of an object may vary in multiple simulation runs with identical parameter setups, as the system dynamics follow a probabilistic model. We analyze the continuity of object lifetimes for highly sensitive as well as robust parameter setups.

- *Error correction capabilities*: Based on the optimal threshold $x = 2$, the correction capabilities can be increased by adapting the portrayal size N. Therefore, a variation of the portrayal length leads to different object lifetimes.

7.5.3. Sensitivity of Expiration

Figure 7.4 summarizes the object lifetime for an increasing portrayal length N at the optimal threshold $x = 2$. With an increasing ratio of $\frac{N}{x}$, the correction capabilities increase and lead to a prolonged object lifetime. We can make use of this characteristic for defining the sensitivity of our scheme,

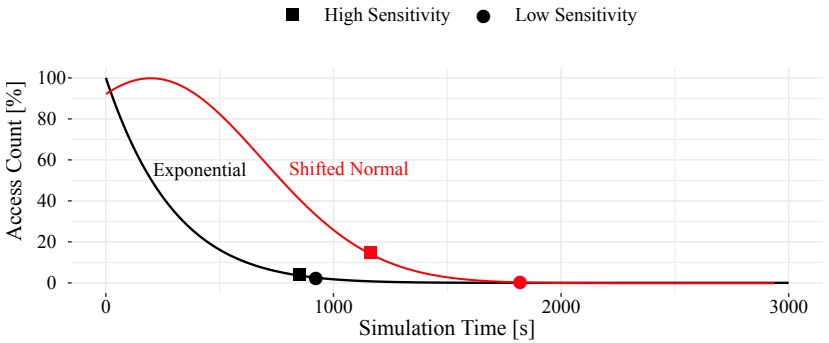


Figure 7.5. Simulation of lifetimes. We influence the point of destruction through the sensitivity parameter of the revocation mechanism. Exponential or shifted normal distribution represent a scenario with dropping interest.

i. e., to provide a robust revocation mechanism we need a ratio of at least 2.5 while smaller ratios lead to a system sensitive to a lack of refresh operations.

We measure the expiration times of an object with respect to exponentially and normally distributed access patterns (Figure 7.5). To assign an object with *low* sensitivity to dropping access rates, we simulate the object lifetime with $N = 10$, $x = 2$ and therefore a high ratio $\frac{N}{x} = 5$. For a scenario with access rates similar to a normal distribution, the object expires as soon as the number of users is close to zero. In a parameter setup with increased sensitivity and $N = 4$, $x = 2$, $\frac{N}{x} = 2$ the expiration is triggered earlier, as a reduced number of expirations in cache entries can be covered. For a scenario with exponentially distributed access rates, a highly sensitive parameter setup would require the choice of more restrictive parameters.

In contrast to a dropping interest over time, the scenario of excessive accesses requires the destruction of the key information in case the interest in the uploaded object increases dramatically. Based on a probabilistic approach, we can reduce or prolong the expiration of an object by adapting the *probability of destruction*. As shown in Figure 7.6, higher probabilities lead to early destruction of key information while reduced probabilities prolong the object lifetime.

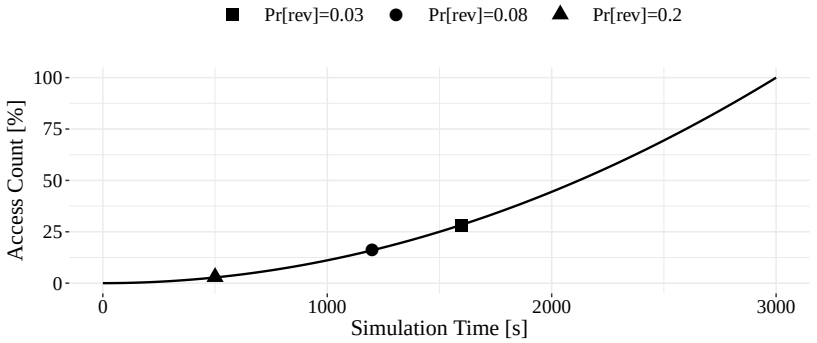


Figure 7.6. Simulation of lifetimes for $x = 2$, $N = 10$ and excessive access. The random destruction of an object allows to define the sensitivity of the revocation mechanism.

7.5.4. Reliability of Results

Given the above parameter setups for a sensitive or robust revocation, we analyze the reliability of results for multiple repetitions. As shown in Figure 7.5, the dynamic characteristics of the simulation model lead to variations in the object lifetime, e. g., for 50 repetitions with TTL values distributed according to the measurements of average lifetimes, the results show slight deviations in the expiration time. Nevertheless, the overall range is limited leading to the reliable destruction of key information for a given parameter set. Based on these characteristics, it is possible to initialize an object with a target sensibility of the drop of interest heuristic.

7.5.5. Error Correction Capabilities

The x threshold can correct a limited amount of errors. We test different x-thresholds with a key portrayal of size $N = 20$, which stays within an acceptable creation time. We apply the best performing setup to a simulation scenario with increasing error rates and accumulate the results for 100 random repetitions. An average error probability of 10^{-4} already leads to

Table 7.3. Object Lifetimes.

	x	0	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
Lifetime [h]	**0**	875	875	760	100	10	0
	2	875	875	875	875	400	60

a significant decrease in the object lifetime when no correction threshold is applied (Table 7.3). However, $x = 2$ provides a constant lifetime up to an error rate of 10^{-3} and still allows for a performing system at 10^{-2}, which is beyond the realistic error rates measured in the prototype implementation.

Overall, the simulation results lead to the conclusion that an application of the drop of interest heuristic is possible with our scheme. Furthermore, a variation of system parameters such as the x-threshold enables for adjusting the sensitivity of the applied heuristic leading to a shift in the expiration time.

7.5.6. Comparison of Correction Capabilities

In the following, we compare our portrayal **P** to the Reed-Solomon **RS** and Golay **G** error correcting codes with respect to each scheme's individual overhead. To do so, we define a fixed number of errors $T = 1000$ and a key length $z = 128$ to analyze the required overhead, i. e., the number of additional bits required for a code that is capable of correcting the fixed error rate, to correct this number of errors. For a binary input of length z the Reed-Solomon code **RS** can correct errors as follows.

$$\mathbf{RS} \begin{cases} \frac{n-k}{2} \cdot \lceil \frac{z}{k \cdot m} \rceil & : n-k \text{ even} \\ \frac{n-k-1}{2} \cdot \lceil \frac{z}{k \cdot m} \rceil & : n-k \text{ odd} \end{cases}, \tag{7.3}$$

where $n = 2^m - 1$ with $7 \leq n \leq 2^{16} - 1$ is the codeword length, $3 \leq m \leq 16$ is the input word length, and $k < n$ is the number of words to be encoded in one code word. In comparison, our portrayal P is defined as follows.

$$\mathbf{P} : z(N - x - 1), \tag{7.4}$$

where N is the length of a portrayal, and x is the correction threshold. To be robust against flipping of bits in both directions, the total correction capability for $e_{0\to1}$ must be at least 1. This requirement is fulfilled for $x \geq 1$ and $N \geq 3$. The parameters for **G** are fixed and allow up to 3 corrections for the extended binary $G(24, 12, 8)$.

A **RS** code is optimal for non-binary input only and can be adapted to encoding the binary encryption key by setting $m = 8$ for representing 1 B per word. Under this assumption, the overhead and correction capability of **RS** (assumed $n - k$ even) is as follows for $T = 1000$:

$$\begin{aligned} T &\leq \frac{n-k}{2} \cdot \lceil \frac{z}{k \cdot m} \rceil && (7.5)\\ \Rightarrow 1000 &\leq \frac{255-k}{2} \cdot \lceil \frac{128}{k \cdot 8} \rceil | \cdot 2 && (7.6)\\ \Leftrightarrow 2000 &\leq 255 - k \cdot \lceil \frac{128}{k \cdot 8} \rceil | k = 1 && (7.7)\\ \Rightarrow 2000 &\leq 2032 && (7.8) \end{aligned}$$

To correct at least $T = 1000$ errors, **RS** must be applied with $m = 8$, $n = 255$, $k = 1$ leading to a total overhead of factor 16.

For any T and an input length $z = 128$ the overhead of **RS** is as follows:

$$2T \leq (n^m - 1 - k) \cdot \lceil \frac{128}{m \cdot k} \rceil \qquad (7.9)$$

For a minimum threshold of $x = 2$, **P** provides the following overhead for $T = 1000$:

$$\begin{aligned} T &\leq z(N - x - 1) && (7.10)\\ \Rightarrow 1000 &\leq 128(N - 2 - 1) && (7.11)\\ \Leftrightarrow 1000 &\leq 128N | N = 8 && (7.12)\\ \Rightarrow 1000 &\leq 1024 && (7.13) \end{aligned}$$

This leads to a parameter set of $N > 8, x = 2$ and a total overhead of factor 9 for the portrayal scheme. The correction capability for any error rate T for an input of length $z = 128$ is as follows:

$$\frac{T}{128} \leq N \tag{7.14}$$

For providing the same minimum correction rate with $\mathbf{G}$, the input length must be extended: to correct at least $T = 1000$ errors, an input length of 334 is required leading to a total overhead of factor 63.

7.6. Prototype *Neuralyzer*

To demonstrate the viability of our approach, we implement our proposed scheme as the framework *Neuralyzer*. Our prototype is capable of dynamically encrypting data with a randomly generated key and then distribute the key bits across multiple real-world DNS resolvers. To this end, we the `PyDNS` module for recursive and non-recursive DNS requests and `PyCrypto` for an AES implementation with a standard key size of 128 bit. Furthermore, we encode each EDO in a `base64` format to limit its content to ASCII conform strings.

7.6.1. Experiments

While the simulation study of Section 7.5 covers the performance of our revocation scheme in a controlled environment, we face additional influencing factors in a real-world deployment of *Neuralyzer*. In the following, we analyze the performance of our prototype implementation in realistic experiments focusing on the reliability of expiration times, the ability to publish long-lasting data, the accuracy of probabilistic revocation, and the liability of a manual revocation.

Expiration Time

The expiration time of an object depends on the TTL of cache entries and their susceptibility to bit flip errors; while we assume empirical parameters

for this in the simulation setup, we must also prove the reliability of our approach with real-world resolvers.

To examine whether our protocol design is accurate, we create two sets of 100 EDO documents each. While the first set of objects remains unaccessed, we use 2 to 10 random accesses per hour, e. g., resembling the typical lifespan of a tweet [Wis19], to refresh the key material of the second set. We monitor the TTL values of all key entries through non-recursive DNS queries to the corresponding servers. This way, we do not change entries in the key portrayal and can measure the object lifetimes. In both scenarios, i. e., with and without user access, the portrayed keys expired during the expected time. Keep in mind that the TTL of the caches define for how long an entry remains available after the last access. In the first scenario, the lack of access and, consequently, refresh actions, limited to 1 h with a majority of resolvers providing a TTL of 3600 s. In the second scenario, the keys remain available during a sufficient number of access actions and prevent access to the object as soon as too many bits of the key portrayal flip from `1` to `0`. Overall, we experience a predictable expiration of the key information that leads to a reliable revocation of access to the uploaded object.

Long Lasting Data

In contrast to short termed data, e. g., social media updates or image uploads, other use cases require a long-term availability of information for several weeks or months. Nevertheless, the revocation mechanism must still assure the expiration of the key information following one of the access heuristics. In the following, we analyze the long-term capabilities of *Neuralyzer*.

To examine the behavior of long-lasting data, we create EDO documents and access them during a period of 33 d. We follow the scenario of a drop in interest over time, i. e., we begin with approximately 400 accesses per day that diminish over time (shifted normal distribution). Our results show that the refresh mechanism of *Neuralyzer* helps to maintain the access to an object even over longer periods, as the uploaded data remained available for one month (cf. Figure 7.7).

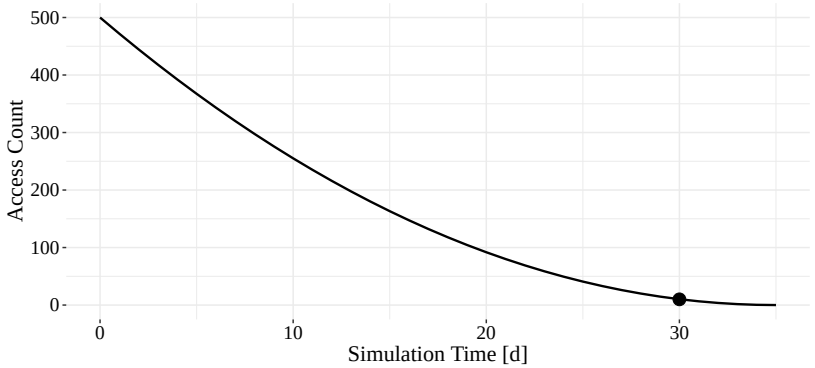


Figure 7.7. Long-term simulation of dropping interest. Results show the number of accesses for a period of one month. The destruction of the key portrayal triggers a revocation after 30 days.

We conclude that the error correction capabilities and the refresh capabilities of *Neuralyzer* help to provide reliable revocation for short-term *and* long-term data using real-world DNS resolvers.

Excessive Access

Most of the times users do not care about the publicity of their exposed information, e. g., status updates or tweets. However, there exist cases in which the number of accesses plays an important role. The DNS infrastructure does not allow us to utilize or introduce a counter for the number of accesses to our publicly available EDOs. Therefore, we use a probabilistic solution that allows us to address this problem (Section 7.4.2). In the following, we analyze the accuracy of this probabilistic destruction of the key information. For this reason, we create 10,000 EDOs and define a maximum number of 10,000 accesses. In our experiments, we measure the average number of accesses and the standard deviation of the time when the key to an object was destroyed.

Our results show the destruction of an EDO after approximately 9928 accesses on average with a standard deviation of 3598 accesses (cf. Figure 7.8).

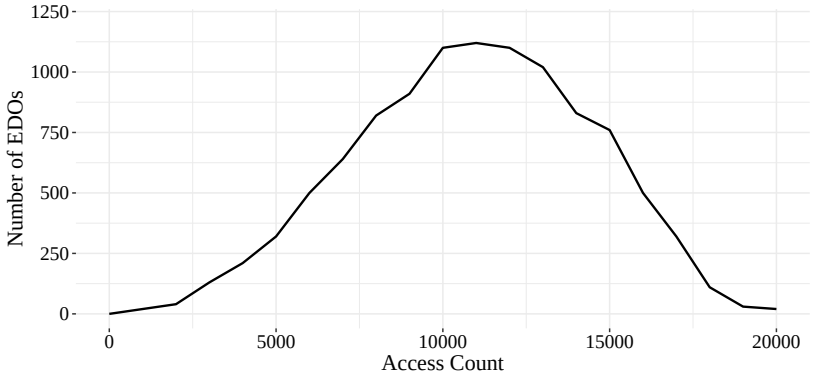


Figure 7.8. Standard deviation for the self destruction of EDOs. On average, an EDO is destroyed after 9928 accesses with a standard deviation of 36 %.

This deviation is in the range of 36 % of the defined maximum number of requests, i. e., we see that exact destruction cannot be achieved with the probabilistic approach . Nevertheless, the average result proves a result close to the defined threshold and indicates that a sufficiently reliable access limit can be defined.

Manual Revocation

While the general motivation of *Neuralyzer* is the flexible revocation of uploaded data, the ability to manually revoke access offers more freedom to the user. In the following, we analyze whether the technical realization of our approach allows performing such manual destruction of the key information. To do so, we create 200 EDO documents and divide them into two sets of 100 documents per set. In the first set, we perform a manual revocation immediately after the creation of the encrypted documents while in the second set we perform the same action *after* they already were accessed. This way, we take into consideration (*i*) the case that the creator of an EDO document wants to recall the access to the document right after its upload to a remote server, and (*ii*) the case that the document already was accessed before the

owner decides to revoke it. As expected, in both cases we were able to destroy the decryption key in all 200 documents completely.

7.7. Discussion

We now discuss the presented scheme concerning the security goal of retrospective privacy and outline further attacks on the system.

7.7.1. Retrospective Privacy

Retrospective privacy is fulfilled if an adversary is unable to access the contents of an EDO after its expiration. Under the assumption of a secure encryption system, this is possible in case the adversary manages to recover the decryption key of an object. Furthermore, we address possible security issues of the DNS infrastructure.

Key recovery

Following the assumptions of Section 6.4, the adversary gains access to an encrypted EDO as well as the respective list of all destinations to where the key share was distributed after the object expired. As we use a DNS implementation, this results in a set of cache entries that previously represented the encryption key.

In an attack, the expired and therefore still encrypted objects along with the list of key share domains are accessible to the adversary. At this point, the information embedded in the acquired EDO is only available if the adversary manages to recover the required amount of key bits from the list of queried domains for reconstructing the encryption key. The chance of successfully guessing the encryption key after expiration decreases over time.

Targeted guessing. Short after the expiration, a high number of cached `1`-entries is still valid and allow the separation of `0` and `1` bits based on an assumed threshold value. In the standard portrayal, a threshold of $x = 2$ is optimal for the given error characteristics. Even though the correction capabilities for $e_{1\to0}$ errors increase when using larger portrayal lengths N, the x threshold must be at least 2 for providing the correction of $e_{0\to1}$ errors.

Therefore, an adversary can assume $x = 2$. The number of valid `1` entries decreases over time, which increases the indistinguishability of key bits. Even though the threshold cannot be fulfilled correctly after some time, the existence of a `1`-entry is more likely to be the remaining cache entry of a `1`-bit of the key rather than a $e_{0\to1}$ error. Targeted guessing may lead to success if a sufficient number of `1`-caches still exist. Both the possible assumption of the optimal x threshold as well as targeted guessing of entries highly increases the chance of an adversary to successfully recovering the encryption key. Nevertheless, the overall success highly depends on the amount of time that passed between the revocation of an object and the actual attack on the key information.

Time. Over time the number of remaining `1`-entries in the cache decreases, which also reduces the probability for an adversary to successfully recover the encryption key from remaining entries. We analyze the period in which a successful attack is likely based on the frequency of common TTL values.

As soon as the expiration of one entry $TTL(c_{exp})$ leads to the destruction of the key bit, a set of remaining `1`-entries $c_{rem} \in C_{rem}$ must have had the same or higher TTL values at the initialization of c_{exp}:

$$\exists c_{rem} \in C_{rem} : TTL(c_{rem}) \geq TTL(c_{exp}) \tag{7.15}$$

That is when c_{exp} was initially set and later triggered the destruction of the encryption key, all remaining entries that are also cached cannot have a TTL that is lower than that of c_{exp}. Due to this restriction, we distinguish two cases: In case (a) the majority of remaining entries has a TTL identical to c_{exp} and, therefore, expires at the same time. In this case, the amount of information left for the adversary is insufficient for efficiently guessing the encryption key. In the second case, (b) a majority of elements in C_{rem} has a TTL higher than the expired one. Based on the measurements on common TTL values, a lifetime of 86,400 s (1 day) can be assumed for 58 % of cache entries, the next greater TTL is $86,400 * 2$ and provides another day to guess the remaining entries of the key. After that, the data is irrecoverably gone.

Security issues of the DNS infrastructure

Even though *Neuralyzer* uses the distributed and decentralized infrastructure provided by public DNS resolvers, several characteristics of this infrastructure can lead to security issues. Kührer et al. [KHB+15] found that in 2015 a significant amount, i. e., that is 20 % of overall 6,753,748 resolvers in that specific measurement, of public resolvers run with BIND version `9.8.2`, which can be manipulated through a remote code execution vulnerability. They also showed that the top 25 networks host a majority of public DNS resolvers wherein at least 20 offer telecommunication and broadband Internet services. Even though these results do not necessarily imply malicious behavior of service providers or the monitoring of traffic through a BIND vulnerability, the above can still lead to security issues for *Neuralyzer*. To overcome the threat of remote code execution, a fingerprinting of resolvers and the according software should be performed. When detecting a software version that enables an adversary to perform monitoring of traffic on the public resolver, it should be excluded from the list of nodes that are considered in the portrayal of key bits.

In a scenario where the adversary can make use of the above issues, we must assume that the traffic of a fraction of resolvers in a portrayal can be monitored. This monitoring enables the adversary to save a history of cached/uncached entries during the lifetime of an object. In a retrospective attack, this knowledge can help to reconstruct the encryption key from the remaining cache entries. As summarized in Section 7.7.1, even limited experience about the content of a portrayal can increase the probability of successfully guessing the related key bit dramatically. A simulation study on the number of monitors required for learning about the encryption key shows that also smaller amounts of malicious resolvers are sufficient for reconstructing a significant portion of the key (as discussed below). When extending the current filtering of resolvers to a selection of maximum distributed hosts of public resolvers, the threat of centrally controlled nodes can be reduced.

Control of M public resolvers

Assuming an adversary capable of using either a remote code execution vulnerability in the software of a resolver or a malicious service provider hosting a fraction of resolver, M entries of the key representation are known. To analyze the effects of such an attack on the security of *Neuralyzer*, we distinguish two cases two cases (for the following we assume a maximum number of resolvers used in the portrayal: each domain in the key portrayal is organized by exactly one resolver):

In the best-case (bc) scenario (user's perspective), the number of different key bits affected by the attack is minimal, i. e., for the portrayal C_{IxN} the number of rows I is minimal while the number of columns N is maximal. For a key length I and portrayal size N, the probability of being attacked in a best-case distribution of M controlled domains is as follows:

$$Pr(bc) = \binom{I \cdot N}{M} \tag{7.16}$$

Opposed to the best-case scenario, the adversary has the maximum possible knowledge about the encryption key in case the controlled resolvers are distributed over a maximum number of rows I in the portrayal. We simulate this attack for different fractions m of controlled resolvers and varying key lengths I and portrayal sizes N. Results show that even for small m the adversary can gather a significant amount of information about the encryption key. For $m = 0.05$ an adversary can control from 14% (at $I = 128$, $N = 3$) up to 40% (at $I = 128$, $N = 10$) of rows in the key portrayal.

7.7.2. Further Attacks

In addition to the above attacks and security issues, we further discuss the threat of brute-force and Sybil attacks.

Brute-force. A brute-force attack on all possible DNS cache resolvers requires guessing all potential destinations for bits of the key share. As summarized in EphPub [CDCFK11], 126 million possible entries (2011) render the use of brute-force an unlikely option for successful key recovery for a

DNS implementation. Moreover, the recovery of single key bits still requires the reconstruction of the full encryption key and matching of this key to the specific object it was applied to.

Sybil and infiltration. The Sybil attack, such as used by Zeng et al. [ZSXF10] for showing vulnerabilities in the Vanish [GKLL09] system, is realized by controlling multiple virtual identities within the target infrastructure. The adversary then can monitor and analyze a large portion of the traffic occurring in a network. In the context of a DNS-based implementation, a Sybil cannot be realized through virtual nodes as there is no such thing as virtual identities for DNS resolvers. The overhead for controlling the required amount of physical nodes in the DNS system makes a successful Sybil or infiltration attack unlikely.

7.8. Conclusion

In this work, we proposed a novel approach for flexible revocation of online data. Other than recent work in this field, our central goal is to provide a metric-driven revocation mechanism that adapts to, e. g., the progression of user accesses over time instead of a predefined lifetime. To this end, we protect the publicly accessible data through encryption, encapsulate the encrypted content, and ensure the destruction of the encryption key when its lifetime expires. To assess our approach, we created *Neuralyzer*, a proof-of-concept system based on the caching mechanism of public DNS resolvers that can refresh key information over time and expire its lifetime based on access heuristics. Results of a simulation study and experiments with a prototype implementation reveal that we can achieve flexible and reliable expiration times for the revocation of online data based on users' interest.

With hindsight I was more than blind.

— Placebo

8
Closing Remarks

Contents

8.1. Key Findings and Future Work

In this work, we have acquired a series of experimental results that serve to answer the initial research questions. In the following, we derive the key findings for each of these research questions and suggest possible directions for future work.

8.1.1. Analyzing Attacks and Countermeasures

In Chapter 3, we addressed the challenges that arise from the diversity of evaluation techniques for traffic analysis attacks. With the goal of minimizing the prevailing discrepancy between scientific work and real-world requirements, we formulated the following research question.

Research Question 1 *How can we overcome the scientific diversity in existing attacks to facilitate the development of new countermeasure techniques?*

Even though prior work introduced a high number of attack concepts, the diversity of their evaluation approaches makes it impossible to compare their performance. This leads to a situation in which we know several open attack vectors in Tor and options to exploit them, but the actual impact of the state of the art remains unclear. As a consequence, we lack a reference for existing *and* new attack concepts. While large-scale adversaries move once only theoretical worst-case adversaries closer to reality, also the technical capabilities of an adversary improve over time. Only by comparing the extent of current threats we will be able to assess the attack landscape with a long-term perspective.

Likewise, the development of new countermeasures depends on the comparison of existing attacks. Although testing new defenses against *individual* attack concepts delivers insights into the feasibility of a new countermeasure concept, such experiments can only give a local performance assessment. However, to create a realistic countermeasure, we must test new defenses against the full state of the art. The current lack of attack performance benchmarks leads to a situation in which we find only a few to none deployed countermeasures against the traffic analysis attacks. With the emerg-

ing threat of deep learning attacks, we must improve the development cycle for new countermeasures for accelerating the defensive side in the arms race.

Key Findings

Motivated by the above challenges, we derive the following key findings from our experiments with *DigesTor*.

- Combining the implementation of attacks and the experimental network setup in one framework allows for evaluating the current state of the art. Furthermore, it allows for extending the configuration with future work and makes long-term performance benchmarks possible.
- Only with this comparability we can put the current attack landscape in a realistic context and estimate its real-world threat. In particular, we cover the *technical* capabilities of traffic analysis attacks and estimate the *operational* characteristics through the definition of an attacker model.
- For the first time, the attack performance benchmarks made it possible to evaluate new countermeasures against the state of the art. Mixing is only one of many possible traffic obfuscation methods, and the attack benchmarks help to test which of these methods and parameter setups provide reasonable security and performance trade off.

Future Work

Based on the key findings of our work, we define possible directions for future work with a focus on defensive research. Simple obfuscation techniques, e. g., mixing, help to hinder the success of current attacks on the short-term and directly interfere with the technical capabilities of end-to-end confirmation. Nevertheless, they are unlikely to counter deep learning attacks, which would require to bypass the automated feature engineering without exceeding existing technical limits. Furthermore, the performance overhead of pure traffic obfuscation makes it a pointless security feature for future systems. We consider *efficient* obfuscation a direct defense that can only provide security with a short-term perspective.

From a long-term perspective, we can take advantage of improving network properties. Overlay networks like Tor depend on the transmission capabilities of the underlying network and the voluntarily operated relays. Increasing transmission capacities might allow using obfuscation techniques, e. g., the injection of cover traffic, without impairing the network performance. We can draw similar advantages from improving computing power and, for example, establish efficient encryption for network-layer anonymity systems.

Overall, the real-world requirements of deployed systems always restrict defensive work—a fact that does not apply for attacks. This unbalanced situation predicts a persisting advantage for offensive work and challenges absolute security. However, the broad range of use cases does not always justify the limitations that would be required for perfect protection. Future systems with options for different degrees of security help to achieve an efficient yet realistic status.

8.1.2. Alternative Defenses

In Chapter 4, we introduced geographical avoidance as an alternative to expensive traffic obfuscation. With the goal of developing an alternative defense, we defined the following research question.

Research Question 2 *Do alternative countermeasures find a better compromise in the security and performance trade off for a long-term defense?*

Obfuscating traffic is expensive in a sense that it increases the transmission delays (mixing) or occupies the available network resources (cover traffic). Circumventing the reach of large-scale adversaries like nation-states is an alternative way to evade the threat of traffic analysis attacks. However, the complex real-world network characteristics of Tor and the underlying network challenge the design of an avoidance system and lead to a complex problem statement. We face problems like untrusted location information or unbalanced network resources. Furthermore, we experience an essential connection between performance and security features with consequences for Tor's network resources *and* the size of the anonymity set.

Circumventing untrusted areas offers a new perspective for countering the persisting threat of traffic analysis attacks. While such alternative defenses offer short-term protection in the current situation, they also suggest possible security mechanisms for the development of future systems.

Key Findings

Following the above problem statement, we derive the following key findings from our experiments with *TrilateraTor*.

- Avoidance systems highly depend on the network infrastructure of Tor and the underlying network. Realistic approaches must measure and understand such network characteristics, as ignoring them can open new attack vectors.

- Performance impairments through avoidance decisions directly relate to possible security issues. One example of this is the minimization of the anonymity set size because of circumventing countries that provide a majority of Tor relays.

- The deployment of a system adds further restrictions to the problem statement. Avoidance systems can only be a realistic security option if these deployment requirements are covered.

Future Work

The above key findings help us to suggest directions for future work. In the following, we focus on a more comprehensive consideration of large-scale adversaries and add the user's perspectives as one crucial input for the deployment of an avoidance system.

AS-level adversaries can access transmission information at critical network nodes and cover broad areas of networks. Furthermore, centralized network infrastructures increase the reach of AS-level adversaries further. To take this into account, we need to include AS regions in the decision process. While this increases the complexity of avoidance decisions, we also depend on another reliable source of information for the areas of different ASes.

In contrast to most nation-state adversaries, AS regions do not have to be coherent, but can span multiple different places. These characteristics add several new challenges to an AS-aware avoidance system.

While *TrilateraTor* already considered several technical requirements for real-world deployment, we must also obtain the user's perspective to design a realistic security feature. One aspect of this is an integration into the current Tor revision, where we need to define how or if users can interfere with the decision mechanism. One example of this is the definition of a decision threshold that allows focusing on either security- or performance-centric decisions. Although user decisions offer individual flexibility, they also increase the decision-making burden. Another aspect is handling collective avoidance decisions that can damage the performance of Tor and minimize the anonymity set. Such collective decisions are a likely scenario, as we can assume that, e. g., a country's decision against net neutrality, motivates many users to circumvent this area. Future systems must be able to handle a critical mass of avoidance decisions to protect Tor's network resources. Furthermore, features like improved load balancing mechanisms can limit such effects.

8.1.3. Traffic Fingerprinting on Mobile Networks

In Chapter 5, we transferred well-known traffic analysis attacks to the context of mobile networks. With the specific radio layer adversary in mind, we analyzed the impact of traffic fingerprinting attacks with respect to the following research question.

Research Question 3 *Can we transfer well-known attack techniques to the context of mobile networks to emphasize the threat of traffic analysis?*

During the last decade, the mobile Internet became increasingly important, and its development faces the same restrictions through the security and performance trade off we already know from the context of Tor. Because Tor's performance-oriented design decisions entailed the persistent attack vector of traffic analysis attacks, we can expect similar vulnerabilities in mobile networks as well. At this point, we benefit from the numerous iterations that have already gone through attacks in the context of Tor.

Successfully transferring state of the art attacks to the current mobile generation (LTE) induces severe security issues, but also future mobile generations that tie in with the current standard continue open attack vectors. As long as we must subordinate security features to real-world performance requirements, well-known side channels persist across different network technologies.

Another critical factor for the impact of traffic fingerprinting on mobile networks is the wireless radio layer adversary. In contrast to conventional attacks on Tor, the radio layer adversary monitors traffic within the cell of a provider and does not depend on any operational challenging network coverage. This puts the concepts of a large-scale adversary in a new context and leads to a higher attack impact with fewer resources.

Key Findings

In this context, we derive the following key findings from our experiments.

- LTE is subject to the same restrictions as before Tor and must prefer performance-oriented design decisions that create a side channel of transmission meta data on the radio layer. Therefore, a radio layer adversary can conduct state of the art attacks that we already know from the context of Tor.

- Future mobile generations continue similar design decisions and we must expect the vulnerabilities to persist.

- Additional attack vectors across other layers of the protocol stack can be combined and amplify the effects of single attacks.

Future Work

Based on the above key findings, we define possible directions for future work. We differentiate the technical and operational aspects of traffic fingerprinting attacks on mobile networks.

The current development of attacks on Tor suggests the use of deep learning attacks in which automated feature engineering helps to create robust

machine learning classifiers. Although the specific traffic features differ between the network technologies, we can assume that deep learning attacks also succeed in the context of LTE. From a research perspective, we then face the challenge of generating large data sets with only limited overhead. If we can overcome this limitation, state of the art deep learning attacks should also be possible on LTE radio layer traffic. Another approach can be a cross-layer combination of attacks. One example of this is using the information leak of one layer for conducting a targeted attack on another layer that increases the overall impact.

From an organizational perspective, adversaries can use traffic fingerprinting for creating comprehensive profiles of users or tracking their whereabouts. The impact of such privacy attacks is amplified through the combination of multiple cells, e. g., adversaries with sensors in different radio cells can derive a large amount of sensitive information and increase their knowledge over time.

8.1.4. Controlling Online Data

In Chapter 7, we switch to a new context for analyzing options that offer users control over their online data. With the goal of minimizing the decision-making burden for users, we defined the following research question.

Research Question 4 *How can we offer users tools to control their online data in the presence of large service providers?*

Even though many online services offer options to delete once uploaded data, their external servers provide no transparency, and we are unable to follow up what happens. Such missing transparency requires trust in big companies like Google or Facebook and prevents us from retaining control over private information. Revocation systems offer us such an option, but they often also burden users with preliminary decisions about the, e. g., lifetime of data. If we can overcome this burden of keeping track of all the information we share on the Internet, we can share data knowing we do not leave persisting traces.

Technical solutions supporting the *right to erasure* not only overcome the above challenges, but they also offer a new perspective on how to handle

online data. More precisely, usable revocation systems provide new ways to protect private information and take away some of the capabilities of large-scale service providers.

Key Findings

Following the above problem statement, we derive the following key findings from our experiments with *Neuralyzer*.

- User-driven revocation schemes minimize the decision-making burden of the user and help to handle the large amounts of information we share on the Internet.
- Distributed infrastructures help to establish an independent key storage as central security mechanism of a revocation system. Nevertheless, referring to one single system can exhaust its capacity.

Future Work

One direction for future work is the identification of alternative infrastructures for the ephemeral key storage. The combination of multiple architectures can help to create a well-scaling system and increase the fault tolerance, but also increases the organizational overhead for managing user accesses and keys. Another direction is the better integration of user perspectives for the design of new systems. *Neuralyzer* provides an example of how access heuristics help to improve the flexibility of a revocation scheme, nevertheless, missing user studies leave us uncertain about other further requirements for a realistic adoption of such a system. As digital forgetting is a user-centric research area, their perspective is inevitable for the next steps.

8.2. Conclusion

Privacy and anonymity are growing concerns in the presence of governments monitoring user behavior or large companies tracking visits across websites. Since the democratization of the Internet, adversaries became increasingly

powerful, and we experience the emerging threat of large-scale adversaries that approach once only theoretical worst-case scenarios. While a broad spectrum of what is considered ethical leads to controversial discussions of how much privacy the Internet should offer, scientific work is an essential accelerator for an unbiased investigation of new possibilities.

However, scientific concepts can expand beyond the possibilities of a real-world system and create sophisticated approaches that cannot always fit the requirements of reality. This creates a gap between the solutions of the academic world and the actual problem statements of the real world. Such a gap allows science to continue, while some real-world issues remain unresolved. Overcoming this discrepancy between both worlds is an essential factor to maintain their connection.

In this thesis, we presented approaches for minimizing the gap between the scientific and the real world. With a focus on the Tor anonymity system, we addressed the persisting threat of traffic analysis attacks that benefit from strong large-scale adversaries. Our work helped to overcome the diversity of experimental evaluation techniques for analyzing possible countermeasures in the presence of state of the art attacks for the first time. While the presented countermeasure helped to limit the success of passive traffic analysis in the short term, we must also identify and propose solutions that help to advance long-term defensive methods. Therefore, we designed an alternative countermeasure that circumvents the threat of nation-state adversaries instead of referring to expensive traffic obfuscation. The corresponding experiments helped to understand that only considering *all* dimensions of a problem statement allows for creating the desired long-term solutions to existing problems. In addition to the well-researched context of Tor, we transferred state of the art traffic fingerprinting attacks to the increasingly established mobile networks. A series of experiments demonstrated that we experience the same critical information leaks as before in the context of Tor. Nevertheless, the specific capabilities of a radio layer attacker add a new perspective to conducting the attack and its impact, respectively. In a final step, we aim for providing users control over online data in the presence of large service providers. In this context, we presented a concept that enforces the right

to erasure for offering new perspectives on how users might experience and manage their online traces.

All results presented in this thesis follow the goal of offering new *perspectives* in situations where powerful large-scale adversaries seem to dominate. Such perspectives help to change the perception of what only appears as a hypothetical right and what can be assumed a real option. Creating such perspectives is only possible if we overcome the discrepancies between the results of scientific freedom and the requirements of the real world. Although this compromise dictates some limitations, it is the condition for forming future systems with even starting positions in the security arms race.

List of Figures

List of Tables

List of Abbreviations

BGP	Border Gateway Protocol
CP-ABE	Cipher-Text-Policy Attribute-Based Encryption
C-RNTI	Cell Radio Network Temporary Identifier
DCI	Downlink Control Information
DHT	Distributed Hash Tables
DNS	Domain Name System
EDO	Ephemeral Data Object
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
GDPR	General Data Protection Regulation
GUTI	Globally Unique Temporary Identifier
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IMSI	International Mobile Subscriber Identity
IXP	Internet Exchange Point
k-NN	k-Nearest Neighbor
LTE	Long Term Evolution
MAC	Medium Access Control
NAS	Non-Access Stratum
NN	Neural Network

P2P . . . . Peer-to-Peer

PDCP . . . Packet Data Convergence Protocol

RAP . . . . Random Access Procedure

RAR . . . . Random Access Response

RLC . . . . Radio Link Control

RNTI . . . . Radio Network Temporary Identifier

RTT . . . . Round-Trip Time

SDR . . . . Software Defined Radio

SVM . . . . Support Vector Machines

TMSI . . . . Temporary Mobile Subscriber Identity

TTL . . . . Time to Live

UE . . . . . User Equipment

Bibliography

[3GP09a] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification. TR TR36.321, 3rd Generation Partnership Project (3GPP), 2009.

[3GP09b] 3GPP. 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), 2009.

[AA17] Judith Aldridge and Rebecca Askew. Delivery Dilemmas: How Drug Cryptomarket Users Identify and Seek to Reduce their Risk of Detection by Law Enforcement. *International Journal of Drug Policy*, 41:101–109, 2017.

[ABF+08] David G. Andersen, Hari Balakrishnan, Nick Feamster, Teemu Koponen, Daekyeong Moon, and Scott Shenker. Accountable Internet Protocol (AIP). In *Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '08, pages 339–350, Seattle, WA, USA, August 2008. ACM.

[ADS03] Alessandro Acquisti, Roger Dingledine, and Paul Syverson. On the Economics of Anonymity. In *Financial Cryptography*, FC '03, pages 84–102, Guadeloupe, French West Indies, January 2003. Springer.

[AG16] Kota Abe and Shigeki Goto. Fingerprinting Attack on Tor Anonymity using Deep Learning. *Proceedings of the Asia-Pacific Advanced Network*, 42:15–20, 2016.

[ASS15] Farhan M. Aziz, Jeff S. Shamma, and Gordon L. Stüber. Resilience of LTE Networks Against Smart Jamming Attacks: Wideband Model. In *Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, PIMRC '15, pages 1344–1348, Hong Kong, China, August 2015. IEEE.

[AT13] Oshrat Ayalon and Eran Toch. Retrospective Privacy: Managing Longitudinal Privacy in Online Social Networks. In *Symposium on Usable Privacy and Security*, SOUPS '13, Newcastle, UK, July 2013. ACM.

[AT17] Oshrat Ayalon and Eran Toch. Not Even Past: Information Aging and Temporal Privacy in Online Social Networks. *Human–Computer Interaction*, 32(2):73–102, 2017.

[AW19] Yawning Angel and Philipp Winter. obfs4, January 2019. `https://github.com/Yawning/obfs4`.

[AYM12] Masoud Akhoondi, Curtis Yu, and Harsha V. Madhyastha. LASTor: A Low-Latency AS-Aware Tor Client. In *IEEE Symposium on Security and Privacy*, SP '12, pages 476–490, San Francisco, CA, USA, May 2012. IEEE.

[BBB+13] Matt Bishop, Emily Rine Butler, Kevin Butler, Carrie Gates, and Steven Greenspan. Forgive and Forget: Return to Obscurity. In *ACM Workshop on New Security Paradigms*, NSPW '13, Alberta, Canada, September 2013. ACM.

[BBD+11] Julian Backes, Michael Backes, Markus Dürmuth, Sebastian Gerling, and Stefan Lorenz. X-Pire!-A Digital Expiration Date for Images in Social Networks. *arXiv preprint arXiv:1112.2649*, 2011.

[BCK+13] Lujo Bauer, Lorrie Faith Cranor, Saranga Komanduri, Michelle L. Mazurek, Michael K. Reiter, Manya Sleeper, and Blase Ur. The Post Anachronism: The Temporal Dimension of Facebook Privacy. In *Workshop on Privacy in the Electronic Society*, WPES '13, pages 1–12. ACM, 2013.

[BFvT02] Mario Blaum, Patrick G. Farrell, and Henk C.A. van Tilborg. *Information, Coding and Mathematics*. The Springer International Series in Engineering and Computer Science, 2002.

[BFZ07] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A Study of Prefix Hijacking and Interception in the Internet. August 2007.

[BGS19] Adam Back, Ian Goldberg, and Adam Shostack. Freedom Systems 2.1 Security Issues and Analysis. White Paper, Zero Knowledge Systems, January 2019.

[BHKL13] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: Elliptic-Curve Points Indistinguishable from Uniform Random Strings. In *ACM Conference on Computer and Communications Security*, CCS '13, pages 967–980, Berlin, Germany, November 2013. ACM.

[BKM+13] Michael Backes, Aniket Kate, Praveen Manoharan, Sebastian Meiser, and Esfandiar Mohammadi. AnoA: A Framework for Analyzing Anonymous Communication Protocols. In *Computer Security Foundations Symposium*, CSF '13, pages 163–178, Hoboken, NJ, USA, June 2013. IEEE.

[BKMM14] Michael Backes, Aniket Kate, Sebastian Meiser, and Esfandiar Mohammadi. (Nothing else) MATor(s): Monitoring the Anonymity of Tor's Path Selection. In *ACM Conference on Computer and Communications Security*, CCS '14, pages 513–524, Scottsdale, AZ, USA, November 2014. ACM.

[BL96] Dan Boneh and Richard Lipton. A Revocable Backup System. In *USENIX Security Symposium*, USENIX '96, San Diego, CA, USA, January 1996. USENIX Association.

[BL02] Oliver Berthold and Heinrich Langos. Dummy Traffic Against Long Term Intersection Attacks. In *Workshop on Privacy Enhancing Technologies*, PET '02, pages 110–128, San Francisco, CA, USA, April 2002. Springer.

[BLJL05] George Dean Bissias, Marc Liberatore, David Jensen, and Brian Neil Levine. Privacy Vulnerabilities in Encrypted HTTP

Streams. In *Workshop on Privacy Enhancing Technologies*, PET '05, pages 1–11, Cavtat, Croatia, May 2005. Springer.

[BMG+07] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-Resource Routing Attacks Against Tor. In *Workshop on Privacy in the Electronic Society*, WPES '07, pages 11–20, Alexandria, VA, USA, October 2007. ACM.

[BMM14] Michael Backes, Praveen Manoharan, and Esfandiar Mohammadi. Tuc: Time-Sensitive and Modular Analysis of Anonymous Communication. In *Computer Security Foundations Symposium*, CSF '14, pages 383–397, Washington, DC, USA, July 2014. IEEE.

[BPW13] Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization. In *IEEE Symposium on Security and Privacy*, SP '13, pages 80–94, San Francisco, CA, USA, May 2013. IEEE.

[BR14] Nihal Balani and Sushmita Ruj. Temporal Access Control with User Revocation for Cloud Data. In *International Conference on Trust, Security and Privacy in Computing and Communications*, TrustCom '14, Beijing, China, September 2014. IEEE.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-Policy Attribute-Based Encryption. In *IEEE Symposium on Security and Privacy*, SP '07, Oakland, CA, USA, May 2007. IEEE.

[Bui17] Nicola Bui. IMDEA's Online Watcher for LTE (OWL) control channel, January 2017. `https://git.networks.imdea.org/nicola_bui/imdeaowl`.

[BW16a] Armon Barton and Matthew Wright. DeNASA: Destination-Naive AS-Awareness in Anonymous Communications. volume 2016 of *PoPETS '16*, pages 356–372. De Gruyter, October 2016.

[BW16b] Nicola Bui and Joerg Widmer. OWL: A Reliable Online Watcher for LTE Control Channel Measurements. In *Workshop on All Things Cellular: Operations, Applications and Challenges*, ATC '16, pages 25–30, New York, USA, October 2016. ACM.

[CA98] Heyning Cheng and Ron Avnur. Traffic Analysis of SSL Encrypted Web Browsing, 1998.

[CAB+15] Chen Chen, Daniele E. Asoni, David Barrera, George Danezis, and Adrain Perrig. HORNET: High-Speed Onion Routing at the Network Layer. In *ACM Conference on Computer and Communications Security*, CCS '15, pages 1441–1454, Denver, CO, USA, October 2015. ACM.

[CBP+14] Sambuddho Chakravarty, Marco V. Barbera, Georgios Portokalidis, Michalis Polychronakis, and Angelos D. Keromytis. On the Effectiveness of Traffic Analysis Against Anonymity Networks Using Flow Records. In *International Conference on Passive and Active Measurement*, PAM '14, pages 247–257, Los Angeles, CA, USA, March 2014. Springer.

[CDCFK11] Claude Castelluccia, Emiliano De Cristofaro, Aurélien Francillon, and M.-A. Kaafar. EphPub: Toward Robust Ephemeral Publishing. In *IEEE International Conference on Network Protocols*, ICNP '11, Vancouver, Canada, October 2011. IEEE.

[CGF10] Henry Corrigan-Gibbs and Bryan Ford. Dissent: Accountable Anonymous Group Messaging. In *ACM Conference on Computer and Communications Security*, CCS '10, Chicago, IL, USA, October 2010. ACM.

[Cha81] David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2), 1981.

[Che17] Giovanni Cherubin. Bayes, not Naïve: Security Bounds on Website Fingerprinting Defenses. volume 2017 of *PoPETS '17*, pages 215–231. De Gruyter, 2017.

[CL05] Jan Camenisch and Anna Lysyanskaya. A Formal Treatment of Onion Routing. In *Annual International Cryptology Conference*, CRYPTO '05, pages 169–187, Santa Barbara, CA, USA, August 2005. Springer.

[Cla99] Roger Clarke. Introduction to Dataveillance and Information Privacy, and Definitions of Terms. *Roger Clarke's Dataveillance and Information Privacy Pages*, 1999.

[Cla06] Roger Clarke. What's Privacy? *Roger Clarke's What's Privacy Pages*, 2006.

[CMU15] CMU BlackHat Submission. You Don't Have to be the NSA to Break Tor, January 2015. `https://web.archive.org/web/20140705114447/http://blackhat.com/us-14/briefings.html#you-dont-have-to-be-the-nsa-to-break-tor-deanonymizing-users-on-a-budget`.

[CNJ14] Xiang Cai, Rishab Nithyanand, and Rob Johnson. CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense. In *Workshop on Privacy in the Electronic Society*, WPES '14, pages 121–130, Scotsdale, AZ, USA, 2014. ACM.

[CNW+14] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In *ACM Conference on Computer and Communications Security*, CCS '14, pages 227–238, Scottsdale, AZ, USA, November 2014. ACM.

[Con10] Chris Conley. The Right to Delete. In *AAAI Spring Symposium: Intelligent Information Privacy Management*, AAAI '10, Palo Alto, CA, USA, March 2010.

[Cot19] Lance Cottrell. Mixmaster 2.0 remailer release, January 2019. `http://mixmaster.sourceforge.net/faq.shtml`.

[Cro19] Crowd Supply. LimeSDR Mini, January 2019. `https://www.crowdsupply.com/lime-micro/limesdr-mini`.

[CSMK15] Jason W. Clark, Peter Snyder, Damon McCoy, and Chris Kanich. "I Saw Images I Didn't Even Know I Had": Understanding User Perceptions of Cloud Storage Privacy. In *Conference on Human Factors in Computing Systems*, CHI '15, pages 1641–1644, Seoul, Republic of Korea, April 2015. ACM.

[CW17] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy*, SP '17, pages 39–57, San Jose, CA, USA, May 2017. IEEE.

[CZJJ12] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a Distance: Website Fingerprinting Attacks and Defenses. In *ACM Conference on Computer and Communications Security*, pages 605–616, Raleigh, NC, USA, October 2012. ACM.

[Dan03] George Danezis. Statistical Disclosure Attacks. In *International Conference on Information Security*, SEC '03, pages 421–426, Athens, Greece, May 2003. Springer.

[DCRS12] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *IEEE Symposium on Security and Privacy*, SP '12, pages 332–346, San Francisco, CA, USA, May 2012. IEEE.

[DCRS13] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. Protocol Misidentification Made Easy with Format-Transforming Encryption. In *ACM Conference on Computer and Communications Security*, CCS '13, pages 61–72, Berlin, Germany, November 2013. ACM.

[DCSTW12] Emiliano De Cristofaro, Claudio Soriente, Gene Tsudik, and Albert Williams. Hummingbird: Privacy at the Time of Twitter. In *IEEE Symposium on Security and Privacy*, SP '12, San Francisco, CA, USA, May 2012. IEEE.

[DDM03] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *IEEE Symposium on Security and Privacy*, SP '03, pages 2–15, Oakland, CA, USA, May 2003. IEEE.

[DDT07] George Danezis, Claudia Diaz, and Carmela Troncoso. Two-Sided Statistical Disclosure Attack. In *Workshop on Privacy Enhancing Technologies*, PET '07, pages 30–44, Ottawa, ON, Canada, June 2007. Springer.

[DDTL10] George Danezis, Claudia Diaz, Carmela Troncoso, and Ben Laurie. Drac: An Architecture for Anonymous Low-Volume Communications. In *Privacy Enhancing Technologies Symposium*, PETS '10, pages 202–219, Berlin, Germany, July 2010. Springer.

[DG09] George Danezis and Ian Goldberg. Sphinx: A Compact and Provably Secure Mix Format. In *IEEE Symposium on Security and Privacy*, SP '09, pages 269–282, Oakland, CA, USA, May 2009. IEEE.

[DL04] George Danezis and Ben Laurie. Minx: A Simple and Efficient Anonymous Packet Format. In *Workshop on Privacy in the Electronic Society*, WPES '04, pages 59–65. ACM, October 2004.

[DMMK18] Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. Anonymity Trilemma: Strong Anonymity, Low Bandwidth Overhead, Low Latency-Choose Two. In *IEEE Symposium on Security and Privacy*, SP '18, pages 108–126, San Francisco, CA, USA, May 2018. IEEE.

[DP04] Claudia Diaz and Bart Preneel. Taxonomy of Mixes and Dummy Traffic. In *Information Security Management, Education and Privacy*, pages 217–232, Toulouse, France, August 2004. Springer.

[DSA+11] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. Analysis of Country-Wide Internet Outages Caused by Censorship. In *ACM SIGCOMM Conference on Internet Measurement*, IMC '11, pages 1–18, Berlin, Germany, November 2011. ACM.

[DSS06] Roger Dingledine, Andrei Serjantov, and Paul Syverson. Blending Different Latency Traffic with Alpha-mixing. In *Workshop on Privacy Enhancing Technologies*, PET '06, pages 245–257, Cambridge, UK, June 2006. Springer.

[EBA+12] Tariq Elahi, Kevin Bauer, Mashael AlSabah, Roger Dingledine, and Ian Goldberg. Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor. In *Workshop on Privacy in the Electronic Society*, WPES '12, Raleigh, NC, USA, October 2012. ACM.

[EC12] Tameem Eissa and Gi-Hwan Cho. A Fine Grained Access Control and Flexible Revocation Scheme for Data Security on Public Cloud Storage Services. In *International Conference on Cloud Computing Technologies, Applications and Management*, ICCCTAM '12, Dubai, UAE, December 2012.

[EEFR16] Anne Edmundson, Roya Ensafi, Nick Feamster, and Jennifer Rexford. Characterizing and Avoiding Routing Detours

Through Surveillance States. *CoRR*, abs/1605.07685, May 2016.

[EEFR18] Anne Edmundson, Roya Ensafi, Nick Feamster, and Jennifer Rexford. Nation-State Hegemony in Internet Routing. In *ACM SIGCAS Conference on Computing and Sustainable Societies*, COMPASS '18, pages 17:1–17:11, Menlo Park and San Jose, CA, USA, June 2018. ACM.

[EJM+15] Daniel Ellard, Christine Jones, Victoria Manfredi, W. Timothy Strayer, Bishal Thapa, Megan Van Welie, and Alden Jackson. Rebound: Decoy Routing on Asymmetric Routes Via Error Messages. In *IEEE Conference on Local Computer Networks*, LCN '15, pages 91–99, Clearwater Beach, FL, USA, October 2015. IEEE.

[ES09] Matthew Edman and Paul Syverson. AS-Awareness in Tor Path Selection. In *ACM Conference on Computer and Communications Security*, CCS '09, pages 380–389, Chicago, IL, USA, November 2009. ACM.

[Eur19] European Commission. Factsheet on the "Right to Be Forgotten" Ruling, C-131/12, January 2019. `http://ec.europa.eu/justice/data-protection/files/factsheets/factsheet_data_protection_en.pdf`.

[EY09] Matthew Edman and Bülent Yener. On Anonymity in an Electronic Society: A Survey of Anonymous Communication Systems. *ACM Computing Surveys (CSUR)*, 42(1):5, 2009.

[fgs18] fgsect. SCAT: Signaling Collection and Analysis Tool, January 2018. `https://github.com/fgsect/scat`.

[Fif19] David Fifield. Meek, January 2019. `https://trac.torproject.org/projects/tor/wiki/doc/meek`.

[FJS07] Joan Feigenbaum, Aaron Johnson, and Paul Syverson. A Model of Onion Routing with Provable Anonymity. In *International Conference on Financial Cryptography and Data Security*, FC '07, pages 57–71, Lowlands, Scarborough, Trinidad/Tobago, February 2007. Springer.

[FJS12] Joan Feigenbaum, Aaron Johnson, and Paul Syverson. Probabilistic Analysis of Onion Routing in a Black-Box Model. *Transactions on Information and System Security (TISSEC)*, 15(3):14, 2012.

[FLH+15] David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. Blocking-Resistant Communication through Domain Fronting. volume 2015 of *PoPETS '15*, pages 46–64. De Gruyter, 2015.

[FLL+09] Xinwen Fu, Zhen Ling, J Luo, W Yu, W Jia, and W Zhao. One Cell is Enough to Break Tor's Anonymity. In *Proceedings of Black Hat Technical Security Conference*, BLACKHAT '09, pages 578–589, Las Vegas, NV, USA, July 2009.

[FM02] Michael J. Freedman and Robert Morris. Tarzan: A Peer-to-Peer Anonymizing Network Layer. In *ACM Conference on Computer and Communications Security*, CCS '02, pages 193–206, Washington, DC, USA, November 2002. ACM.

[GA05] Ralph Gross and Alessandro Acquisti. Information Revelation and Privacy in Online Social Networks. In *Workshop on Privacy in the Electronic Society*, WPES '05, Alexandria, VA, USA, November 2005. ACM.

[GDP19] GDPR. Art. 17 GDPR: Right to erasure, January 2019. `https://gdpr-info.eu/art-17-gdpr/`.

[GH12] Yossi Gilad and Amir Herzberg. Spying in the Dark: TCP and Tor Traffic Analysis. In *Privacy Enhancing Technologies*

Symposium, PETS '12, pages 100–119, Vigo, Spain, July 2012. Springer.

[GH13] Nethanel Gelernter and Amir Herzberg. On the Limits of Provable Anonymity. In *Workshop on Privacy in the Electronic Society*, WPES '13, pages 225–236. ACM, 2013.

[GKK+11] Roxana Geambasu, Tadayoshi Kohno, Arvind Krishnamurthy, Amit Levy, Henry Levy, Paul Gardner, and Vinnie Moscaritolo. New Directions for Self-Destructing Data Systems. Technical report, University of Washington, 2011.

[GKLL09] Roxana Geambasu, Tadayoshi Kohno, Amit A Levy, and Henry M Levy. Vanish: Increasing Data Privacy with Self-Destructing Data. In *USENIX Security Symposium*, USENIX '09, San Diego, CA, USA, June 2009. USENIX Association.

[GRPS03] Sharad Goel, Mark Robson, Milo Polte, and Emin Sirer. Herbivore: A Scalable and Efficient Protocol for Anonymous Communication. Technical report, Cornell University, 2003.

[GRS96] David M Goldschlag, Michael G Reed, and Paul F Syverson. Hiding Routing Information. In *Inernational Workshop on Information Hiding*, IH '96, pages 137–150, Cambridge, UK, May 1996. Springer.

[GSH+17] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. A Look at Router Geolocation in Public and Commercial Databases. In *ACM SIGCOMM Conference on Internet Measurement*, IMC '17, pages 463–469, London, UK, November 2017. ACM.

[GT96] Ceki Gulcu and Gene Tsudik. Mixing E-mail with Babel. In *Network and Distributed System Security Symposium*, NDSS '96, pages 2–16, San Diego, CA, USA, February 1996. The Internet Society.

[Gua19] Guardian Project. Orbot: Tor for Android, January 2019. `https://guardianproject.info/apps/orbot/`.

[GZCF06] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. Constraint-Based Geolocation of Internet Hosts. *IEEE/ACM Transactions on Networking (TON)*, 14(6):1219–1232, 2006.

[GZPH16] Manaf Gharaibeh, Han Zhang, Christos Papadopoulos, and John Heidemann. Assessing Co-Locality of IP Blocks. In *Computer Communications Workshops*, INFOCOM WKSHPS '16, pages 503–508, Atlanta, GA, USA, May 2016. IEEE.

[HB11] Amir Houmansadr and Nikita Borisov. SWIRL: A Scalable Watermark to Detect Correlated Network Flows. In *Network and Distributed System Security Symposium*, NDSS '11, San Diego, CA, USA, February 2011. The Internet Society.

[HB13] Amir Houmansadr and Nikita Borisov. The need for Flow Fingerprints to Link Correlated Network Flows. In *Privacy Enhancing Technologies Symposium*, PETS '13, pages 205–224, Bloomington, IN, USA, July 2013. Springer.

[HBK18] Byeongdo Hong, Sangwook Bae, and Yongdae Kim. GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier. In *Network and Distributed System Security Symposium*, NDSS '18, San Diego, CA, USA, February 2018. The Internet Society.

[HBS13] Amir Houmansadr, Chad Brubaker, and Vitaly Shmatikov. The Parrot is Dead: Observing Unobservable Network Communications. In *IEEE Symposium on Security and Privacy*, SP '13, pages 65–79, San Francisco, CA, USA, May 2013. IEEE.

[HD16] Jamie Hayes and George Danezis. k-Fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *USENIX Security Symposium*, USENIX '16, pages 1187–1203, Washington, DC, USA, August 2016. USENIX Association.

[HEC+19] Syed Rafiul Hussain, Mitziu Echeverria, Omar Chowdhury, Ninghui Li, and Elisa Bertino. Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information. February 2019.

[HF+11] Bradley Huffaker, Marina Fomenkov, et al. Geocompare: A Comparison of Public and Commercial Geolocation Databases. Technical report, Cooperative Association for Internet Data Analysis (CAIDA), 2011.

[Hin02] Andrew Hintz. Fingerprinting Websites Using Traffic Analysis. In *Workshop on Privacy Enhancing Technologies*, PET '02, pages 171–178, San Francisco, CA, USA, April 2002. Springer.

[HK06] Rob J. Hyndman and Anne B. Koehler. Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting*, 22(4):679 – 688, 2006.

[HKP+12] Hsu-Chun Hsiao, Tiffany Hyun-Jin Kim, Adrian Perrig, Akira Yamada, Samuel C Nelson, Marco Gruteser, and Wei Meng. LAP: Lightweight Anonymity and Privacy. In *IEEE Symposium on Security and Privacy*, SP '12, pages 506–520, San Francisco, CA, USA, May 2012. IEEE.

[HN11] Junbeom Hur and Dong Kun Noh. Attribute-Based Access Control With Efficient Revocation in Data Outsourcing Systems. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1214–1221, 2011.

[Huf19] Huffington Post. Experts Say Facebook Leak of 6 Million Users' Data Might Be Bigger Than We Thought, January 2019. `http://www.huffingtonpost.com/2013/06/27/facebook-leak-data_n_3510100.html`.

[HWF09] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier.

In *ACM Workshop on Cloud Computing Security*, CCSW '09, pages 31–42, Chicago, IL, USA, November 2009. ACM.

[Int19] Internet Live Stats. Total Mumber of Websites, January 2019. `http://www.internetlivestats.com/total-number-of-websites/`.

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

[JGW+14] Rob Jansen, John Geddes, Chris Wacek, Micah Sherr, and Paul Syverson. Never Been KIST: Tor's Congestion Management Blossoms with Kernel-Informed Socket Transport. In *USENIX Security Symposium*, USENIX '14, pages 127–142, San Diego, CA, USA, August 2014. USENIX Association.

[JH12] Rob Jansen and Nicholas Hopper. Shadow: Running Tor in a Box for Accurate and Efficient Experimentation. In *Network and Distributed System Security Symposium*, NDSS '12, San Diego, CA, USA, February 2012. The Internet Society.

[JJ16] Rob Jansen and Aaron Johnson. Safely Measuring Tor. In *ACM Conference on Computer and Communications Security*, CCS '16, pages 1553–1567, Vienna, Austria, October 2016. ACM.

[JJD+15] Joshua Juen, Aaron Johnson, Anupam Das, Nikita Borisov, and Matthew Caesar. Defending Tor from Network Adversaries: A Case Study of Network Path Prediction. volume 2015 of *PoPETS '15*, pages 171–187. De Gruyter, June 2015.

[JJG+17] Rob Jansen, Marc Juarez, Rafael Galvez, Tariq Elahi, and Claudia Diaz. Inside Job: Applying Traffic Analysis to Measure Tor from Within. In *Network and Distributed System Security*

Symposium, NDSS '17, San Diego, CA, USA, February 2017. The Internet Society.

[Jol02] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

[Jov13] Roger Piqueras Jover. Security Attacks Against the Availability of LTE Mobility Networks: Overview and Research Directions. In *International Symposium on Wireless Personal Multimedia Communications*, WPMC '13, Atlantic City, NJ, USA, October 2013. IEEE.

[Jov16a] Roger Piqueras Jover. LTE security and protocol exploits. Technical report, ShmooCon, January 2016.

[Jov16b] Roger Piqueras Jover. LTE security, protocol exploits and location tracking experimentation with low-cost software radio. *arXiv (1607.05171)*, 2016.

[JSM+18] Kai Jansen, Matthias Schäfer, Daniel Moser, Vincent Lenders, Christina Pöpper, and Jens Schmitt. Crowd-GPS-Sec: Leveraging Crowdsourcing to Detect and Localize GPS Spoofing Attacks. In *IEEE Symposium on Security and Privacy*, SP '18, pages 1018–1031, San Francisco, CA, USA, May 2018. IEEE.

[JWJ+13] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries. In *ACM Conference on Computer and Communications Security*, CCS '13, pages 337–348, Berlin, Germany, November 2013. ACM.

[KAL+15] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. In *USENIX Security Symposium*, USENIX '15, Washington, DC, USA, August 2015. USENIX Association.

[KAP02] Dogan Kesdogan, Dakshi Agrawal, and Stefan Penz. Limits of Anonymity in Open Environments. In *Inernational Workshop on Information Hiding*, IH '02, pages 53–69, Noordwijkerhout, The Netherlands, October 2002. Springer.

[KBJK+06] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. Towards IP Geolocation Using Delay and Topology Measurements. In *ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 71–84, Rio de Janeiro, Brazil, October 2006. ACM.

[KEB98] Dogan Kesdogan, Jan Egner, and Roland Büschkes. Stop-And-Go-MIXes Providing Probabilistic Anonymity in an Open System. In *Inernational Workshop on Information Hiding*, IH '98, pages 83–98, Portland, OR, USA, April 1998. Springer.

[KEJ+11] Josh Karlin, Daniel Ellard, Alden W. Jackson, Christine E. Jones, Greg Lauer, David P. Mankins, and W. Timothy Strayer. Decoy Routing: Toward Unblockable Internet Communication. FOCI '11, San Francisco, CA, USA, August 2011. USENIX Association.

[KFR09] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. Nation-State Routing: Censorship, Wiretapping, and BGP. *CoRR*, abs/0903.3218, 2009.

[KHB+15] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. Going Wild: Large-Scale Classification of Public DNS Resolvers. In *ACM SIGCOMM Conference on Internet Measurement*, IMC '15, Tokyo, Japan, 2015. ACM.

[KHKU18] Mohammad Taha Khan, Maria Hyun, Chris Kanich, and Blase Ur. Forgotten But Not Gone: Identifying the Need for Longitudinal Data Management in Cloud Storage. In *Conference on*

Human Factors in Computing Systems, CHI '18, pages 543:1–543:12, Montreal QC, Canada, April 2018. ACM.

[KJR+19] Katharina Kohls, Kai Jansen, David Rupprecht, Thorsten Holz, and Christina Pöpper. On the Challenges of Geographical Avoidance for Tor. In *Network and Distributed System Security Symposium*, NDSS '19, San Diego, CA, USA, February 2019. The Internet Society.

[KKHK12] Denis Foo Kune, John Koelndorfer, Nicholas Hopper, and Yongdae Kim. Location Leaks on the GSM Air Interface. In *Network and Distributed System Security Symposium*, NDSS '12, San Diego, CA, USA, 2012. The Internet Society.

[KP18] Katharina Kohls and Christina Pöpper. DigesTor: Comparing Passive Traffic Analysis Attacks on Tor. In *European Symposium on Research in Computer Security*, ESORICS '18, pages 512–530, Barcelona, Spain, September 2018. Springer.

[Kre19] Brian Krebs. Online Cheating Site AshleyMadison Hacked, January 2019. `http://krebsonsecurity.com/2015/07/online-cheating-site-ashleymadison-hacked/`.

[KRHP19] Katharina Kohls, David Rupprecht, Thorsten Holz, and Christina Pöpper. Lost Traffic Encryption: Fingerprinting LTE Traffic on Layer Two. In *Security and Privacy in Wireless and Mobile Networks*, WiSec '19. ACM, 2019.

[KSD13] Dongmin Kim, Suvrit Sra, and Inderjit S. Dhillon. A Non-Monotonic Method for Large-Scale Non-Negative Least Squares. *Optimization Methods and Software*, 28(5):1012–1039, October 2013.

[LBCC+15] Stevens Le Blond, David Choffnes, William Caldwell, Peter Druschel, and Nicholas Merritt. Herd: A Scalable, Traffic Analysis Resistant Anonymity Network for VoIP Systems. In *Confer-*

ence of the ACM Special Interest Group on Data Communication, SIGCOMM '15, pages 639–652, London, United Kingdom, 2015. ACM.

[LBCZ+13] Stevens Le Blond, David Choffnes, Wenxuan Zhou, Peter Druschel, Hitesh Ballani, and Paul Francis. Towards Efficient Traffic-analysis Resistant Anonymity Networks. In *Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '13, pages 303–314, Hong Kong, China, 2013. ACM.

[LFJ+13] Zhen Ling, Xinwen Fu, Weijia Jia, Wei Yu, Dong Xuan, and Junzhou Luo. Novel Packet Size-Based Covert Channel Attacks Against Anonymizer. In *IEEE Conference on Computer Communications*, INFOCOM '13, pages 2411–2426, Shanghai, China, April 2013. IEEE.

[LHL17] Zhihao Li, Stephen Herwig, and Dave Levin. DeTor: Provably Avoiding Geographic Regions in Tor. In *USENIX Security Symposium*, USENIX '17, pages 343–359, Vancouver, BC, Canada, August 2017. USENIX Association.

[LJL+16] Marc Lichtman, Roger Piqueras Jover, Mina Labib, Raghunandan Rao, Vuk Marojevic, and Jeffrey H. Reed. LTE/LTE-A Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation. *IEEE Communications Magazine*, 54(4):54–61, April 2016.

[LL06] Marc Liberatore and Brian Neil Levine. Inferring the Source of Encrypted HTTP Connections. In *ACM Conference on Computer and Communications Security*, CCS '06, pages 255–263, Alexandria, VA, USA, October 2006. ACM.

[LLV+15] Dave Levin, Youndo Lee, Luke Valenta, Zhihao Li, Victoria Lai, Cristian Lumezanu, Neil Spring, and Bobby Bhattacharjee. Alibi Routing. In *Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '15, pages 611–624, London, United Kingdom, August 2015. ACM.

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

[LRCN13] Marc Lichtman, Jeffrey H. Reed, T. Charles Clancy, and Mark Norton. Vulnerability of LTE to Hostile Interference. In *IEEE Global Conference on Signal and Information Processing*, GlobalSIP '13, pages 285–288, Austin, TX, USA, December 2013. IEEE.

[LRWW04] Brian N Levine, Michael K Reiter, Chenxi Wang, and Matthew Wright. Timing Attacks in Low-Latency Mix Systems. In *Financial Cryptography*, FC '04, pages 251–265, Key West, FL, USA, February 2004. Springer.

[LZC+11] Xiapu Luo, Peng Zhou, Edmond WW Chan, Wenke Lee, Rocky KC Chang, and Roberto Perdisci. HTTPOS: Sealing Information Leaks with Browser-side Obfuscation of Encrypted Flows. In *Network and Distributed System Security Symposium*, NDSS '11, San Diego, CA, USA, February 2011. The Internet Society.

[Mar19] Carolyn Duffy Marsan. 15 Worst Internet Privacy Scandals of All Time, January 2019. `http://www.networkworld.com/article/2185187/security/15-worst-internet-privacy-scandals-of-all-time.html`.

[Mas19] Mashable. 98,000 Hacked Snapchat Photos and Videos Posted Online, January 2019. `http://mashable.com/2014/10/13/the-snappening-photos-videos-posted`.

[MD04] Nick Mathewson and Roger Dingledine. Practical Traffic Analysis: Extending and Resisting Statistical Disclosure. In *Workshop on Privacy Enhancing Technologies*, PET '04, pages 17–34, Toronto, Canada, May 2004. Springer.

[MD05] Steven J. Murdoch and George Danezis. Low-Cost Traffic Analysis of Tor. In *IEEE Symposium on Security and Privacy*, SP '05, pages 183–195, Oakland, CA, USA, May 2005. IEEE.

[Met19] Tor Metrics. Tor Metrics: Users by Country, January 2019. `https://metrics.torproject.org/userstats-relay-table.html`.

[MFLS17] J.M. Myre, E. Frahm, D.J. Lilja, and M.O. Saar. TNT-NN: A Fast Active Set Method for Solving Large Non-Negative Least Squares Problems. In *International Conference on Computational Science*, ICCS '17, pages 755–764, Zurich, Switzerland, June 2017. Elsevier.

[MJB11] Michelle Madejski, Maritza Lupe Johnson, and Steven Michael Bellovin. The Failure of Online Social Network Privacy Settings. Technical report, Columbia University, 2011.

[MKJ+11] Prateek Mittal, Ahmed Khurshid, Joshua Juen, Matthew Caesar, and Nikita Borisov. Stealthy Traffic Analysis of Low-Latency Anonymous Communication Using Throughput Fingerprinting. In *ACM Conference on Computer and Communications Security*, CCS '11, pages 215–226, Chicago, IL, USA, October 2011. ACM.

[MMG+16] Mainack Mondal, Johnatan Messias, Saptarshi Ghosh, Krishna P. Gummadi, and Aniket Kate. Forgetting in social media: Understanding and controlling longitudinal exposure of socially shared data. In *Symposium on Usable Privacy and Security*, SOUPS '16, Denver, CO, USA, June 2016. USENIX Association.

[MO17] Stig F. Mjølsnes and Ruxandra F. Olimid. Easy 4G/LTE IMSI Catchers for Non-Programmers. In *Mathematical Methods, Models, and Architectures for Computer Network Security*, MMM-ACNS '17, pages 235–246, Warsaw, Poland, August 2017. Springer.

[Moc83] Paul V Mockapetris. RFC 883, Domain Names – Implementation and Specification. 1983.

[MR16] Daniel Moore and Thomas Rid. Cryptopolitik and the Darknet. *Survival*, 58(1):7–38, 2016.

[MW08] Steven J. Murdoch and Robert NM Watson. Metrics for Security and Performance in Low-Latency Anonymity Systems. In *Privacy Enhancing Technologies Symposium*, PETS '08, pages 115–132, Leuven, Belgium, 2008. Springer.

[MZ07] Steven J. Murdoch and Piotr Zieliński. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In *Workshop on Privacy Enhancing Technologies*, PET '07, pages 167–183, Ottawa, ON, Canada, June 2007. Springer.

[NBH18] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In *ACM Conference on Computer and Communications Security*, CCS '18, pages 1962–1976. ACM, 2018.

[NDCT07] Srijith K. Nair, Mohammad T. Dashti, Bruno Crispo, and Andrew S. Tanenbaum. A Hybrid PKI-IBC Based Ephemerizer System. In *New Approaches for Security, Privacy and Trust in Complex Environments*, SEC '07, Sandton, South Africa, May 2007. Springer.

[NHM17] Milad Nasr, Amir Houmansadr, and Arya Mazumdar. Compressive Traffic Analysis: A New Paradigm for Scalable Traffic Analysis. In *ACM Conference on Computer and Communications Security*, CCS '17, pages 2053–2069, Dallas, TX, USA, October 2017.

[NSZ+16] Rishab Nithyanand, Oleksii Starov, Adva Zair, Phillipa Gill, and Michael Schapira. Measuring and Mitigating AS-level Adversaries Against Tor. In *Network and Distributed System Se-*

curity Symposium, NDSS '16, San Diego, CA, USA, February 2016. The Internet Society.

[NWN+11] Jad Naous, Michael Walfish, Antonio Nicolosi, David Mazières, Michael Miller, and Arun Seehra. Verifying and Enforcing Network Paths with ICING. In *International Conference on emerging Networking EXperiments and Technologies*, CoNEXT '11, pages 30:1–30:12, Tokyo, Japan, December 2011. ACM.

[O'C05] Luke O'Connor. On Blending Attacks for Mixes with Memory. In *Inernational Workshop on Information Hiding*, IH '05, pages 39–52, Barcelona, Spain, June 2005. Springer.

[PBČC10] Christina Pöpper, David Basin, Srdjan Čapkun, and Cas Cremers. Keeping Data Secret Under Full Compromise Using Porter Devices. In *ACM Annual Computer Security Applications Conference*, ACSAC '10, Austin, TX, USA, December 2010. IEEE.

[PEL+17] Paul Pearce, Roya Ensafi, Frank Li, Nick Feamster, and Vern Paxson. Augur: Internet-Wide Detection of Connectivity Disruptions. In *IEEE Symposium on Security and Privacy*, SP '17, pages 427–443, San Jose, CA, USA, May 2017. IEEE.

[Per05a] Radia Perlman. File System Design With Assured Delete. In *International Security in Storage Workshop*, SISW '05. IEEE, 2005.

[Per05b] Radia Perlman. The Ephemerizer: Making Data Disappear. *Journal of Information System Security (JISSec)*, 1:51–68, 2005.

[Per11] Mike Perry. Experimental website fingerprinting defense, September 2011. `https://blog.torproject.org/experimental-defense-website-traffic-fingerprinting`.

[PHE+17] Ania M. Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. The Loopix Anonymity System. In *USENIX Security Symposium*, USENIX '17, pages 16–18, Washington, D.C., USA, August 2017. USENIX Association.

[PK01] Andreas Pfitzmann and Marit Köhntopp. Anonymity, Unobservability, and Pseudonymity—A Proposal for Terminology. In *Designing Privacy Enhancing Technologies*, pages 1–9. Springer, 2001.

[PLZ+18] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. Website Fingerprinting at Internet Scale. In *Network and Distributed System Security Symposium*, NDSS '16, San Diego, CA, USA, February 2018. The Internet Society.

[PNZE11] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website Fingerprinting in Onion Routing Based Anonymization Networks. In *Workshop on Privacy in the Electronic Society*, pages 103–114, Chicago, IL, USA, 2011. ACM.

[PPP+13] Panagiotis Papadopoulos, Antonis Papadogiannakis, Michalis Polychronakis, Apostolis Zarras, Thorsten Holz, and Evangelos P. Markatos. K-Subscription: Privacy-Preserving Microblogging Browsing Through Obfuscation. In *ACM Annual Computer Security Applications Conference*, ACSAC '13, New Orleans, LA, USA, December 2013. IEEE.

[PS01] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Computer Communication Review*, volume 31, pages 173–185. ACM, 2001.

[Ray01] Jean-François Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In *Designing Privacy Enhancing Technologies*, pages 10–29. Springer, 2001.

[RD12] Sirke Reimann and Markus Dürmuth. Timed Revocation of User Data: Long Expiration Times From Existing Infrastructure. In *Workshop on Privacy in the Electronic Society*, WPES '12, Raleigh, NC, USA, October 2012. ACM.

[RKHP19] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. Breaking LTE on Layer Two. In *IEEE Symposium on Security and Privacy*, SP '19, San Francisco, CA, USA, May 2019. IEEE.

[Ros07] David Rosenblum. What Anyone Can Know: The Privacy Risks of Social Networking Sites. *IEEE Symposium on Security and Privacy*, (3):40–49, 2007.

[RP17] Florentin Rochet and Olivier Pereira. Waterfilling: Balancing the Tor Network with Maximum Diversity. volume 2017 of *PoPETS '17*, pages 4–22. De Gruyter, 2017.

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

[San12] Sanjole Inc. WaveJudge 4900A LTE Analyzer. Technical report, February 2012.

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

[SBS05] Rob Sherwood, Bobby Bhattacharjee, and Aravind Srinivasan. P5: A protocol for Scalable Anonymous Communication. *Journal of Computer Security*, 13(6):839–876, 2005.

[SDB16] M. Syamkumar, R. Durairajan, and P. Barford. Bigfoot: A Geo-Based Visualization Methodology for Detecting BGP Threats. In *Symposium on Visualization for Cyber Security (VizSec)*, VizSec '16, pages 1–8, Baltimore, MD, USA, October 2016. IEEE.

[SDM04] Paul Syverson, Roger Dingledine, and Nick Mathewson. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium*, USENIX '04, Boston, MA, USA, June 2004. USENIX Association.

[SDS02] Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a Trickle to a Flood: Active Attacks on Several Mix Types. In *Inernational Workshop on Information Hiding*, IH '02, pages 36–52, Noordwijkerhout, The Netherlands, October 2002. Springer.

[Sec19] Securelist. Law Enforcement Agencies in Tor, January 2019. `https://securelist.com/law-enforcement-agencies-in-tor-impact-over-the-dark-web/67574/`.

[SEV+16] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. RAPTOR: Routing Attacks on Privacy in Tor. In *USENIX Security Symposium*, USENIX '16, pages 271–286, Washington, DC, USA, August 2016. USENIX Association.

[SFP16] Anant Shah, Romain Fontugne, and Christos Papadopoulos. Towards Characterizing International Routing Detours. In *Asian Internet Engineering Conference*, AINTEC '16, pages 17–24, Bangkok, Thailand, November 2016. ACM.

[SGR97] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous Connections and Onion Routing. In *IEEE Symposium on Security and Privacy*, SP '97, pages 44–54, Oakland, CA, USA, May 1997. IEEE.

[SGTH12] Max Schuchard, John Geddes, Christopher Thompson, and Nicholas Hopper. Routing Around Decoys. In *ACM Conference on Computer and Communications Security*, CCS '12, pages 85–96, Raleigh, NC, USA, October 2012. ACM.

[Sof19] Software Radio Systems. AirScope, January 2019. `http://www.softwareradiosystems.com/products/`.

[SRG00] P. Syverson, M. Reed, and D. Goldschlag. Onion Routing Access Configurations. In *DARPA Information Survivability Conference and Exposition*, DISCEX '00, pages 34–40, South Carolina, USA, 2000. IEEE.

[srs18] srsLTE. Open source SDR LTE software suite, January 2018. `https://github.com/srsLTE/srsLTE`.

[SRW+10] Hemant Sengar, Zhen Ren, Haining Wang, Duminda Wijesekera, and Sushil Jajodia. Tracking Skype Voip Calls Over the Internet. In *IEEE Conference on Computer Communications*, INFOCOM '10, pages 1–5, San Diego, CA, USA, July 2010. IEEE.

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

[STRL01] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an Analysis of Onion Routing Security. In *Designing Privacy Enhancing Technologies*, pages 96–114. Springer, 2001.

[SW06] Vitaly Shmatikov and Ming-Hsiu Wang. Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses. In *European Symposium on Research in Computer Security*, ES-

ORICS '06, pages 18–33, Hamburg, Germany, September 2006. Springer.

[SW14] Jody Sankey and Matthew Wright. Dovetail: Stronger Anonymity in Next-Generation Internet Routing. In *Privacy Enhancing Technologies Symposium*, PETS '14, pages 283–303, Amsterdam, The Netherlands, July 2014. Springer.

[SZ11] Yuval Shavitt and Noa Zilberman. A Geolocation Databases Study, 2011.

[The19a] The Register. iCloud Fiasco: 100 Famous Women Exposed Nude Online, January 2019. `http://www.theregister.co.uk/2014/08/31/jlaw_upton_caught_in_celeb_nude_pics_hack`.

[The19b] The Tor Project. Ethical Tor Research: Guidelines, January 2019. `https://blog.torproject.org/blog/ethical-tor-research-guidelines`.

[The19c] The Tor Project. The Onion Router, January 2019. `https://www.torproject.org`.

[The19d] The Tor Project. Tor Metrics, January 2019. `https://metrics.torproject.org`.

[The19e] The Tor Project. Tor Security Advisory: "Relay Early" Traffic Confirmation Attack, January 2019. `https://blog.torproject.org/blog/tor-security-advisory-relay-early-traffic-confirmation-attack`.

[TIM+13] Z.H. Talukder, S.S. Islam, D. Mahjabeen, A. Ahmed, S. Rafique, and M.A. Rashid. Cell Coverage Evaluation for LTE and WiMAX in Wireless Communication System. *World Applied Sciences Journal*, 22(10):1486–1491, January 2013.

[Tor19a] Tor Tickets. Check Maxmind GeoIPLocation Database before distributing, January 2019. `https://trac.torproject.org/projects/tor/ticket/26240`.

[Tor19b] Torspec. NTor Handshake Proposal, January 2019. `https://gitweb.torproject.org/torspec.git/tree/proposals/216-ntor-handshake.txt`.

[TVSS18] Mina Tsay-Vogel, James Shanahan, and Nancy Signorielli. Social Media Cultivating Perceptions of Privacy: A 5-Year Analysis of Privacy Attitudes and Self-Disclosure Behaviors among Facebook Users. *New Media & Society*, 20(1):141–161, 2018.

[vH05] S. Čapkun and J. P. Hubaux. Secure Positioning of Wireless Devices with Application to Sensor Networks. In *IEEE Conference on Computer Communications*, INFOCOM '05, pages 1917–1928, Miami, FL, USA, March 2005. IEEE.

[Wan15] Zhanyi Wang. The Applications of Deep Learning on Traffic Identification. In *Proceedings of Black Hat Technical Security Conference*, BLACKHAT '15, Las Vegas, NV, USA, August 2015.

[WBF+11] Yong Wang, Daniel Burgener, Marcel Flores, Aleksandar Kuzmanovic, and Cheng Huang. Towards Street-Level Client-Independent IP Geolocation. In *USENIX Symposium on Networked Systems Design and Implementation*, NSDI '11, pages 27–27, Boston, MA, USA, March 2011. USENIX Association.

[WBFG12] Tao Wang, Kevin Bauer, Clara Forero, and Ian Goldberg. Congestion-Aware Path Selection for Tor. In *Financial Cryptography*, FC '12, pages 98–113, Kralendijk, Bonaire, February 2012. Springer.

[WCJ05] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. Tracking Anonymous Peer-to-Peer VoIP Calls on the Internet. In *ACM Conference on Computer and Communications Security*, CCS '05, pages 81–91, Alexandira, VA, USA, November 2005. ACM.

[WCJ07] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems. In *IEEE Symposium on Security and Privacy*, SP '07, pages 116–130, Oakland, CA, USA, May 2007. IEEE.

[WCM09] Charles V. Wright, Scott E. Coull, and Fabian Monrose. Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis. In *Network and Distributed System Security Symposium*, NDSS '09, San Diego, CA, USA, February 2009. The Internet Society.

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

[WCSG18] Zachary Weinberg, Shinyoung Cho, Vyas Sekar, and Phillipa Gill. How to Catch when Proxies Lie: Verifying the Physical Locations of Network Proxies with Active Geolocation. In *ACM SIGCOMM Conference on Internet Measurement*, IMC '18, Boston, MA, USA, October 2018. ACM.

[WG13] Tao Wang and Ian Goldberg. Improved Website Fingerprinting on Tor. In *Workshop on Privacy in the Electronic Society*, WPES '13, Berlin, Germany, 2013. ACM.

[WG17] Tao Wang and Ian Goldberg. Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks. In *USENIX Security Symposium*, USENIX '17, pages 1375–1390, Washington, D.C., USA, August 2017. USENIX Association.

[WHH+10] Scott Wolchok, Owen S. Hofmann, Nadia Heninger, Edward W. Felten, J. Alex Halderman, Christopher J. Rossbach, Brent Waters, and Emmett Witchel. Defeating Vanish With Low-Cost

Sybil Attacks Against Large DHTs. In *Network and Distributed System Security Symposium*, NDSS '10, San Diego, CA, USA, February 2010. The Internet Society.

[Wis19] Wisemetrics. Your Tweet Half-Life Is 1 Billion Times Shorter Than Carbon 14's, January 2019. `http://blog.wisemetrics.com/tweet-isbillion-time-shorter-than-carbon14/`.

[WL12] Philipp Winter and Stefan Lindskog. How the Great Firewall of China is Blocking Tor. FOCI '12, Bellevue, WA, USA, August 2012. USENIX Association.

[WSH14] Eric Wustrow, Colleen M. Swanson, and J. Alex Halderman. TapDance: End-to-Middle Anticensorship without Flow Blocking. In *USENIX Security Symposium*, USENIX '14, pages 159–174, San Diego, CA, USA, August 2014. USENIX Association.

[WSS07] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. In *USENIX Symposium on Networked Systems Design and Implementation*, NSDI '07, pages 23–23, Santa Clara, CA, USA, June 2007. USENIX Association.

[WTBS13] Chris Wacek, Henry Tan, Kevin S Bauer, and Micah Sherr. An Empirical Evaluation of Relay Selection in Tor. NDSS '13, San Diego, CA, USA, February 2013. The Internet Society.

[WWY+12] Zachary Weinberg, Jeffrey Wang, Vinod Yegneswaran, Linda Briesemeister, Steven Cheung, Frank Wang, and Dan Boneh. StegoTorus: A Camouflage Proxy for the Tor Anonymity System. In *ACM Conference on Computer and Communications Security*, CCS '12, pages 109–120, Raleigh, NC, USA, October 2012. ACM.

[YFG+07] Wei Yu, Xinwen Fu, Steve Graham, Dong Xuan, and Wei Zhao. DSSS-Based Flow Marking Technique for Invisible Traceback.

In *IEEE Symposium on Security and Privacy*, SP '07, pages 18–32, Oakland, CA, USA, May 2007. IEEE.

[ZFG+04] Ye Zhu, Xinwen Fu, Bryan Graham, Riccardo Bettati, and Wei Zhao. On Flow Correlation Attacks and Countermeasures in Mix Networks. In *Workshop on Privacy Enhancing Technologies*, PET '04, pages 207–225, Toronto, Canada, May 2004. Springer.

[ZKDP16] Apostolis Zarras, Katharina Kohls, Markus Dürmuth, and Christina Pöpper. Neuralyzer: Flexible Expiration Times for the Revocation of Online Data. In *Conference on Data and Application Security and Privacy*, CODASPY '16, pages 14–25, New Orleans, LA, USA, March 2016. ACM.

[ZNR07] David John Zage and Cristina Nita-Rotaru. On the Accuracy of Decentralized Virtual Coordinate Systems in Adversarial Networks. In *ACM Conference on Computer and Communications Security*, CCS '07, pages 214–224, Alexandria, Virginia, USA, October 2007. ACM.

[ZSXF10] Lingfang Zeng, Zhan Shi, Shengjie Xu, and Dan Feng. Safe-Vanish: An Improved Data Self-Destruction for Protecting Data Privacy. In *International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010.