Security, Privacy and Economics of Online Advertising

THÈSE N^{O} 5664 (2013)

PRÉSENTÉE LE 22 MARS 2013

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 1

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Nevena VRATONJIC

acceptée sur proposition du jury:

Prof. K. Aberer, président du jury
Prof. J.-P. Hubaux, directeur de thèse
Prof. A. Argyraki, rapporteur
Prof. J. Grossklags, rapporteur
Dr M. Strasser, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2013

To my family

Abstract

Online advertising is at the core of today's Web: it is the main business model, generating large annual revenues expressed in tens of billions of dollars that sponsor most of the online content and services. Online advertising consists of delivering marketing messages, embedded into Web content, to a targeted audience. In this model, entities attract Web traffic by offering the content and services for free and charge advertisers for including advertisements in this traffic (i.e., advertisers pay for users' attention and interests). Online advertising is a very successful form of advertising as it allows for advertisements (ads) to be targeted to individual users' interests; especially when advertisements are served on users' mobile devices, as ads can be targeted to users' locations and the corresponding context.

However, online advertising also introduces a number of problems. Given the high ad revenue at stake, fraudsters have economic incentives to exploit the ad system and generate profit from it. Unfortunately, to achieve this goal, they often compromise users' online security (e.g., via malware, phishing, etc.). For the purpose of maximizing the revenue by matching ads to users' interests, a number of techniques are deployed, aimed at tracking and profiling users' digital footprints, i.e., their behavior in the digital world. These techniques introduce new threats to users' privacy. Consequently, some users adopt ad-avoidance tools that prevent the download of advertisements and partially thwart user profiling. Such user behavior, as well as exploits of ad systems, have economic implications as they undermine the online advertising business model. Meddling with advertising revenue disrupts the current economic model of the Web, the consequences of which are unclear.

Given that today's Web model relies on online advertising revenue in order for users to have access and consume content and services for "free", coupled with the fact that there are many threats that could jeopardize this model, in this thesis we address the security, privacy and economic issues stemming from this fundamental element of the Web.

In the first part of the thesis, we investigate the vulnerabilities of online advertising systems. We identify how an adversary can exploit the ad system to generate profit for itself, notably by performing inflight modification of ad traffic. We provide a proof-of-concept implementation of the identified threat on Wi-Fi routers. We propose a collaborative approach for securing online advertising and Web browsing against such threats. By investigating how a certificate-based authentication is deployed in practice, we assess the potential of relying on certificate-based authentication as a building block of a solution to protect the ad revenue. We propose a multidisciplinary approach for improving the current state of certificate-based authentication on the Web.

In the second part of the thesis, we study the economics of ad systems' exploits and certain potential countermeasures. We evaluate the potential of different solutions aimed at protecting ad revenue being implemented by the stakeholders (e.g., Internet Service Providers or ad networks) and the conditions under which this is likely to happen. We also study the

economic ramifications of ad-avoidance technologies on the monetization of online content. We use game-theory to model the strategic behavior of involved entities and their interactions.

In the third part of the thesis, we focus on privacy implications of online advertising. We identify a novel threat to users' location privacy that enables service providers to geolocate users with high accuracy, which is needed to serve location-targeted ads for local businesses. We draw attention to the large scale of the threat and the potential impact on users' location privacy.

Keywords: online advertising, ad fraud, inflight ad-traffic modification, economics, game theory, Web certificates, botnets, ISPs, location privacy

Résumé

La publicité en ligne est au coeur du Web d'aujourd'hui: elle est le principal modèle d'affaires, générant d'énormes revenus annuels (dizaines de milliards de dollars) qui financent la plupart des contenus et services en ligne. La publicité en ligne consiste à distribuer des messages marketing, intégrés dans du contenu Web, à un public ciblé. Dans ce modèle, des entités attirent du trafic Web en offrant du contenu et des services gratuitement et en faisant payer les publicitaires pour inclure des publicités dans ce trafic (i.e., les publicitaires paient pour l'attention et les intérêts des utilisateurs). La publicité en ligne est une forme de publicité très fructueuse car elle permet de cibler la publicité par rapport aux intérêts des utilisateurs. Spécialement quand les publicités sont délivrées sur les appareils mobiles des utilisateurs car ces publicités peuvent être adaptées à la localisation et au contexte des utilisateurs.

Cependant, la publicité en ligne génère aussi un certain nombre de problèmes. Etant donné les revenus élevés en jeu, les fraudeurs ont un intérêt économique à exploiter le système de publicité à leur propre profit. Malheureusement, ils compromettent souvent la sécurité des utilisateurs afin d'atteindre leur but (par ex., par des malwares, de l'hameçonnage, etc.). Afin de maximiser les profits en personnalisant les publicités aux intérêts des utilisateurs, de nombreuses techniques sont déployées, traçant et profilant les empreintes numériques des utilisateurs, i.e., leur comportement dans le monde numérique. Ces techniques entraînent de nouvelles menaces sur la protection des données privées des utilisateurs. Par conséquent, certains utilisateurs installent des outils de blocage des publicités qui empêchent le téléchargement de ces publicités et déjouent partiellement le profilage. De tels comportements, ainsi que l'utilisation malicieuse des systèmes publicitaires, ont des implications économiques car ils sapent le modèle d'affaires de la publicité en ligne. L'altération du revenu de la publicité perturbe le modèle économique actuel du Web, et ses conséquences sont encore floues.

Etant donné que le Web repose sur les revenus publicitaires afin de donner accès "gratuitement" à du contenu et des services aux utilisateurs, ainsi que le fait qu'il y ait de nombreuses menaces qui puissent compromettre ce modèle, cette thèse aborde les problématiques économiques, de sécurité, et de protection des données découlant de cet élément essentiel du Web.

Dans la première partie de la thèse, nous examinons les vulnérabilités des systèmes de publicité en ligne. Nous décrivons commment un attaquant peut exploiter le système publicitaire pour générer des profits personnels, notamment en modifiant le trafic publicitaire en transit. Nous fournissons une implémentation démontrant la menace sur un routeur WiFi. Nous proposons une approche collaborative pour sécuriser la publicité en ligne et la navigation Web contre de telles menaces. Nous évaluons le potentiel de l'authentification basée sur des certificats comme composante d'une solution pour protéger les revenus publicitaires, en étudiant comment cette authentification est déployée en pratique. Nous proposons une approche pluridisciplinaire afin d'améliorer la situation actuelle de l'authentification basée sur

des certificats sur le Web.

Dans la seconde partie de la thèse, nous étudions l'économie des attaques contre les systèmes publicitaires et les potentielles contre-mesures. Nous évaluons différentes solutions visant à protéger les revenus publicitaires implémentées par les parties prenantes (par ex., fournisseurs d'accès Internet ou réseaux publicitaires) et les conditions dans lesquelles celles-ci ont des chances de s'appliquer. Nous étudions également les implications économiques des technologies de blocage des publicités vis-à-vis de la monétisation du contenu en ligne. Nous utilisons la théorie des jeux pour modéliser le comportement stratégique des entités impliquées, et leurs interactions.

Dans la troisième partie de la thèse, nous nous concentrons sur les implications de la publicité en ligne vis-à-vis de la sphère privée. Nous identifions une nouvelle menace sur la protection des données de localisation des utilisateurs qui permet aux fournisseurs de service de géolocaliser les utilisateurs avec une grande précision, ce qui est nécessaire afin de fournir des publicités ciblées par la localisation. Nous attirons l'attention sur cette menace à grande échelle et l'impact potentiel sur la protection des données de localisation des utilisateurs.

Mots-clés: publicité en ligne, fraude publicitaire, modification du trafic publicitaire en transit, économie, théorie des jeux, certificats Web, botnets, FAIs, protection des données de localisation

Acknowledgments

I would like to express my gratitude to all the wonderful people who have contributed to this thesis and who have made my PhD years such a wonderful time of my life.

First and foremost, I am grateful to my advisor Prof. Jean-Pierre Hubaux for giving me the opportunity to do research in a positive and stimulating environment of LCA1, and above all, for his invaluable guidance and support throughout the PhD process. I have learned a lot from him, on both a professional and personal level. Jean-Pierre, thank you for everything.

I would like to thank my thesis committee members: Prof. Katerina Argyraki, Prof. Jens Grossklags, Dr Mario Strasser and Prof. Karl Aberer, for the time and effort they put into reviewing this dissertation. Special thanks to Prof. Jens Grossklags; during his stay at EPFL we elaborated the ideas on the problem of online content monetization.

My appreciation goes to my co-authors for fruitful collaborations and for contributing considerably to this thesis: Prof. Hossein Manshaei, Dr Julien Freudiger, Dr Maxim Raya, Prof. Jens Grossklags, Prof. David Parkes, Prof. Márk Félegyházi, Dr Kévin Huguenin and Vincent Bindschaedler. In particular, thanks to Julien and Mark for our early work related to online advertising systems, the foundation for this thesis.

I am thankful to all my colleagues at EPFL who have also contributed to this thesis in many ways: Marcin, Igor, Mathias, Reza, Jacques, Berker, Murtuza, Panos, Aravind, Priya, Steve, George, Pedram, Wojciech, Mirco, for discussing ideas, challenging and reviewing my work and creating a great atmosphere. I would like to express my gratitude to Patricia, Danielle and Angela for all their help with administration issues, and to Holly for her persistent efforts to improve my English and to system administrators for providing a great computing infrastructure.

The years of my PhD experience have been enriched with many wonderful people I have met and the great experiences we have shared. Special thanks to Kasia, Michal, Adriana, Gleb, Maxim and all DOSErs for the many fun events, extraordinary discussions and the enjoyable time spent together; to Julien, for the numerous interesting conversations, for sharing his time, ideas and expertise; and to my officemate Marcin, for being a good friend and for all the support, patience, discussions and advice throughout our PhD life in BC 200.

I am indebted to my Serbian friends for making me feel at home. Furthermore, I am grateful to Ruzica for her unconditional friendship; to Milica and Andrijana for all the joyful girly times and laughs we shared; and to Bane, for all the wonderful times he has generously shared with me and for his help in numerous occasions. My appreciation goes to my flatmate Aleksandar, for always being cheerful and filling our house with laughter. I am thankful to Ana and Zarko for making our first days in Lausanne much easier and enjoyable. I also want to pay tribute to my friends with whom I shared many special moments at home in Serbia and around the world, Milena, Natasa, Nevena, Maja, Marko, Milos, Vlada. Thank you for cherishing and nurturing our friendship regardless of the distance.

My heartfelt thanks go to Sam, for his endless understanding, support and encouragement each step of the way. Thank you for your love and for bringing happiness into my life.

Finally, I want to thank my family, my parents Danica and Milan, and my sister Milena, for their unconditional support and encouragement. They have my eternal gratitude and love. To them, I dedicate this thesis.

Mama, tata, Milena, hvala vam puno na svemu. Vama posvećujem ovu tezu.

Contents

Introduction

> It is a lazy Sunday morning and Alice is at a coffee shop, reading an online issue of her favorite fashion magazine on her tablet. Just next to the content of the article she is reading, she notices an advertisement for a nearby shoe store. Surprised, she thinks "Oh, how convenient, let me check out what they have and if I like something I can actually go there and try it on.". She clicks on the ad that leads her to the shoe store's website. Another thought quickly crosses her mind: "Was it just luck that the advertised shoe store is located so close by or was it selected on purpose because someone knows where I am?". But the Web page loads and the store features so many nice shoe models that Alice likes that she immediately forgets about her concerns and only thinks about how glad she is that she has learned about the store: "Ads are really cool and useful!". She spends some time thinking about a particular pair of shoes, but decides she should try them on first before buying. In the evening, Alice logs into her Facebook account. While she is checking the newsfeed of her friends' activities, she notices on the side an image of the shoes she was thinking of buying earlier in the day. Alice is confused: "OMG! How come the exact same model of shoes is on my Facebook page? Is this a weird coincidence? Does Facebook know I am interested in these shoes and is now showing me this ad?". She concludes that the fears of her online actions being tracked are the most likely scenario and that this fishy situation is probably happening because she has clicked on the shoe store advertisement earlier. She decides: "This is, like, so creepy, I won't be clicking on ads anymore, they are so privacy-intrusive!"

This example illustrates how online advertising works: online ads appear together with the content while users browse the Web; advertisers pay for users' exposure to ads, which generates revenue for the mediators who include ads with content (i.e., ad networks) and content providers. Online advertising has been so successful that it has become the main business model on the Web, generating annual revenues of tens of billions of dollars, sufficient to sponsor most of the online content and services [148]. This has encouraged the industry (e.g., ad networks) to increasingly deploy techniques to track and profile users' online behavior, in an attempt to learn users' interests and needs in order to target them with relevant advertisements. Unfortunately, these techniques are not always privacy friendly. Users' awareness of privacy issues related to such practices is increasing, especially because they find themselves more and more often in the situations like in the aforementioned example [77].

The example also illustrates users' dual perception of online advertisements: the line between ads being perceived as valuable versus privacy intrusive can be thin and easy to cross [216]. An acceptable use of private information in one situation might be an unacceptable in-

vasion of privacy in another. An ad delivered to a user, supposedly at the right place and time in order to make the message relevant, could be very intrusive if the message is unanticipated by the user. Providing relevant, targeted, yet privacy non-intrusive advertisements is thus a complex concept. As seen in the example, users' attitudes towards ads can easily change [119]. A contributing factor is that, as in the example, users typically do not account for the benefits of accessing the content for "free", whereas they would have to pay for a printed copy at the newsstand; or benefiting from social networking services free of charge. All these benefits are actually paid for with users' attention, i.e., users' exposure to ads. Thus, users becoming adverse to ads could disrupt the online advertising business model and without the ad revenue service providers would need to look for alternative monetization strategies, which might result in users having to pay for services they have been using free of charge so far.

Online advertising can also be undermined by an adversary who, driven by monetary incentives, engages in ad fraud and interferes with the ads users' receive. For example, a hotspot provider can try to generate profit by providing Internet access for free and injecting ads into users' traffic. Such practices create a number of security and privacy issues for end users and deprive legitimate advertising entities and the content providers from ad revenue.

Thus, not surprisingly, both industry and academia are working towards protecting the ad revenue from the many perilous threats that can meddle with the online advertising business model. Efforts have been directed towards various relevant aspects of online advertising (e.g., privacy, security, economics, ad targeting efficiency, etc.). However, with the rapidly evolving technologies, proliferation of powerful mobile devices and the changing trends in how users access and consume content and services, new issues and challenges are emerging. Therefore, much research is still needed in order to design and maintain more robust ad systems and protect the Internet business model. This has motivated us to pursue the investigations that comprise this thesis.

Contributions

In this thesis, we address security, privacy and economic issues related to online advertising. We make the following main contributions:

1. We identify a new type of ad fraud and the underlying attacks on online advertising systems that can generate a significant revenue for fraudsters. The fraud consists of inflight modification of ad traffic, resulting in users receiving ads of an adversary's choice, thus generating ad revenue for the adversary instead of the legitimate entities. We contribute a proof-of-concept implementation on a wireless router to demonstrate that the attacks can successfully run even on resource constrained devices. Over the course of our work, several instances of the identified ad fraud were detected in practice, further validating our findings of the feasibility and incentives to engage in such activities. We propose a collaborative approach for providing authenticity and integrity of Web content and advertisements in order to protect against this type of fraud.

2. We provide a comprehensive assessment of the current level of security provided by HTTPS and certificate-based authentication on the Web. We conduct a large-scale empirical analysis that considers the top one million most popular websites and show that HTTPS is not sufficiently deployed (e.g., only 22.6% of the websites requiring users' login credentials are implement via HTTPS) and that only 16% of the surveyed web-

sites implement certificate-based authentication properly. We discuss multiple reasons (economic, legal and social) that might have lead to the failure of the current model for Web security, and we propose a multidisciplinary approach to improve certificate-based authentication. Our findings are important because HTTPS and certificate-based authentication are de-facto the solutions used to secure Internet communications and because of their potential to secure ad revenue.

3. We provide game-theoretic frameworks that allow for the study of strategic interactions: (i) between the stakeholders, notably ad networks and ISPs, and (ii) between online content providers and users. The outcomes of such mutually dependent actions can have a great effect on the Internet, in particular on users' security and privacy online. We investigate the conditions under which the stakeholders have incentives to invest (individually or jointly) in improving Web security (e.g., by securing the delivery of online content and advertisements or by thwarting botnets) or in improving user profiling to provide more personalized services and targeted advertisements. We also investigate alternative monetization strategies of content providers that face users who make use of ad-avoidance tools. We show that a strategic approach of treating users individually, investing into ad-avoidance detection tools and offering alternative monetization strategies (e.g., fee-financed content) can yield higher revenues and better respects users' preferences. This framework allows content providers to better understand their users' preferences for content and advertisements. Such investigation is much needed and timely, because of users' increasing aversion to ads due to online tracking and profiling practices. Otherwise, this trend can lead to a wider adoption of ad-avoidance tools and disrupt the online advertising business model.

4. We reveal a new threat to users' location-privacy, based on which service providers can geolocate users who connect through shared Wi-Fi access points, and they can do so with high accuracy (within a few hundreds of meters) simply based on users' IP addresses. The service provider only needs to perform a passive analysis of the received traffic, which is what Web services already do to improve the quality and relevance of the offered services. In fact, once the service provider has access to sporadic user-location information, it is also able to reconstruct entire trajectories, produce patterns of user movement habits, or infer other information about the user, e.g., users' real identities, interests and activities. Service providers have incentives to learn such information as it can significantly improve personalized services and ad targeting, thus the generated revenue as well. Because the threat is inherent in the way networks operate, specifically Network Address Translation (NAT), the scale of the threat is significant and potentially affects a large fraction of users. We propose an analytical framework that quantifies location-privacy threat at a given Wi-Fi access point. In addition, we experimentally show the large scale of the threat based on users' traffic to Google, which we collect at a few deployed hotspots: Google is able to quickly correlate a given IP address with the location of the hotspot a user connects from, consequently Google can successfully geolocate almost all the users who make use of Google services.

Thesis Outline

Part I is devoted to the security of online advertising systems. In Chapter 1, we provide a general introduction to online advertising, we address vulnerabilities of online advertising systems, the attacks and possible countermeasures. In Chapter 2, we assess the potential of certificate-based authentication being used to secure online advertising systems by investigating deployment practices on the Web. Part II is devoted to the economics of meddling with the online advertising business model. In Chapter 3, we investigate the problem of ISPs becoming strategic participants in the online advertising business. In Chapter 4, we analyze the incentives of ISPs and ad networks to fight botnet ad fraud. In Chapter 5, we study ad-avoidance tools and the economic ramifications of their use on the monetization of online content. Part III is devoted to privacy issues stemming from online advertising. In Chapter 6, we identify a novel threat to users' location-privacy, motivated by the need of service providers to geolocate users with an accuracy that allows ad-targeting for local businesses.

Publications

Chapter 1 is a combination of the results from [224] and [225]. Chapter 2 is a presentation of the results from [223]. Chapter 3 and Chapter 4 are based on [221] and [227], respectively. Chapter 5 is based on [226]. Chapter 6 is an extended version of [222]; the Chapter content is currently under submission to Privacy Enhancing Technologies Symposium.

Part I

Security of Online Advertising

Chapter 1

Ad Fraud and Countermeasures

Over the last decade, online advertising has become a major component of the Web, leading to large annual revenues (e.g., $31.7 billion in the US in 2011 [148]). Internet advertising is a very successful form of advertising as it provides an easy and effective way for advertisements to be targeted to individual users' interests. Unfortunately, fraudsters were able to exploit several vulnerabilities of the online advertising model and started abusing the system in order to make a profit out of it. These attacks are illegal only in a few countries and states (e.g., click fraud is a felony covered by Penal Code 502 in California and the computer misuse act in the UK). In most of the cases, the fraudsters instead violate terms of service of online advertising networks (e.g., in Nigeria where there is no law against this type of cybercrime and in India, where companies even advertised in national newspapers looking for people willing to use computers to click on ads, with no repercussions from authorities). In this chapter, we aim to provide a better understanding of vulnerabilities of online advertising systems, the attacks, and possible countermeasures. We first address the online advertising system model and discuss different revenue models for it. We explain vulnerabilities of these models and how they can be exploited. We survey the well-known types of ad fraud: click fraud, malvertising and adware. We discuss in more detail some of the attack instances reported in practice. We then focus on a novel type of ad fraud: inflight modification of ad traffic. We identify inflight attacks on ad traffic and provide a proof-of-concept implementation on Wi-Fi routers. For each type of ad fraud, we address how fraudsters can make profit from the existing advertising system and we address possible countermeasures.

Chapter Outline In Section 1.1 we explain how online advertising systems work, the ad serving system architecture, different techniques to target ads and revenue models. We present well known types of ad fraud, notably click fraud, malvertising and adware, we discuss how fraudsters generate profit and address possible countermeasures in Section 1.2. In Section 1.3 we identify a novel type of ad fraud consisting in inlight modification of ad traffic and we estimate the revenue a fradster can generate from this type of fraud. We also provide a proof-of-concept implementation of the introduced inflight attacks on ad traffic on a Wi-Fi router. Finally, in Section 1.4 we propose a collaborative approach to provide authenticity and integrity of Web content and advertisements and we summarize our findings in Section 1.5.

1.1 Advertising on the Internet

Online advertising is a form of marketing that relies on the Internet to deliver marketing messages to the targeted users. Internet advertisement typically comprises a short text, an image, or an animation embedded into a Web page. The purpose of an ad is generally to capture a user's attention and persuade him to purchase or to consume a particular product or a service and, consequently, to increase the revenue of the advertiser. Advertisers pay for their ads to appear online, thus online advertising has become the major business model for monetizing online content. In contrast to other types of media (e.g., television or radio), online advertisements are not limited to an audience at a given time or a geographic location. An additional benefit is that online advertising allows for the customization of advertisements, thus increasing the probability that a user is interested in the advertised products and services. Hence, many advertisers, realizing the opportunities of online advertising, invest significant budgets into this form of advertising. Consequently, for many of the websites that users visit, a number of advertisements appear together with the content of a Web page.

Next, we look at how online advertisements are embedded into the content of Web pages, which techniques are used to tailor ads to users' interests and the main revenue models in online advertising.

1.1.1 Ad Serving Architecture

Ads are embedded into Web pages either through an ad serving system, or by websites themselves. Although it might be a straightforward task for major publishers with marketing units to sell the advertising space on their Web pages to advertisers, this is not the case for a large number of small publishers for which the overhead of doing so may surpass the benefits. *Ad networks* emerged as a solution to increase the reach of online advertising campaigns across these small publishers as well. Publishers offer their advertising space to ad networks that deal with advertisers and sell the advertising space on behalf of publishers.

The prevalent model of the Internet advertisement serving architecture is depicted in Figure 1.1. In this model, an ad network plays the role of intermediary between advertisers and publishers, and its job is to automatically include ads into the appropriate online content. For this purpose, the ad network provides publishers with the HTML code that publishers should include (i.e., copy-paste) into the HTML code of their Web pages. When users browse these Web pages, relevant ads appear together with the publishers' content.

The protocol, illustrated in Figure 1.1, can be represented as follows:

1. A user's browser issues a request for the Web page corresponding to the URL the user types into browser's address bar.

2. The downloaded Web page contains the publisher's content and the block of the HTML code provided by the ad network. The HTML code redirects the browser to communicate with an *ad server*, that belongs to the ad network, and download the ads that should accompany the publisher's content. This approach makes the ad serving system scalable, as the workload is distributed across users, rather than having a website communicate with an ad network on behalf of each user and deliver the ads together with the content. In addition, it allows ad servers and advertisers to keep the control, as ads are stored and maintained at their servers.

3. Typically, the user's browser first requests a script (e.g., JavaScript) from the ad server.

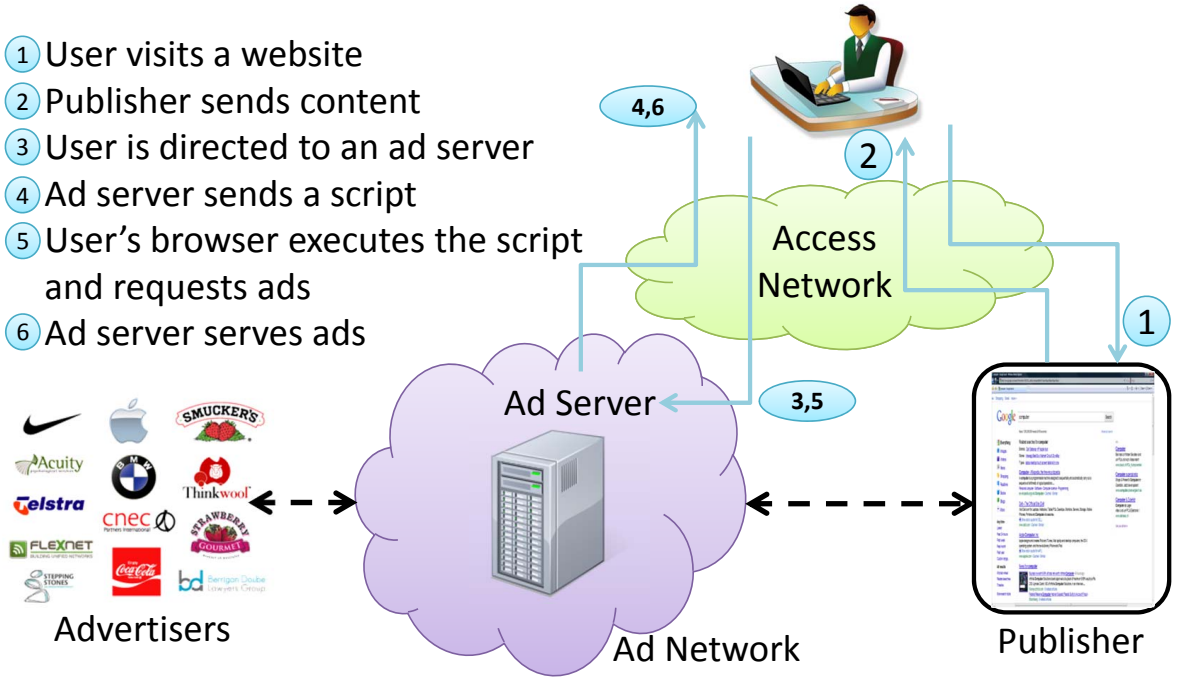


Figure 1.1: The ad serving architecture. *Advertisers* subscribe to an *Ad Network* whose role is to automatically embed ads into related Web pages. *Publishers* and ad networks have a contractual agreement (dashed arrow) that lets ad networks add advertisement to publishers' Web pages. Ads are stored at *Ad Servers*, which belong to the ad network. When a *User* visits a website of a publisher that hosts ads (step 1), the user's browser starts downloading the content of the Web page (step 2), and is then directed to one of the ad servers belonging to the ad network (step 3). During the first communication with the ad server, a script is served to the user (step 4) that executes on the user's machine and requests ads from the ad server (step 5). The ad server chooses and serves ads that match users' interest (step 6) in order to maximize the potential ad revenue.

4. When the script is fetched, it executes locally on the user's machine and collects certain parameters that influence the selection of ads by the ad server, including the HTTP cookies if they were deposited by the ad server during previous interactions. Cookies uniquely identify users and enable the profiling of their browsing preferences. This enables ad servers to track users across multiple websites. Besides collecting relevant information about the user before actually serving the ads, an additional benefit of having the HTML code that directs users to first download the script is that it is simple and easy to maintain, as only a few lines of a generic code (a reference to the JavaScript) are added in the code of Web pages. Thus, if the ad network wants to modify the way ads are included in online content, it can simply modify the script that is hosted on their servers, rather than requesting each of the associated publishers to implement the corresponding modifications.

5. Information collected by the script is communicated back to the ad server with the request for ads.

6. The ad server chooses and serves the most appropriate ads for the given user. The browser merges the ads with previously downloaded elements of the Web page.

Due to its many advantages, this approach is widely used in practice. However, there are several drawbacks as well. Because users fetch ads from third-party servers (i.e., servers different from publishers' servers), the ad serving technology slows down the display of Web pages, consumes extra bandwidth, can be used as an attack vector to compromise the security of users' machines and affects the privacy of users [160, 161].

In an alternative online ad serving architecture, a website can embed advertisements locally and serve them to users, together with the content of a Web page. This ad serving technique is not very popular because it puts more workload on the Web servers compared to the previous approach, thus it does not scale as well. Some of the ad networks deploy this model when serving mobile advertisements, i.e., advertisements that are displayed on users' mobile devices. In this particular case it might be justified to put the overhead on Web servers rather than on users, because with mobile devices the available bandwidth and computational power is limited, the latencies are higher and the communication is more expensive for the user. The previous approach is also preferred by ad networks because direct communication of users to ad servers allows for better profiling of users' online behavior, thus better matching of ads to users interests and consequently higher potential revenues.

1.1.2 Targeted Advertising

A notable difference between online and traditional advertising (e.g., television, radio) is that online ads can be *targeted* to individual user's interests. To maximize the potential revenue, ad networks use *ad targeting techniques* to serve the ads that match users' interests. The most popular ad targeting techniques are *contextual*, *behavioral* and *location-based* targeting. With contextual targeting, ads are related to the content the user is currently viewing. Behavioral targeting customizes ads based on users' *digital footprints*, i.e., information about the observed behavior of the users in the digital world, including usage of the Internet, mobile phone, etc. With location-based targeting, users receive location-specific ads on their (mobile) devices.

Targeted advertising aims at providing each user with the ads that best suit his interests. At ad servers, users' interests can be expressed with *keywords*. The ad server associates ads with each keyword and runs auction algorithms to select the most relevant ads and the order in which they appear on the Web page, with the goal of maximizing the profit of both advertisers and publishers hosting the ads. In particular, small businesses find that online advertising offers maximum exposure for a minimal cost.

1.1.3 Revenue Models

There are three main revenue models: Advertisers pay the ad network on a per *impression*, per *click* or per *action* basis.

In the *pay-per-impression* (PPI) model, advertisers pay the ad network for the exposure of their ads to end users, i.e., there is a *cost-per-mille* (CPM) (cost to expose one ad to one thousand users). This model is widely used for *brand advertising*, i.e., increasing customers' awareness and ability to recall and recognize the brand, typically by displaying banner ads. Brand awareness is of critical importance as customers will not consider a brand if they are not aware of it. The impression-based model is an online counterpart to the traditional mediums for conveying a brand image to customers, such as print (where impressions are created by

the placement of ads in subway cars, billboards, etc.) and television (where impressions are created by the emission of commercials). Thus, many advertisers choose impression-based online advertising as a way to establish their brand as a trustworthy friend to the consumer.

In the *pay-per-click* (PPC) model, advertisers pay the ad network a *cost-per-click* (CPC) for each user-generated click on an ad that directs the user's browser to the advertised website. From an advertiser's point of view, a click on an ad represents a user's choice. The benefit of the PPC model is that it offers instant feedback and the opportunity to measure the effectiveness of an advertising campaign. The success of an advertising campaign can be expressed with *clickthrough rate* (CTR). A CTR is obtained by dividing the "number of users who clicked on an ad" on a Web page by the "number of times the ad was delivered" (impressions). As of 2006, PPC started gaining prevalence over other revenue models. The trend continued over the years to reach approximately 65% of the advertising revenues that are priced based on this model in 2011, according to the Interactive Advertising Bureau [148].

In the *pay-per-action* (PPA) model, if a click on an ad is followed by a predefined action on the advertiser's website (e.g., online purchase or registration for a newsletter), advertisers pay a *cost-per-action* (CPA) to the ad network. This model is widely used by many organizations primarily in service-based businesses, rather than by companies who sell tangible "mail order" types of products online. These service-based businesses (e.g., insurance companies, mortgage companies, real estate brokerage) are aware that customers generally do not buy these kind of services on a first impression. Therefore, these organizations using CPA media are instead generally far more interested in collecting initial, focused, targeted leads (i.e., potential sales contacts) from their advertising. As these markets are very competitive, businesses know and appreciate the fact that if they can get someone to join their e-mail list or find some other method of encouraging people to complete their online form, they would instantly have a significant head start on their competition. Therefore, they are willing to pay for CPA ads knowing that they are paying only for leads that are focused, refined and targeted for their business.

The ad network gives a fraction of the ad-generated revenue to the publisher who hosted the ad that resulted in an impression, a click or an action. These revenue models provide incentive to participate in the ad serving system: advertisers earn the revenue created by ads, ad networks earn money for storing the ads and finding proper publishers to display ads, and the publishers earn money for hosting ads and directing users towards advertised websites. Users benefit from obtaining advertisements that are tailored to their interests.

1.2 Exploits of Online Advertising Systems

Surprisingly, online advertising and Web browsing still rely on the Hypertext Transfer Protocol (HTTP), which does not provide any guarantees on the integrity or the authenticity of online content. Given the lack of security protocols, an adversary may perform ad fraud attacks to exploit the online ad serving system for its own benefit. Considering the amount of money at stake, the security of online advertising is becoming a pressing concern for advertisers, ad networks and publishers. Because online advertising has emerged as the main source of revenues for most online activities, the attacks on online advertising systems could undermine the business model of the participating stakeholders and thus could represent a concern for the future of the Internet.

Adversary

An adversary launching an attack on an advertising system can take various forms in practice. We consider a *selfish* adversary intending to take advantage of the ad serving system: A selfish adversary exploits the system with the goal of diverting part of the ad revenue for itself. In contrast, a *malicious* adversary can perform any types of attack on the ad system, typically for nefarious purposes (e.g., launching a Denial-of-Service (DoS) attack, spreading a malicious software or hurting a competitor).

The adversary can be part of the ad serving architecture or part of the *access network* that provides Internet connectivity to end users (Figure 1.1). As discussed, all entities of the ad serving architecture benefit from the delivery of ads to end users, however there are various ways in which they could try to increase their revenue. In contrast, the access network that carries all users' traffic does not receive any ad revenue. Thus, the access network might also be tempted to tamper with the transiting data to generate benefits for itself.

Depending on the amount of resources and know-how available to the adversary, it can either attempt simple attacks from a single computer or it may deploy automated mechanisms to perpetrate large scale attacks from a number of machines worldwide. Today, *botnets* are a very popular tool for perpetrating distributed attacks on the Internet and are used very often to commit ad fraud. A botnet is a collection of software robots, or *bots*, that run autonomously and automatically. Bots are typically compromised computers running software, usually installed via drive-by downloads (i.e., downloads that happen without users' knowledge or consent) exploiting Web browser vulnerabilities, worms, Trojan horses or backdoors, under a common command-and-control infrastructure. A *bot master* controls the botnet remotely, usually through a covert channel (e.g., Internet Relay Chat) for the botnet to be stealth and to protect against detection or intrusion into the botnet network. An adversary wanting to use a botnet for ad fraud could build its own or rent an existing botnet from another botnet master.

Although botnets typically enslave PCs to act like zombies in a botnet, a (believed to be the first) botnet of compromised wireless routers was detected in 2009 [24]. The botnet was used to launch a Distributed Denial-of-Service (DDoS) attack on DroneBL, a distributed Domain Name System (DNS) Blacklist service. It was estimated that the botnet gained control of approximately one hundred thousand routers, targeting home routers that have Web interface and an SSH port directly accessible without requiring a password or with a weak username and password combinations. This problem was later solved with a firmware update. Once it gained access to the system, the botnet loaded a file that turned routers into bots. This example demonstrates that the botnet problem is not something that only affects PCs. An adversary can use *warkitting attacks* to subvert home wireless routers [214]. Warkitting refers to a drive-by subversion of wireless home routers through unauthorized access by mobile Wi-Fi clients. It is shown that in practice an adversary can perform warkitting with low-cost equipment and that a large number of routers are susceptible to such attacks.

A botnet of wireless routers can perpetrate powerful *man-in-the-middle* attacks, as routers are in a position to *eavesdrop*, *alter*, *inject* and *delete* communications. It also has the advantage of having the bots almost always connected to the Internet (compared to the typical end-user machine that is connected to the Internet only from time to time). In addition, it is more difficult to detect that a device has been compromised, due to the lack of security software for such devices (e.g., no anti-virus software).

Ad Fraud

An online advertising system can be abused in many ways. We first survey the ad fraud attacks that have been the most prevalent in practice and that yield monetary benefits for the adversary: click fraud, malvertising and adware. Next, we focus on a novel attack based on inflight modification of ad traffic. We also address possible countermeasures to these attacks.

1.2.1 Click Fraud

In each of the revenue models (i.e., impression-based, click-based and action-based) an advertiser who pays for his ads to be included in online content has a positive return on investment (ROI) only when genuine impressions, clicks and actions are generated by legitimate users. ROI is used to express the actual or perceived future value of a marketing campaign and is calculated as the ratio of the revenue gained or lost, relative to the initial investment. An adversary can simulate interest in ads (by creating illegitimate ad impressions, clicks or conversions in the corresponding revenue models) that provides advertisers with little or no ROI, because they are not a result of legitimate users being exposed to ads. We refer to this type of ad fraud as *click fraud*.

The two most occurring types of click fraud are *publisher click inflation* and *advertiser competitor clicking*.

With a **publisher click inflation attack**, a publisher tries to over-report its contribution in exposing users to ads. As publishers are rewarded by ad networks proportionally to the number of impressions or user-generated clicks and actions on the ads included in the publisher's Web pages, publishers sometimes inflate the numbers in order to obtain more revenue from ad networks. To do so, they generate fraudulent impressions, clicks and actions for which advertisers are charged by ad networks, and the fraudulent publishers receive a share of that revenue.

With an **advertiser competitor clicking attack**, an advertiser tries to undermine the advertising campaigns of its competitors. In order to increase the visibility of its own advertisements, an advertiser can create artificial impressions, clicks or actions on advertisements of its competitors. If its competitor advertisers are charged for these, their daily budgets can be exhausted rapidly and the fraudulent advertiser's ads would have the advantage of being selected and served to legitimate users.

Depending on the revenue model, an adversary generates artificial interest in ads as follows:

- In the pay-per-impression model an adversary generates fraudulent ad impressions by issuing HTTP requests for Web pages containing ads that users never see.
- In the pay-per-click model an adversary generates fraudulent clicks on ads by issuing HTTP requests for ad impression URLs, that were not generated by legitimate users.
- In the pay-per-action model an adversary can produce fraudulent click-actions by issuing HTTP requests that represent an advertiser-defined action, such as a subscription, in order to simulate the action of a legitimate user.

Fraudsters can generate ad fraud themselves or deploy a third-party or automated programs to do so. Automated ad fraud attacks very often rely on botnets. An example of a botnet click fraud in the PPC model is `Clickbot.A`, the botnet that executed a low-noise click fraud attack against syndicated search engines and was investigated in detail by Google [110].

The botnet consisted of over one hundred thousand compromised machines and it perpetrated a publisher click inflation ad fraud. The bot operator acted as a publisher and created several websites that contained links that eventually led to ads on which the clickbot would click.

Automated ad fraud attacks can also be executed without compromising the end users' machines. For example, in the PPC model an attacker can launch a stealthy, automated click-fraud attack called *badvertisement* where fraudulent clicks are generated on ads hosted by the attacker [128]. The goal is accomplished by corrupting the JavaScript required to properly include ads into Web pages and does not depend on any client-side vulnerability. The script causes an ad to be automatically clicked and processed by a client's Web browser. Consequently, the click is accounted for by the ad network, the advertiser is charged and part of the revenue is transferred to the fraudulent publisher. Badvertisement attack is also an example of a publisher click inflation ad fraud.

An attacker can also generate fraudulent clicks by tricking users with *clickjacking* attacks to click on ads. Clickjacking happens when the attacker uses multiple transparent layers of Web pages to trick a user into clicking on a button or a link on a hidden page when they were intending to click on the bottom visible page [193]. Therefore, the attacker can trick users into performing actions that the users never intended and thus "hijack" their clicks. The clicks can then be turned into fraudulent clicks on CPC ads. Figure 1.2 shows an example of a clickjacking attack where a victim surfs the bottom page (a fraudulent site that launches the clickjacking attack, e.g., `myphotos.com`), while actually affecting the site in the top frame (e.g., Google search result page) that the victim does not see. In the example, we have made the top page partially transparent for the purpose of illustration, whereas in the actual attack the top page is invisible to the victim. When the victim clicks on the button "`Next`" to proceed to the following photo on the Web page of `myphotos.com`, the click is hijacked and turned into a click on one of the CPC ads positioned on the right side of the Google search result page. In order to generate profit from clickjacking attacks, the fraudster can load his own website in the top frame (instead of Google search results as in the example) and turn hijacked clicks into clicks on CPC ads that appear on the fraudster's Web pages (i.e., perform a publisher click inflation attack). Alternatively, the fraudster can load Web pages on which its competitors' ads appear and generate fraudulent clicks on these (i.e., perform an advertiser competitor clicking attack). Clickjacking is possible because of Web browser vulnerabilities and more details about countermeasures can be found in [193].

Fraudulent clicks have a negative effect on advertisers' returns on investment and ideally, the ad network would detect all of the fraudulent clicks, mark them as *invalid* and not charge advertisers for those clicks. To avoid the detection and ensure the revenue, the fraudulent clicks should be indistinguishable from the legitimate ones generated by users such that ad networks charge advertisers and share the revenue with publishers. That is why the fraudsters try to generate behavior patterns that resemble the behavior of legitimate users. Consequently, it is not possible to know with an absolute certainty whether a click is fraudulent or legitimate. Therefore, in order to preserve a good user experience even when a click is marked invalid, the user agent is still redirected to advertiser's website.

Estimates of the extent of the click fraud vary widely, and this is a subject of much discussion among advertisers and PPC search engines. According to Adometry (formerly, Click Forensics, Inc.), a company that performs ad traffic quality control, the click fraud rate has been on the rise for years, reaching the maximum 22.3% of the clicks being fraudulent in the third quarter of 2010 [38]. The rate has then declined in the fourth quarter of 2010 to 19.1%. However, Adometry's CEO says that this trend might not last: While the overall

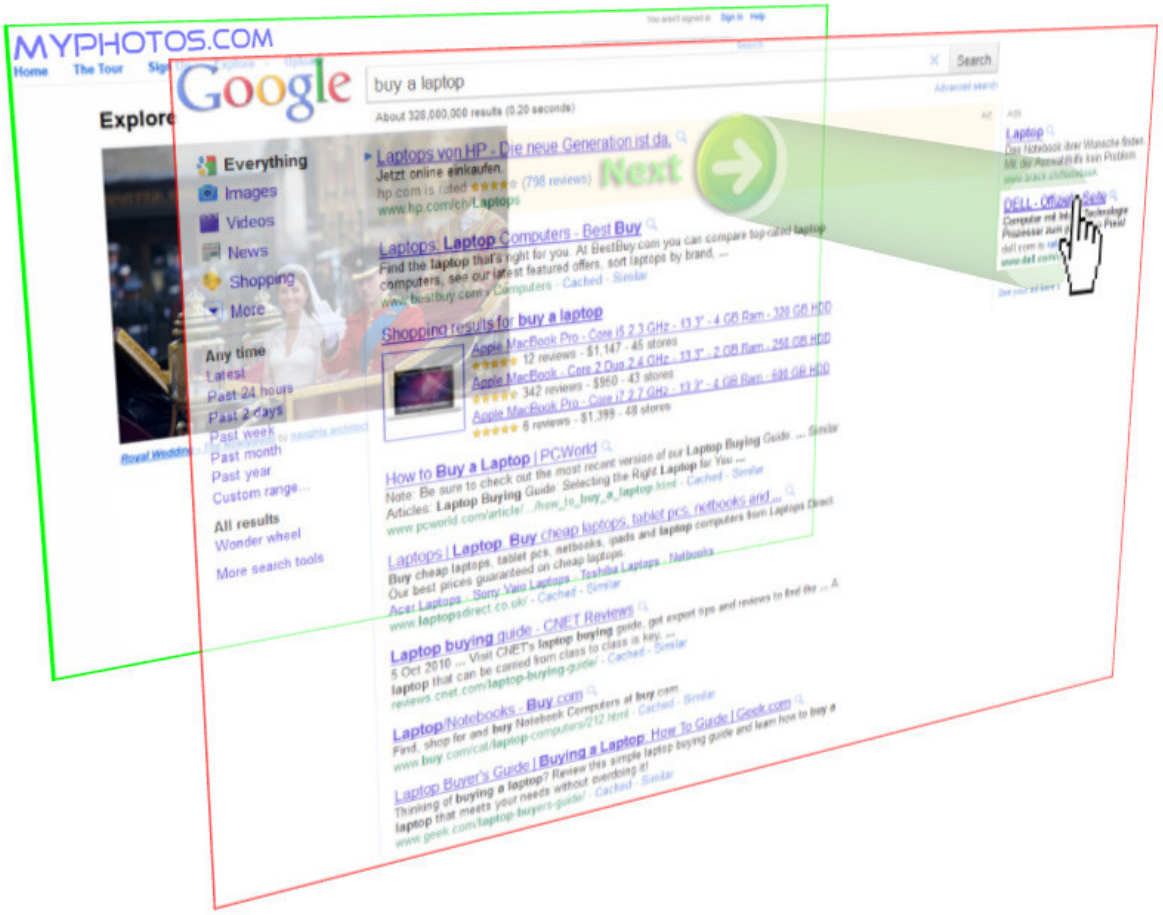


Figure 1.2: Clickjacking attack. In a clickjacking attack, a victim browses a Web page (in this example `myphotos.com` in the bottom frame) that loads an invisible top frame (in this case a Google search result page) and tricks the victim into clicking on the bottom frame while actually affecting the site in the top frame. We have made the top frame partially transparent for the purpose of illustration, whereas in the actual attack the top page is invisible to users. When the victim clicks on the button "`Next`" to proceed to the following photo on the `myphotos.com` page, the click is hijacked and turned into a click on a CPC ad on the invisible Google search result page.

click fraud rate dropped for PPC advertising, new schemes focused on display advertisements have emerged [38].

We next present a case study of an ad fraud scheme that targets websites with display advertisements.

Case Study: Advertisers Scammed by Porn Sites

A case of click fraud reported in 2011 is a very good example of how a fraudster can orchestrate a large scale automated attack in a way that is difficult for ad networks to distinguish fraudulent clicks from legitimate clicks, thus producing high revenue for the fraudster.

The overview of the scheme is presented in Figure 1.3. The fraudster hosts a website ac-

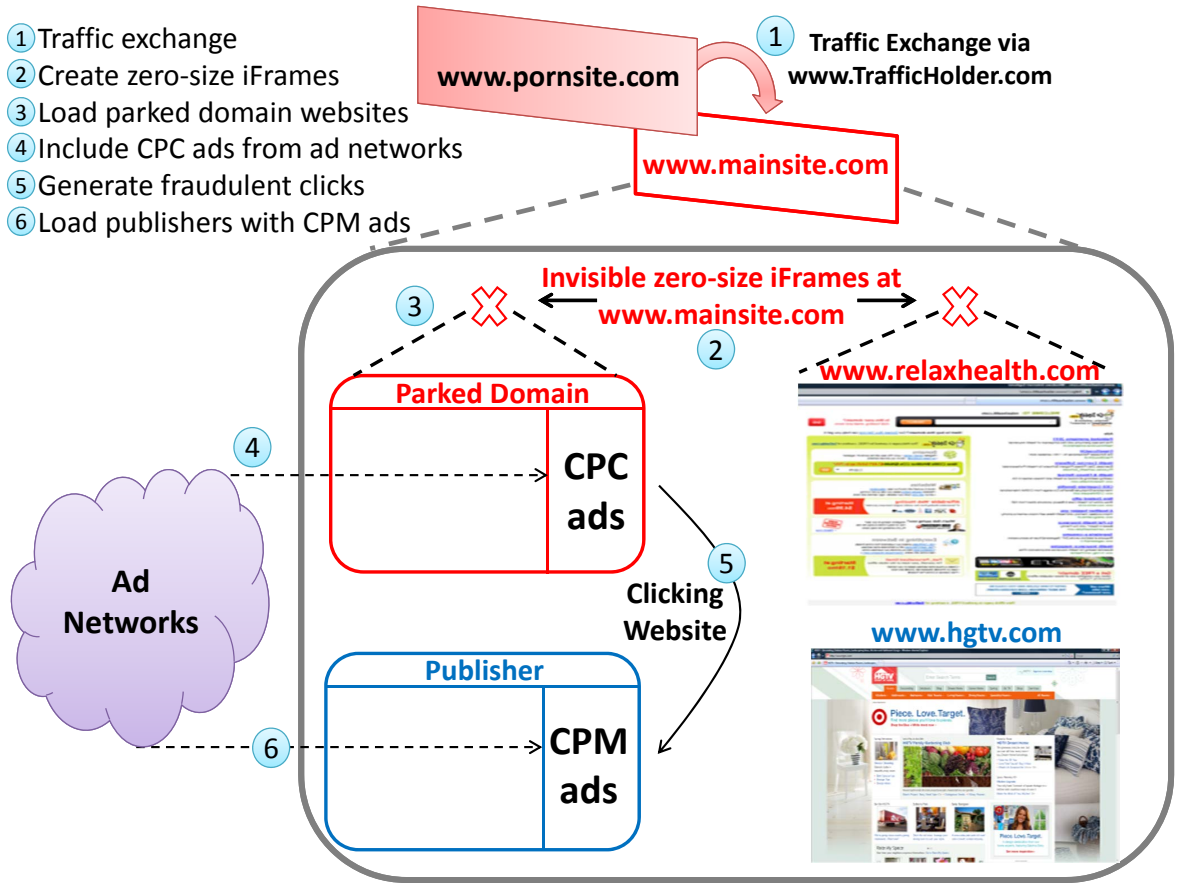


Figure 1.3: Schema of a click fraud attack. A fraudster buys legitimate traffic from a pornographic website in order to generate traffic at its own website (step 1) and produce legitimate-looking click traffic on ads. The fraudster's website creates a number of invisible iFrames (step 2) that load parked domain websites (step 3) owned by the fraudster and hosting CPC ads (step 4). In collaboration with clicking websites, the parked domain websites produce fraudulent clicks on their CPC ads (step 5). Fraudulent clicks result in loading of legitimate big-name-brand publishers with their own CPM ads (step 6). Reputable advertisers that pay for their ads to appear on quality publishers' websites have their ads appear within pornographic websites, which enables AdSafe to detect the fraud.

cessible at `www.mainsite.com` that contains links to pornographic video websites. In order to attract visitors to its website, the fraudster participates in the traffic exchange with a popular pornographic website (e.g., `www.pornsite.com`). In this traffic exchange, `www.pornsite.com` sends traffic of its legitimate visitors to `www.mainsite.com` for monetary remuneration. The traffic exchange is made possible by a man-in-the-middle website (`TrafficHolder.com`) that provides a catalogue of the traffic and corresponding prices that one can buy. The scheme then executes as follows:

1. To implement the traffic exchange, when legitimate users visit `www.pornsite.com`, the site opens a pop-under window and loads the fraudster's website `www.mainsite.com`, which generates traffic at the fraudster's site. According to the agreement, `www.pornsite.com`

in return receives money from the fraudster. By cooperating in this way with popular websites, the fraudster is able to obtain millions of unique visitors for its own website.

2. When `www.mainsite.com` loads in the pop-under window, it generates a number of invisible zero-sized (i.e., 0x0 pixel) iFrames.

3. Each of the iFrames will load one of the *parked domain* websites registered by the fraudster. Parked domain websites are single page websites that typically do not have any content on these domains. These domains might be reserved for future development or to protect against the possibility of cybersquatting, i.e., registration of Internet domain names that contain trademarks with no intention of creating a legitimate website, but instead of selling the domain name to the trademark owner. Domain parked websites typically display advertisements and thus generate revenue for the registrant. In this scheme, the parked domains loaded in invisible iFrames are all registered by the fraudster and they all include advertisements. The domain names do not seem suspicious and are not related to pornographic websites (e.g., `www.relaxhealth.com` or `style-andmore.net`). This is important as most of the ad networks are not likely to include ads in pornographic websites.

4. Parked domain sites with corresponding advertisements are loaded in invisible iFrames.

5. A number of clicks on ads occur. To generate the clicks, the fraudster can simply deploy one of the "clicking websites" that already have such techniques.

6. The fraudulent clicks on the ads in the parked domains will eventually result in loading one of the big-brand-name publishers (e.g., HGTV) with its own CPM advertisements.

How does this scheme actually generate money for the fraudster? The monetization scheme is represented in Figure 1.4. It is important to note that the big-brand-name publishers have a dual role, acting as (i) *CPC advertisers*, paying ad networks to include CPC ads with links to their websites into online content and (ii) *publishers*, collaborating with ad networks to host ads of CPM advertisers.

1. The fraudster generates fraudulent clicks on CPC ads of big-brand-name publishers that appear on his parked domain websites.

2. Ad networks charge the big-brand-name publishers (now playing a role of CPC advertisers) for the corresponding fraudulent clicks.

3. Ad networks pay a percentage of the CPC revenue to the registrant of the parked domains where the fraudulent clicks occur, i.e., to the fraudster.

4. By receiving traffic from the parked domains, ad impressions on big-brand-name publishers' Web pages are generated. For these ad impressions the publishers (now acting indeed as publishers hosting the ads) will obtain the revenue themselves. For this reason the fraudster cleverly targets only big-brand-name publishers that sell pay-per-impression and video ads and do not measure conversions, because for them only impressions count and any traffic is good. If the fraudster tries to load an e-commerce site that actually checks the quality of the traffic, this scheme would be detected. In addition, one more reason not to be suspicious is that the traffic towards publishers originates from legitimately-sounding domain names.

5. Ad networks charge the CPM advertisers for the impressions generated on big-brand-name publishers' websites.

6. Ad networks share the CPM revenue with the big-brand-name publishers.

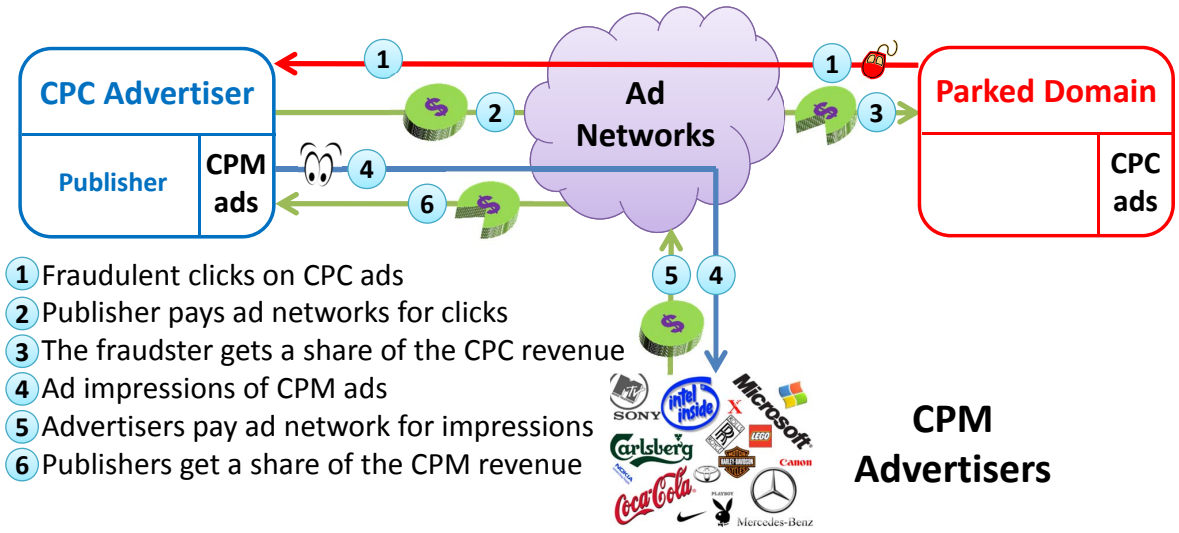


Figure 1.4: Monetization of the attack. The fraudster generates fraudulent clicks on CPC ads on his parked domain websites (step 1), for which the ad network charges the big-brand-name publishers (step 2) and shares part of the profit with the fraudster (step 3). The publishers are willing to pay as these clicks result in traffic on their websites that creates ad impressions (step 4) for which ad networks charge CPM advertisers (step 5) and share the profit with the publishers (step 6). In this scheme, the fraudster, the publishers and the ad networks make profit whereas the CPM advertisers lose revenue.

The ones that are hurt the most by this scheme are big-brand advertisers whose ads normally appear on reputable publishers' websites. The publishers and ad networks earn revenue by serving and displaying CPM ads, thus they are not concerned with the scheme. Therefore, big-brand advertisers are the ones who should fight the fraud. However, they do not want their reputation to be damaged by being associated with the fraud and in addition, the scheme does not target a single party, it rather distributes the damage across a number of advertisers such that each individual does not have much incentive to fight the fraud itself.

This type of fraud has been detected by AdSafe, a company that ensures that brand advertisements appear with an appropriate content. Loading big-brand-name publishers' content and ads within the iFrames of the fraudster's `www.mainsite.com` has triggered an alarm at AdSafe as advertisements of reputable brands have appeared within frames of the pornographic website. As the fraud is based on the traffic of the legitimate users who visit `www.pornsite.com`, the click patterns appear as genuine (having different IP addresses, different Web browsers and at different times of the day) and are difficult to distinguish from legitimate clicks. Ad networks have a hard time detecting these clicks as fraudulent, because these clicks do not follow typical bot-generated click patterns.

A back-of-the-envelope calculation [149] shows that the fraudster might have earned between $50K to $700K per month with this scheme. Given that the scheme has been running

for eight months, in total the fraudster might have earned $400K to $5M. This proves that a moderately sophisticated ad fraud scheme can result in substantial monetary gain. In addition, the fraudster does not violate law in most of the cases, but only terms of service of online advertising networks. Therefore, legal repercussions might not be sufficient to deter the fraudster from committing ad fraud, given the revenue at stake.

Countermeasures to Fight Click Fraud

As ad networks charge advertisers based on a number of impressions or clicks on advertisements, it might be counterintuitive for ad networks to have incentive to fight click fraud. In the short term, ad networks indeed earn more revenue by not filtering out fraudulent clicks. However, in the long run, the bad quality of the ad traffic could affect the reputation of the ad networks and result in poor performance of advertising campaigns, thus advertisers might stop investing in this form of advertising and publishers might not want to host the ad networks' ads within their content. Also, users might perceive ads as useless. Basically, if fraudulent clicks are not filtered out, an entire system could be ruined. An economic analysis [95], based on a game-theoretic model of the online advertising market, shows that ad networks that deploy effective countermeasures against click fraud gain a significant competitive advantage, as both publishers and advertisers will choose ad networks that offer the best return on investment.

The goal of the countermeasures deployed by ad networks is to make a successful attack more difficult or more costly for an attacker, rather than to absolutely eliminate click fraud. Most of the ad networks' techniques are kept confidential, otherwise it would be easy for the attacker to avoid detection. Typically, ad networks look for signals that indicate fraudulent click activity. Those signals could be different characteristics of HTTP traffic, anomalies in the ad click and conversion traffic, browser and user behavior that deviates from the expected behavior. Some techniques can be deployed to prevent click fraud as well, such as setting up a trust boundary between a publisher's content and ad slots on the publisher's Web pages. For example, assume that a publisher embeds a script into the content of his Web pages with a purpose of generating fraudulent clicks on ads that appear on these pages. If an ad network includes ads dynamically in the content in a way that the browser does not allow any script on other parts of the Web page to access the ads (e.g., by including ads in an iFrame), it may prevent a potential publisher click inflation attack.

In the case suspicious activities are noticed, ad networks may set an ad traffic monitoring team to investigate and potentially terminate collaboration with publishers on whose pages a lot of fraudulent clicks occur. Trusted third-party companies are employed to verify the practices of ad networks in examining ad traffic. Such companies are independent from ad networks and advertisers, and their job is to make sure that the clicks are properly labeled as legitimate or invalid, thus assuring advertisers that they are justifiably charged for the clicks.

1.2.2 Malvertising: Spreading Malware via Ads

Malvertising, one of the fastest growing security threats on the Web, is a class of online ads that attempt to infect an ad viewer's computer. It is particularly scary, because any site hosting ads and any operating system could be a potential target. Moreover, users do not even have to click on ads to trigger malware. For example, according to the report published by *Blue Coat's* research lab [166], an ad server can serve a JavaScript that, instead

of fetching the legitimate ads, injects a hidden iFrame tag into the original Web page. The iFrame instructs the victim's browser to silently communicate with a malware server in the background, eventually resulting in the download of a PDF exploit file.

An advertiser can launch a malvertising attack, by adding its ad to a legitimate ad network. The ad network embeds the ad in publishers' websites and users click on it eventually. Publishers can also embed malvertisements into the content of their Web pages to direct a user to the malicious website and install malwares.

Most of malvertisements are hosted by so-called *remnant advertising networks*. These networks sell empty advertising slots at the last opportunity. They aggregate advertisements and charge low rates. Consequently, there is less revenue and possibly less caution over the quality of advertisements.

Malvertisements can even appear at well-known websites, such as *New York Times* (reported on September 14, 2009 [155]), *Facebook* (reported on April 12th, 2010 [33]) and *London Stock Exchange* (reported on March 1, 2011 [147]). For example, visitors of the London Stock Exchange's website were exposed to malicious ads, that were designed to pop up fake security messages on their computers in order to sell anti-virus software.

Figure 1.5: Malvertisement promoting the latest version of Adobe Flash Player was embedded in Microsoft's search engine Bing. Bing included the malvertisement as one of the sponsored search results for the keyword search "`adobe flash player`". The malvertisement thus appeared in a colored box that marks sponsored links on the top of the results page. We single out the malvertisement with the red rectangle and a danger symbol. Web browsers cannot distinguish malvertisements from legitimate links and warn the users.

Figure 1.5 shows a malvertisement that was embedded in Microsoft's search engine Bing (reported on July 3, 2010 by `StopMalvertising.com` [34]). The ad appears among the sponsored results and it refers to Macromedia Flash, while it points to `Flash.Player-Pro-Download.com` that does not belong to Adobe. Users who click on the ad go through

`rc12.overture.com` and from there browsers are redirected to `player-pro-download.com`. This looks like a clear and nice website, but no mention of Adobe anymore. Instead there are promotions for online Flash games and professional Flash tutorials. If a user tries to download such software, browsers may issue security warnings because the content is not signed with a valid security certificate.

According to Dasient (an Internet security company that protects businesses from losses of traffic, reputation, and revenue caused by Web-based malware attacks) in the last three months of 2010 attackers managed to serve three million malvertisement impressions every day. In another study, it was identified that about 2% of malicious websites were distributing malware through advertisements, based on an analysis of about 2,000 known advertising networks [185].

Countermeasures to Fight Malvertising

Appropriate and regular checks of advertisements are the best way to avoid malvertising. The publishers and ad networks should perform regular checks to verify the advertising content providers for any kind of active or malicious code. If they detect any unexpected or unwanted behaviour such as automated redirections, they should not publish the ads to the end users. In June 2009, Google launched a new search engine called *investigative research engine*, publicly available at `www.Anti-Malvertising.com`. This is to help ad network partners, identify potential providers of malvertising. The Internet users should also install and update appropriate anti-malware softwares on their machines to minimize the risk.

1.2.3 Adware: Unsolicited Software Ads

The term adware refers to any software that displays advertisements without users' permission [102, 197]. They are often designed to present advertisements according to the websites users visit. Adwares are produced by advertisers or by publishers of free software. Accordingly, adwares can broadly be divided into two main groups.

The first group of adwares are published for users who do not wish to pay for certain software. Many programs, games or utilities are ad-supported and distributed as adware (or freeware). If users purchase a registration key, they can disable displays of ads. The ads should also disappear as soon as the user uninstalls the software. In this case, adware is usually seen by the developer as a way to recover development costs, and in some cases it may allow for the software to be provided to users free of charge or at a reduced price. The income derived from presenting advertisements to users may allow or motivate the developer to continue to develop, maintain and upgrade the software product. As an example, the *Eudora* mail client displays advertisements as an alternative to shareware registration fees.

The second group of adware can be described as a form of *spyware* that collects information about users in order to display advertisements in Web browsers. In other words, it displays advertisements related to the data it collects by spying on users. When adware becomes intrusive like this, it can be categorized as *spyware* and users should avoid it for privacy and security reasons. In this case, adware can intercept all information that users enter via the Web, add unauthorized sites to desktops and Internet favorites, track and monitor browser activity or attach the unwanted toolbars and searchbars to browsers without users' knowledge or approval. Moreover, the personal information can be sold to other parties without users' knowledge or consent. Finally, adware can hijack the default homepage and settings so the

user cannot change them.

As an example, *YapBrowser* is an adware (spyware) that served unsolicited, aggressive advertisements, redirected users to undesirable websites, and modified essential system settings. This product was designed to be illegally installed on users' computers in order to make profit for spyware and adware creators. It must be noted that YapBrowser was bundled with the *Zango* software, a software company that provided users access to its partners' videos, games, tools and utilities in exchange for viewing targeted advertisements on their computers. In June 2006, YapBrowser was acquired by UK's *SearchWebMe*. *SearchWebMe* assures that the new *YapBrowser* download does not contain any adware or harmful applications. *Gator Software* from *Claria Corporation* and *Exact Advertising's BargainBuddy* are two other famous adwares in this category.

Countermeasures to Fight Adware

Users should avoid visiting untrusted websites because they are mainly delivering adware and spyware to unsuspecting users. Moreover, they can also install and update regularly anti-adware softwares. Finally, they should carefully read the terms of use for free software as they potentially install the adware as well. Note that it is required by law to state whether or not software has adware bundled with it.

1.3 Inflight Modification of Ad Traffic

We have identified a novel type of ad fraud, consisting in the inflight modification of the ad traffic itself. Over the course of this thesis, several instances of this type of ad fraud have appeared in practice. A prominent example is the *Bahama botnet*, in which malware causes infected systems to display to end users altered ads, as well as altered search results (e.g., Google) [20]. The difference, compared to the traditional click fraud where ad networks could even earn revenue from fraudulent clicks, is that the traffic and the revenue is diverted from ad networks.

In the case of the Bahama botnet, compromised machines take their users to a fake page that looks just like the real Google search page and even returns results for queries entered into its search box. The attacker redirect users' traffic to a fake website by corrupting the DNS translation method on the infected machines. As a result, the domain name `Google.com` is translated to an IP address that is not owned by Google, but by an attacker. When a user enters a query into the search box on what he believes is a Google server, the traffic actually goes to the fraudulent server that pulls the search results for the given query from Google, meddles with them (notably, it turns organic search results into paid links) and sends the results back to the user. Consequently, the results displayed are different from what they would otherwise be. A click on an "organic" link (in this case actually a masked CPC ad) will result in a paid click through several ad networks or parked domains. Advertisers will be charged and the click fraud has occurred. Essentially, the Bahama botnet diverts the traffic and the revenue from major ad networks (e.g., Google) and redirects it to smaller ad networks and publishers.

Instead of compromising the users' machines, an attacker can also rely on botnets of compromised wireless routers [24]. Once a wireless router is infected with a malware and turned into a bot, the botnet master can instruct the bot to perform inflight modifications of the traffic that passes through the router. Many public hotspots rely on a similar business

model: providing free Internet access to users and in return generate revenue by embedding ads into the users' traffic [68].

There are reports of similar behavior of some Internet Service Providers (ISPs) [188, 238]. TrendWatch, the malware research team of Web security company TrendMicro, has investigated the practices of an Estonian ISP that was replacing ads included in the Web pages users were browsing [213]. The ISP was in charge of a number of DNS servers and was redirecting ad traffic from legitimate ad servers (e.g., Google ad network) to the servers of its choice. Consequently, users received ads that websites did not intend to show to their visitors. The investigation shows that thousands of ads were replaced per day, which implies that a significant ad revenue was diverted from legitimate victim ad networks.

Besides undermining the business model of ad networks, inflight modification of ad traffic could also negatively affect the security of end users (malvertisement could be included instead of legitimate ads), the reputation of websites and the revenue of legitimate advertisers.

1.3.1 Inflight Attacks on Ad Traffic

We focus on the inflight attacks on ad traffic that can be perpetrated by a selfish adversary located in the access network, as described in Section 1.2. Such an adversary is in a favorable position to implement the inflight attacks. In addition, it has a significant economic incentive to exploit the online ad serving in order to divert part of the revenues because it is the only entity that does not benefit in the traditional ad serving model. We describe various inflight attacks that an adversary can perform. In general, they are based on *injecting* or *deleting* advertisements.

Injecting Advertisements

An adversary can inject ads in Web pages by either *adding* new advertisements or *replacing* already embedded ads. By injecting ads, the adversary thus bypasses the traditional ad serving model. The attack is successful if the adversary can obtain revenue with the injected advertisements. The achievable revenue depends on users seeing the advertisement (PPI model) or finding an ad interesting and taking an action, e.g., clicking on it (PPC model) or subscribing to a newsletter (PPA model). To maximize the success of the attack, the adversary should thus increase the *visibility* of injected ads and *target* them to users' interests and the content of the corresponding Web pages. We present two types of injections of ads: *pollution attack* and *targeted attack*.

Pollution Attack We call a *pollution attack* the injection of advertisements not necessarily targeted to the Web page's content or users' interests. Rogers, a Canadian ISP, was reported to add content, notably advertisement for their own services, into any Web page that traversed their access network [229]. This was done by injecting into Web pages a single line of code that causes the user to fetch and execute a JavaScript as if it was part of the content of the Web page. Pollution attack is therefore trivial to implement and does not require a lot of resources and yet it generates revenue for the adversary. This attack might be particularly effective for the purposes of brand advertising. However, it could also spoil the appearance of Web pages and thus might harm the reputation of websites.

Targeted Attack A more sophisticated version of the attack consists in injecting ads *targeted* to users' interest and the content of Web pages. For example, an adversary can add highly targeted and visible ads into search engine results [20]. Search engines facilitate targeted advertising as search queries indicate users' interest at the considered moment. In addition, surveys have shown that more than half of users click on one of the first two organic (i.e., non-sponsored) results of Search Engine Result Pages (SERPs) [11]. Knowing this, the adversary can inject its ads at the top of SERPs, resulting in a substantial increase of users' traffic on a website of the adversary's choice. It could also commercialize such services to advertisers.

Similarly, the adversary could inject ads into Location-Based Services (LBSs) results. LBSs provide *points of interests* (POIs) near users' locations. LBS results are typically presented on a map (e.g., Google maps) to help users locate POIs. Usually, LBS include ads in their results that are targeted to users' locations and interests. Knowing this, the adversary can inject its ads at the top of the list of POIs. Hence, when a user queries an LBS for a POI in his vicinity, the adversary can influence the user's choice.

In practice, several ISPs already work with advertisers to legally add ads to users' Web traffic. For example, Phorm [76], is a personalization technology company that offers an ad serving platform to ISPs. It is currently engaged in market trials ISPs in the UK (e.g., Virgin Media), Brasil, Romania, etc. Other examples are "free" ISPs [64] and 3G data services [78] whose business model is based on providing free Internet access and generating revenue from injected ads.

Deleting Advertisements

An adversary can also remove ads from Web pages. For example, an ISP can automatically filter out all the ads and offer this as a service to its customers. Blocking ads is already possible at the end users [53], but doing it network-wide would be a transparent and more efficient solution. Also, the adversary could have an agreement with certain advertisers or ad networks to filter out their competitors' ads. In practice, the infrastructure to block specific HTTP content is in place.

1.3.2 Economic Impact of Inflight Modification of Ad Traffic

The consequence of inflight modification of the ad traffic is that when users click on altered ads they generate revenue for the attacker instead of the legitimate ad network. Thus, the modification of the ads undermines the business model of ad networks. We take a *bottom-up* approach to assess the potential revenue of an adversary modifying ad traffic on-the-fly: we model the browsing behavior of users, estimate the number of ads affected by attacks and derive the ad revenue at stake. Notation of symbols used in the computation is presented in Table 1.1.

Users' browsing behavior is profiled by Web analytic companies, such as *Compete.com*. We base our analysis on measurement data we obtained from *Compete.com* about the number of page views and the number of unique visitors on each of the 1000 most popular websites in 2009 (Figure 1.6). The exposure of users to online ads has been evaluated extensively in [161] showing that $h = 58\%$ of the top websites host ads and that on average there are $a = 8$ ads per page. We estimate the total number of ads users see on the the top 1000 websites

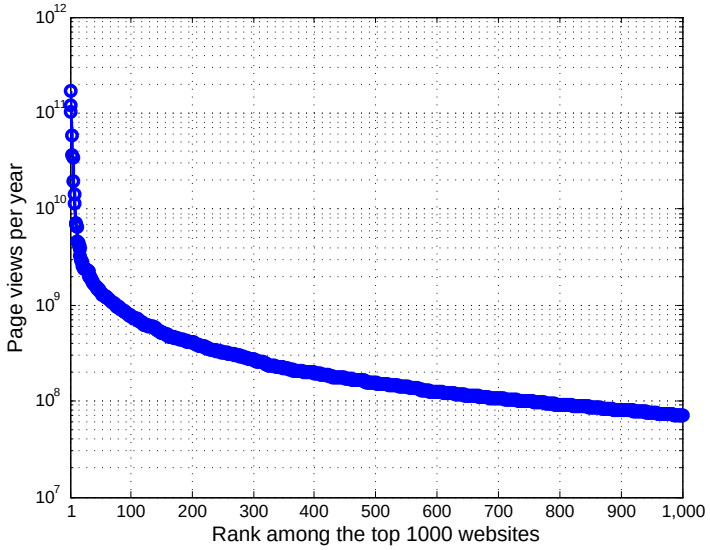


Figure 1.6: Popularity of the top 1000 websites based on page views per year.

Table 1.1: Symbols used for calculating the economic impact.

Symbol	Definition
h	Percentage of top websites hosting ads.
a	Average number of ads per Web page.
I_t	Total number of ads seen on the top 1000 websites over a year.
I_u	Average number of ads a user sees on the top 1000 websites over a year.
ϕ	Fraction of ads charged for based on impressions.
p	Probability of a clickthrough on an ad.
CPC	Cost-per-click.
CPI	Cost-per-impression.
R	Potential annual ad revenue generated per user.
$R_{\mathcal{A}}$	Adversary's potential annual revenue gain.
α	Number of users affected by the attack.
β	The fraction of the ad traffic that is modified.

during a year (I_t) as done in previous work [23, 221]:

$$I_t = \sum_{i=1}^{1000} (\text{Page views on website } i) \cdot h \cdot a. \tag{1.1}$$

The data from *Compete.com* is aggregated over all the visitors of websites and does not give individual user browsing profiles. Thus, the average number of ads I_u a user sees on the the top 1000 websites during a year is:

$$I_u = \sum_{i=1}^{1000} \frac{\text{Page views on website } i}{\#\text{ of visitors of website } i} \cdot h \cdot a. \tag{1.2}$$

We now compute the potential annual ad revenue R generated per user. To do so, we take into account that for a fraction ϕ of ads the ad network charges advertisers based on the number of *impressions* and for the remaining $1-\phi$ based on *performance* (e.g., clickthroughs) [148]:

$$R = \phi \cdot I_u \cdot CPI + (1-\phi) \cdot I_u \cdot p \cdot CPC. \tag{1.3}$$

where CPI is the cost-per-impression, CPC is the cost-per-click and p is the probability that a click occurs on an ad (i.e., clickthrough rate). Due to the large number of ads, the cost-per-mille (CPM) representation is usually preferred for impression based ads ($CPM = CPI \cdot 1000$). Both CPM and CPC depend on the type of ads and the hosting website. It is difficult to obtain a complete picture of CPMs and CPCs for the online advertising space, thus we rely on the average estimates reported in practice: $CPM = \$2.39$ [23] and $CPC = \$0.5$ [26]. The probability p that a click occurs on an ad is around 0.1% [18]. The pay-per-impression pricing model accounts for $\phi = 35\%$ of ad revenues and pay-per-click for 65%, as reported in [32]. Based on expression (1.3), we estimate that the annual ad revenue generated on the top 1000 websites per user is $R = \$494$. The total ad revenue generated at the top 1000 websites is \$4.88 billion.

We differentiate between adversaries based on: (i) the number of users α the adversary can affect and (ii) the resources the adversary has to implement the attacks, which determines the fraction β of ad traffic (and consequently ad revenue) it can modify. The upper bound of estimated revenue ($R_{\mathcal{A}}$) the adversary can gain by perpetrating attacks is:

$$R_{\mathcal{A}} = \alpha \cdot \beta \cdot R. \tag{1.4}$$

This model assumes that advertisers are willing to pay to the adversary at most the same CPMs and CPCs as to the original ad network. We consider various values of α and β corresponding to different adversaries that appear in practice and derive the associated revenue gains in Table 1.2.

Table 1.2: Adversary's potential annual revenue gain.

Adversary	α	β	$R_{\mathcal{A}}$ **(in US \$)**
Home wireless AP	[1, 10]	1	[494, 4.94K]
Hot Spot AP	[10, 100]	1	[4.94K, 49.4K]
Botnet	[1K, 100K]	1	[494K, 49.4M]
WSC	[10K, 2M]	1	[4.94M, 988M]
ISP	[50K, 15M]	1	[24.7M, 7.41B]

Note that these results are obtained from a sample of users visiting the top 1000 websites exclusively, and hence cannot be trivially generalized to the entire US population. Instead, our results measure the *economic incentive* of an adversary to tamper with the traffic of the users that access the top 1000 websites.

We consider a single compromised home wireless AP, a compromised hotspot AP, a network of compromised APs [214] or a botnet [24], a wireless social community (WSC) [63] and an ISP [29]. Figure 1.7 represents the estimated revenue $R_{\mathcal{A}}$ for the entire range of values $\beta \in (0, 1]$, considering the maximal value of α of each adversary from Table 1.2. The results show that even a small subset of routers controlled by an adversary can cause a significant

loss of ad revenue for ad networks. Also, even by applying the attack on a small portion of traffic ($\beta = 0.1$), the adversary can earn a significant revenue.

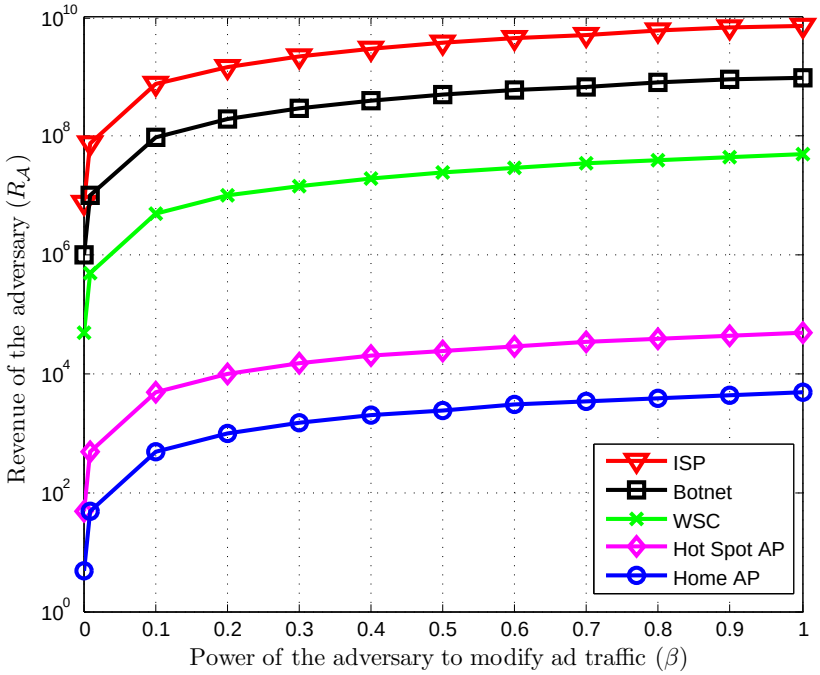


Figure 1.7: Adversary's potential annual revenue gain (in US $).

We note that ISPs have a tremendous incentive to divert even a small fraction of the ad revenue. The total ad revenue in the US in 2011 is $31.7 billion [148], meaning that the average ad revenue per day is $86.8 million. Although some ISPs would not engage in such activities due to unforeseen legal consequences or the risk of damaging their reputation, reports [12] mention that such behavior is observed in practice in some countries. In Chapter 3, we use game theory to model ISPs' economic incentives to perform inflight attacks on ad systems and show that under certain conditions diverting revenue from online advertisements can maximize the revenue of a rational ISP.

1.3.3 Implementation of Inflight Modification of Ad Traffic

To implement attacks on advertisements in practice the adversary must first identify ad objects in the HTTP traffic. A straightforward approach is to check the destination IP addresses or URLs of the requested objects. The adversary can leverage on the lists of IP addresses and domain names of the most popular ad servers, e.g., those that are used by ad blocking softwares to filter out ads [53]. If there is a match between a URL of a requested object and a URL in the list of ad servers, the adversary can classify the requested object as an ad. A powerful adversary can deploy more sophisticated technologies to identify ad traffic. For example, an adversarial ISP can use Deep Packet Inspection (DPI) technology to identify packets containing ads. DPI is a form of network packet filtering that enables automatic examination and tampering of both the header and data payload of packets. This technology enables advanced network management, data mining, application of security features, as well as eavesdropping and Internet censorship.

Once the adversary identifies ad objects, it can alter the ad traffic either: (i) *locally*, without the help of external resources or (ii) *remotely*, by redirecting users' requests towards servers chosen by the adversary. To locally alter ad traffic, the adversary relies only on the locally available resources (e.g., an access point). To remotely alter ad traffic, the adversary can for example redirect the ad traffic to another ad server by modifying URLs of objects referenced in ad frames. When a user fetches a Web page, the adversary modifies on-the-fly the payload of packets carrying the URLs of the ad servers. Hence, ads are fetched from different ad servers.

1.3.4 Proof-of-Concept Implementation on Wi-Fi Routers

In order to test the feasibility and efficiency of the attacks, we implemented them on a resource constrained device, notably a wireless router. The goal is to verify whether the attacks can be executed locally on the wireless router in a transparent way to users. If this proof-of-concept implementation is successful, then a more powerful adversary, controlling more resources, can successfully implement the inflight attacks as well.

We used an Asus WL-500G Premium wireless router with 32Mb of memory and a 266Mhz processor. We uploaded an OpenWRT [73] firmware on the router as it provides many customization features. We used the latest compatible OpenWRT version, the Kamikaaze 8.09 with kernel 2.6.27.

The attacks rely on two main components: (i) a transparent proxy (Squid v2.6) to parse HTTP traffic and (ii) executables to implement the attacks. This setting ensures that the URL appearing in the address bar of a user's browser does not change due to the attack. It is a necessary requirement for the attacks to be transparent to users. We use the built-in Busybox HTTPD server with PHP to generate Web pages dynamically and forward them to the users. Squid is also a caching proxy but we disabled the caching feature for the purposes of this implementation.
Each of the attacks consist of the following three steps (Figure 1.8):

1. User generated HTTP traffic on port 80 is intercepted and sent to the transparent proxy (Squid) running on port 3128. The interception is done using Network Address Translation (NAT) with a simple pre-routing rule.

2. The proxy calls a `C` program called `redirector.c` that analyzes all requested URLs.

3. The redirector program detects matches with predefined rules (e.g., a request to an ad server). If there is a match, the redirector program executes the corresponding PHP script implementing one attack depending on the matched rule. If there is no match, the redirector program outputs the original link and the proxy serves the original Web page.

Using this setting, we implemented three different attacks: pollution attack, injecting advertisements and targeted injection.

Pollution Attack We implemented the pollution attack using a similar script as in [229]. We inject a JavaScript into every Web page (i.e., altering the communication in step 2, Figure 1.1) which results in an HTML frame being created. In our implementation, the HTML frame shows an EPFL logo.

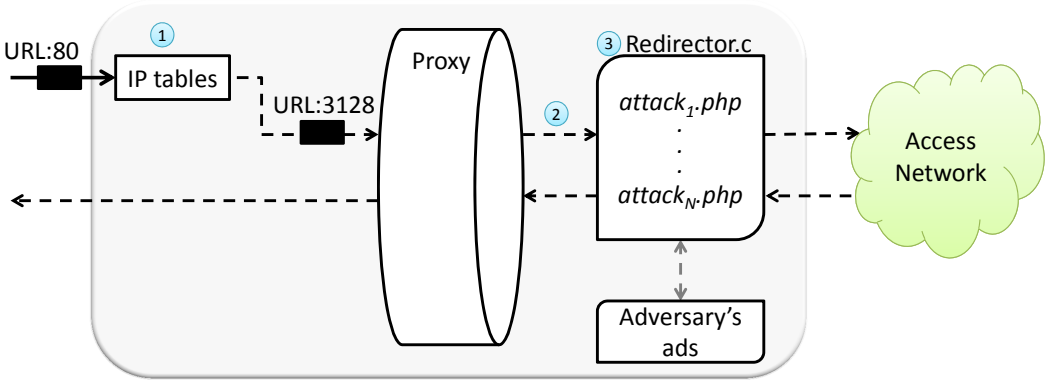


Figure 1.8: Representation of the inflight attacks implementation on a Wi-Fi router: Users' traffic on port 80 is re-routed to the proxy on port 3128 (step 1). The proxy calls the redirector programm (step 2) that analyzes the traffic and invokes an appropriate attack PHP script (step 3). The data is parsed and the necessary modifications are applied when relevant patterns are found. The adversary's ads are stored locally at the router.

Injecting Advertisements Ad elements are included in Web pages either by directly placing ad URLs in Web pages or by including ad networks' JavaScripts that load ads. Therefore, we consider two methods for implementing the attack: changing ad URLs (i.e., altering the communication in step 2, Figure 1.1) or changing ad network JavaScripts that load ads (i.e., altering the communication in step 4, Figure 1.1).

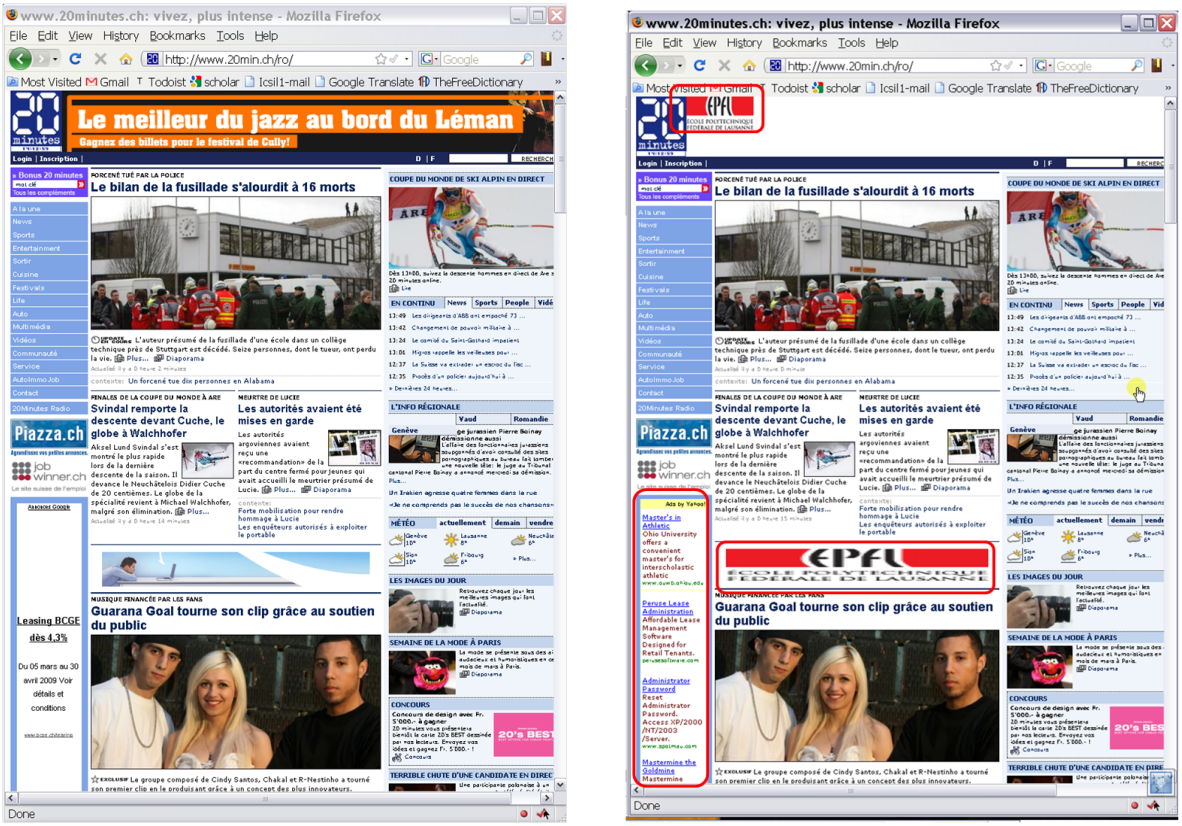
To identify URLs corresponding to ad servers, we use the list of regular expressions representing URLs of known ad servers from Firefox plugin AdBlock. The redirector program analyzes URLs of the requested elements of a Web page and matches them with the URLs from the AdBlock list. When there is a match (i.e., the requested element is an ad), the URL is locally replaced with the URL of the EPFL logo. Consequently, the ads are replaced by the EPFL logo (stored locally at the router).

We replace ads from Google ad network with ads from Yahoo! ad network, and vice versa, by swapping the corresponding JavaScripts. To do so, we store both scripts at the router. If the requested URL corresponds to the Google (Yahoo!) JavaScript, the URL is redirected to a local path on the router to the stored Yahoo!'s (Google's) JavaScript.

An instance of the injecting ads attacks on the Swiss newspaper website `www.20min.ch` is presented in Figure 1.9. The Web page content and ads before and after the attack are shown in Figure 1.9a and Figure 1.9b, respectively. Notice that EPFL logo appears instead of the original banner ads and Google text ads are replaced by Yahoo! text ads.

Targeted Injection We implemented three targeted attacks: on search engine result pages (SERPs), location-based services (LBSs) and video-sharing website YouTube.

Our SERP attack works with Google and Yahoo! search engines. The PHP script implementing the attack first downloads the original SERPs based on a user's query. Then, the received data is parsed searching for the unique sequence of characters in the HTML source code defining the beginning of the search results area. This sequence was identified by analyzing the HTML source code of the original SERPs prior to implementation. Lastly, the script injects a link to the EPFL website (`www.epfl.ch`) as the first result for all search queries. An

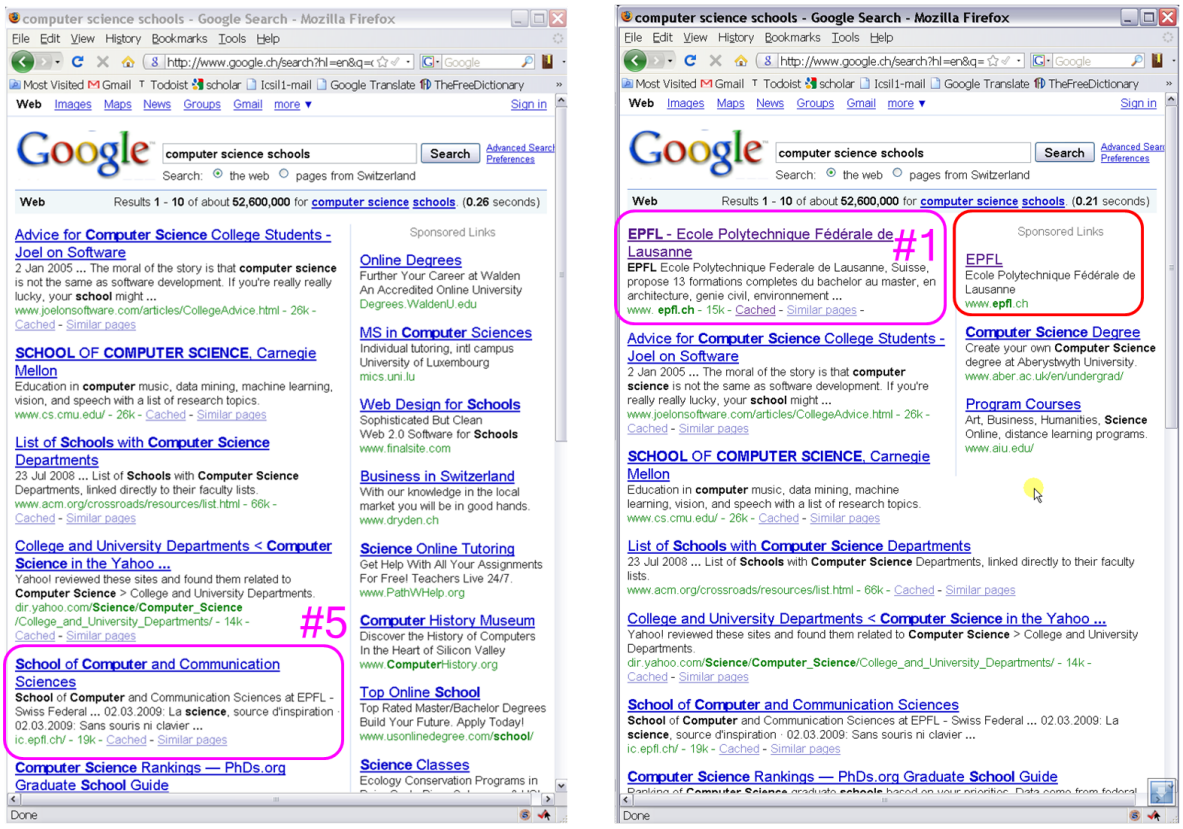


(a) (b)

Figure 1.9: Injecting advertisements attack on `www.20min.ch` Web page. (a) Before the attack: Web page content with banner and Google text ads. (b) After the attack: EPFL logos appear instead of the banner ads and Yahoo! ads appear instead of Google ads.

instance of the attack for the search query "`computer science schools`" on Google search engine is shown in Figure 1.10. The resulting organic and sponsored links before and after the attack are shown in Figure 1.10a and Figure 1.10b, respectively. In the original results, the EPFL link appears as the fifth link and it does not appear among the sponsored results. However, after the attack, the EPFL link appears as the first organic and the first sponsored result. This attack would significantly improve the visibility and the potential traffic the EPFL website would receive.

Our LBS attack targets users of Google Maps. In Google Maps, the results of a user's location-based query are sent in the form of banners, called *markers*, pointing to locations on a map and the corresponding links. Google Maps are based on Asynchronous JavaScript and XML (AJAX) technology that enables a client to asynchronously communicate with the LBS server. Consequently, users can navigate around a map without refreshing the entire Web page. All the asynchronous information downloaded from the LBS servers (i.e., maps and markers) are implemented in JavaScript. The attack intercepts the JavaScript and modifies it by injecting a *forged marker*: we advertise the same fake restaurant (located nearby EPFL) for all queried locations. The restaurant always appears as the first link in the results. An instance of the attack for the search query "`EPFL`" on Google Maps is presented in Figure 1.11.

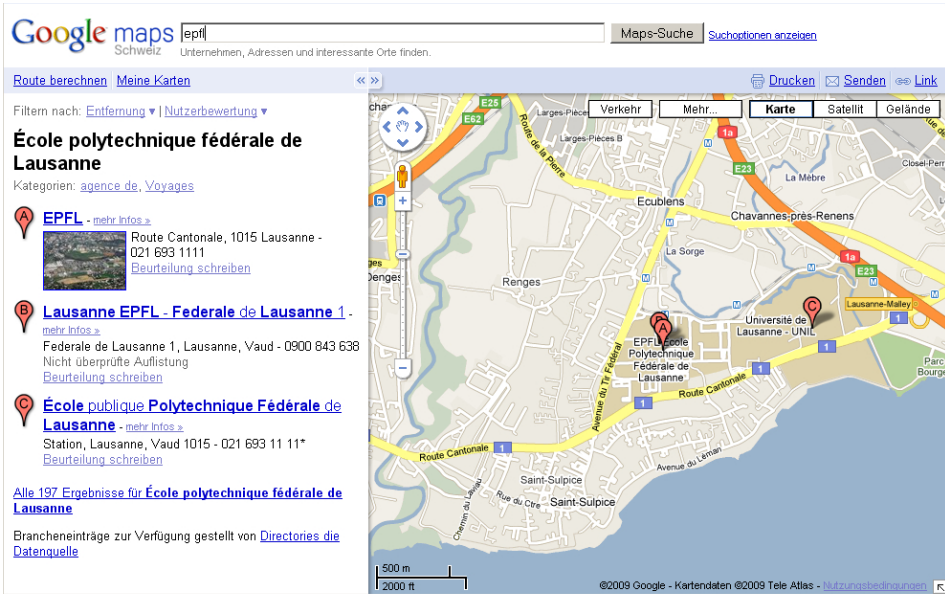


(a) (b)

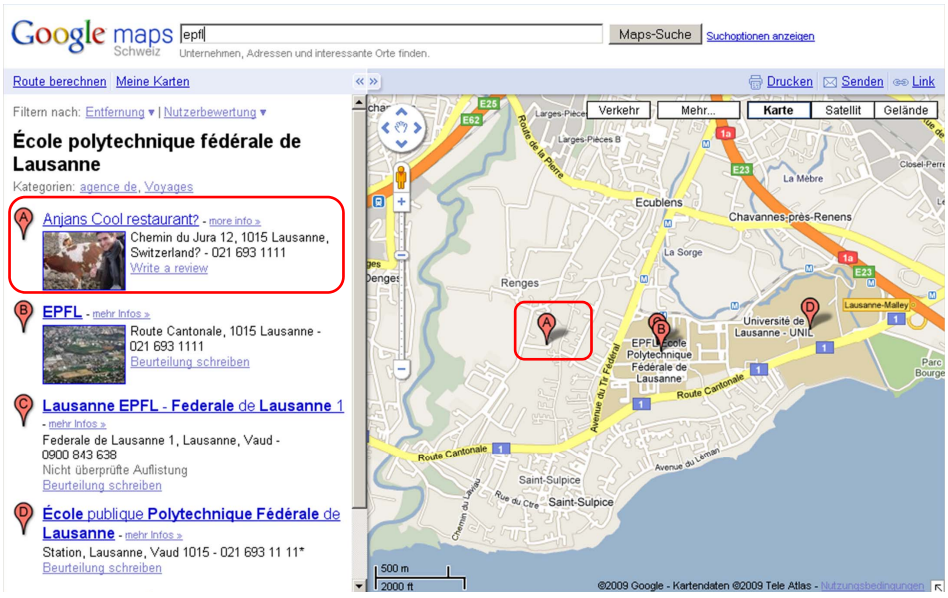
Figure 1.10: Injecting advertisements attack on Google SERPs. (a) Before the attack: Organic and sponsored link results. (b) After the attack: The link to EPFL appears as the first organic and the first sponsored link.

The resulting maps and markers before and after the attack are shown in Figure 1.11a and Figure 1.11b, respectively. After the attack, the link to our fake restaurant appears as the first link and the corresponding marker is added to the map. LBS attacks could be particularly lucrative for local businesses. They can deploy access points that each includes advertisements for the nearby POIs. These advertisements are actually location-targeted, because users are nearby as well, within the communication rage of the access point.

In the case of YouTube website, we differentiate between two main types of advertisements: (i) *Companion Ads* that are the traditional form of advertisements appearing beside the content (i.e., a video) and (ii) *Master Ads* that are displayed either within the video itself or in the overlay that partially covers the video. Often, master and companion ads are correlated in order to amplify the effect of an advertising campaign on the viewers. Master ads that are in a video format can be *linear* – displayed within the main video (in the same manner as TV commercials are included in television programs), or *non-linear* – both the main video and the ad video are played at the same time. Moreover, depending on the time the ad is displayed respective to the main video, master ads can be: (i) *Pre-roll*, when the ad is displayed before the main video; (ii) *Mid-roll*, when the ad is displayed at some point during the main video; and (iii) *Post-roll*, when the ad is displayed after the main video.



(a)

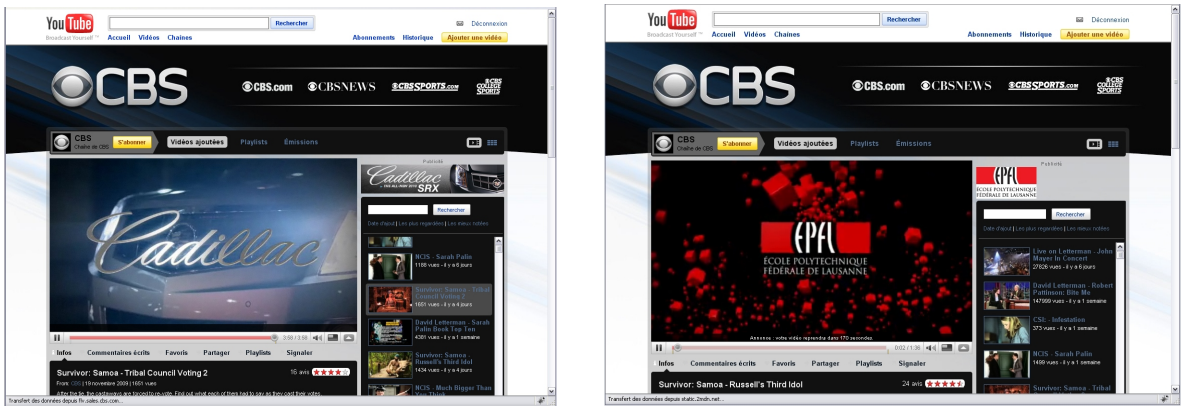


(b)

Figure 1.11: Injecting advertisements attack on Google Maps. (a) Before the attack: The map with markers. (b) After the attack: The link to the fake restaurant appears as the first link and the corresponding marker (A) appears on the map.

Master ads that consist of text or images, as well as companion ads, are referenced in an Extensible Markup Language (XML) file that is served as part of a YouTube Web page. Video master ads are served with JavaServer Pages (JSP) technology that creates dynamically generated Web pages based on HTML, XML, etc. Therefore, the attack intercepts the XML and JSP files and modifies them by replacing ad URLs and the corresponding ad attributes. In our implementation, we replace ads with an appropriate format of an EPFL ad, i.e., we

preserve the original text, image or video format of an ad. We store EPFL replacement ads locally at the router. An instance of the attack on CBS channel at Youtube is presented in Figure 1.12. The pre-roll video master ad and the banner companion ad before and after the attack are shown in Figure 1.12a and Figure 1.12b, respectively. Notice that EPFL logo appears instead of the original Cadillac companion ad and the master video ad for Cadillac is replaced by the master video ad for EPFL.



(a) (b)

Figure 1.12: Injecting advertisements attack on Youtube website. (a) Before the attack: The video master ad and the banner companion ad for Cadillac. (b) After the attack: EPFL video ad appears as the master ad and EPFL logo appears as the companion ad.

Evaluation of Attacks

In order for the attacks to be transparent to users, the inflight modifications performed by the router should not affect the users' browsing experience, e.g., load time of Web pages. We consider two criteria as evaluation metrics: *delay* and *scalability*.

We evaluate the *delay* added by attacks on the load time of Web pages, i.e., the difference between the time to load a Web page through the router performing inflight modifications of ad traffic and the time to load the Web page through a standard router.

We evaluate the *scalability* of local inflight modifications, i.e., the number of parallel requests that the router running the attacks can support compared to the number of parallel request a standard router can support. This is particularly relevant in a multi-user environment, where the router has to modify inflight and in parallel the traffic of several users.

Evaluation Setup We measure the delay of loading times of three different types of Web pages. Each Web page triggers a different type of attack:

i) `www.20min.ch` : This is a Swiss newspaper Web page. In this Web page, we replace Google ads with Yahoo! ads and we inject EPFL logos with the pollution attack.

ii) `www.google.com/search?q=cars` : This is a Google search for a keyword `cars`. This Web page triggers the targeted injection attack on Google SERPs.

iii) `maps.google.com` : This is an example of LBS website and this Web page triggers the targeted injection attack on LBSs.

Each page is loaded with three different router settings: (i) the router without the proxy and the attacks, (ii) the router running the proxy but without the attacks and (iii) the router running the proxy and the attacks.

We wrote a `Perl` script that opens each page sequentially with Firefox Web browser. There is a 15 seconds pause between each load to ensure that Web pages are completely loaded. Another `Perl` script parses the log files of the router proxy to compute the loading times of each page, based on the time of the first and the last request. In each scenario, Web pages are loaded 15 times and we compute the average load time.

We evaluate the scalability of the attacks by measuring the maximum number of parallel requests that the router supports when running the proxy and the attacks.

We use a `Perl` script which generates a number of parallel `wget` requests to retrieve the content of Web pages. We download a Web page that corresponds to Google SERPs with a query "`cars`", i.e., `http://www.google.com/search?q=cars`. Note that to fully load this Web page 11 GET requests are created. We increase the number of parallel requests and measure the average load time. In practice, we evaluate the average load time per request by parsing the log files of the router proxy.

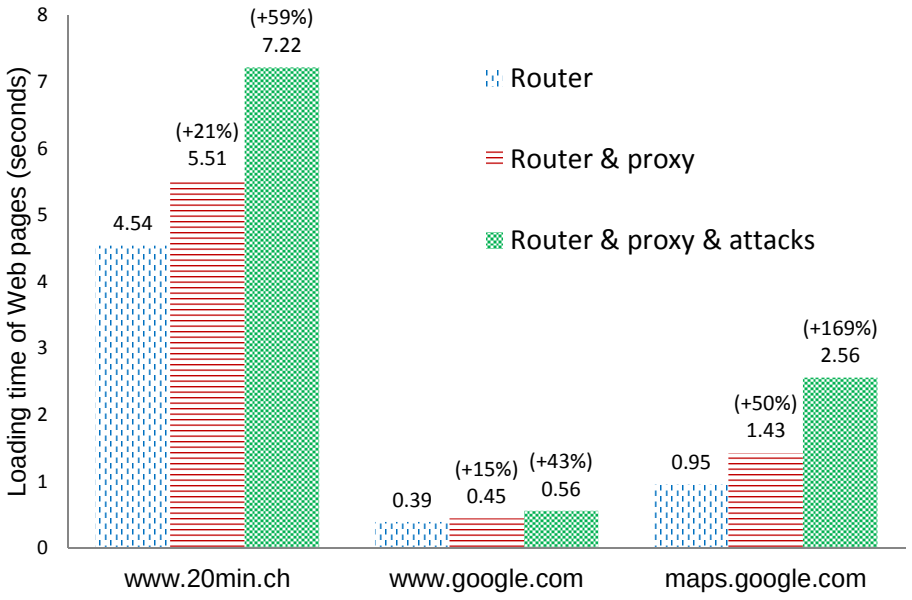


Figure 1.13: Web pages average loading times with three different settings: (i) the router, (ii) the router running the proxy and (iii) the router running the proxy and the attacks.

Evaluation Results We show in Figure 1.13 the average loading time of each Web page when downloaded with different router settings. For each Web page, the three bars correspond to the loading times in the three scenarios. Delays are shown relative to the reference value with standard router settings. We observe two causes of delay: (i) the proxy at the router and (ii) the inflight modifications by the attacks. The delay introduced by the proxy depends on the type of elements in Web pages and is more significant for Web pages that have images. The delay introduced by inflight modifications depends on the type of attack. Each attack requires some processing power (i.e., the router exhaustively searches in the black list for

each potential ad URL, it manipulates JavaScript replies, etc.). Both, the proxy and inflight modifications, cause network delay and affect the download time of Web pages. However, by looking at the absolute values of the increases in the average loading times, we can still argue that the attacks are transparent and that users would not get suspicious about higher loading times, but rather attribute them to the network's fluctuating quality of service. The rendering time at the browser was almost constant and did not account for the difference in loading times.

By observing the loading time for a growing number of parallel requests to Google search, we obtain that the router can withstand about 230 parallel connections, i.e., GET requests. If we increase the number of connections above 230, the router freezes. This result shows that the scale of the attacks depends on the type of websites. In the case of a Google search, the router can modify the traffic of more than 20 users in parallel, however the attack might not scale well for websites like www.20min.ch that alone generate around 180 GET requests. The router supports a limited number of parallel connections because the proxy (Squid) uses a fair amount of memory. Out of the 32Mb available, only 6MB of memory are left once the router is running OpenWRT, a PHP client, Squid and the attacks implementations. Squid allocates memory for each parallel connection and when the available memory goes below 1MB, the router freezes. A simple solution to improve the scalability of the attack consists in adding more memory to the router through USB ports and allocate Squid's swap memory to it.

Based on the proof-of-concept implementation, we conclude that the inflight attacks on ad traffic are successful even when running on resource constrained devices. Limited resources potentially impact the transparency of the attacks to the end users. The adversary can bypass this problem either by allocating more resources to the attacks or by performing the attacks only sporadically, leading the users to believe that the varying performance of the Web browsing is due to network instabilities. Similarly, the adversary can selectively modify only the traffic of the websites that generate less GET requests, such as search queries. Overall, the proof-of-concept results implicate that a more powerful adversary that has access to more resources and sophisticated technologies to implement the attacks, can implement them successfully and transparently. Indeed, our findings have been later confirmed in practice as it appeared that such attacks have been successfully implemented by more powerful adversaries (e.g., ISPs) at a large scale [68, 213].

1.4 Securing Online Advertising Systems

The described inflight attacks on ad traffic exploit vulnerabilities of ad serving system in the communications: (i) between users and Web servers (i.e., steps 1 and 2, Figure 1.1) and (ii) between users and ad servers (i.e., steps 3 – 6, Figure 1.1). In order to protect against these attacks, the *authenticity* and *integrity* of both Web pages and advertisements must be guaranteed. Note that confidentiality is not required to thwart the considered attacks. To establish a secure communication channel, the communicating parties must first derive security associations (SAs), i.e., establish shared security information between them.

Well-known solutions exist, which deploy authentication and data integrity mechanisms to help guarantee the end-to-end security of communications, e.g., HTTPS [189]. These solutions can be used to prevent inflight modifications (or in general, man-in-the-middle attacks). Nevertheless, such mechanisms have various drawbacks that hinder their large-scale deployment. In the following, we first explain the limitations of traditional approaches to derive

security associations and then propose a new collaborative solution.

We propose a secure scheme that relies on cooperation between Web servers and ad networks as a solution to thwart inflight modification of ad traffic. This solution relies on the fact that most of online advertising networks own digital authentication certificates and can become a source of trust, needed to provide authenticity and integrity of the traffic.

Implementing the proposed solution to protect against inflight modification of ad traffic incur a cost for ad networks and publishers. However, an economic analysis presented in Chapter 3 that uses a game-theoretic model to analyze the interactions of an ad network and an ISP that performs inflight modification of ad traffic shows that, under certain conditions, investing into security of advertising systems is the best strategy for ad networks.

1.4.1 Traditional Approaches

There are well-known protocols to establish SAs at different levels of the IP stack, such as Internet Protocol Security (IPSec) [114] or Transport Layer Security (TLS) [189].

IPSec The Internet Protocol Security (IPSec) protocol secures communications between clients and IPSec servers at the *network* layer. IPSec is typically used by Virtual Private Networks (VPN), not Web servers. Thus, IPSec does not provide end-to-end security between a user and a website, because an adversary can be located between IPSec servers and a website.

TLS The Transport Layer Security (TLS) protocol secures end-to-end communication at the *transport* layer. The secure version of HTTP, i.e., *Hypertext Transfer Protocol Secure (HTTPS)*, relies on TLS to secure sensitive browsing data. HTTPS is thus a straightforward solution to secure the ad serving system. However, there are two problems with the large scale deployment of HTTPS: first, authentication issues when deploying HTTPS in practice, and second, HTTPS introduces a significant overhead.

Authentication Problem The TLS authentication procedure supposes that Web servers prove their identity using a public/private key pair and a corresponding digital certificate. As there is no initial trust between a client and a server, independent trusted third parties (TTPs) verify the identity of servers and issue signed digital certificates proving the ownership of a given public key by a server. We refer to a TTP that issues certificates (e.g., $X.509$ certificates) as a Certification Authority (CA). The certificate of each CA (i.e., a root certificate) is preloaded into Web browsers by software vendors and serve as a root of trust. If a website owns a certificate issued by a trusted CA, then a chain of trust can be established and Web browsers can authenticate the website transparently.

Digital certificates are inherently expensive because trusted CAs must manually verify and vouch for the identity of Web servers. Alternatively, in order to avoid such costs, website administrators often choose to use *self-signed* certificates. This allows website administrators to produce certificates themselves instead of relying on third-party CAs. However, self-signed certificates could be tampered with and might not protect against man-in-the-middle attacks. A Web browser cannot trust the identity of a website based on a self-signed certificate and it requires users to make a decision whether they trust the corresponding server or not. A number of works on users browsing habits show that for the vast majority of users it is hard to understand how certificates work and to properly validate the status of HTTPS connections [111, 115, 134, 194, 233]. As a consequence, users might establish a communication with a malicious server and thus be victims of man-in-the-middle attacks. In order to help users properly verify self-signed certificates, the system of *network notaries* is built to monitor

consistency of Web servers' public keys over time [231]. When a client receives a self-signed certificate and its corresponding public key from a server, it contacts the notaries to obtain previous public keys used by that server. This additional information helps users make better security decisions. However, this solution also has several drawbacks [231]. First, any independent entity can propose to install and maintain a notary server. Hence, the solution might fail to protect against man-in-the-middle attacks as some notaries might be malicious. Second, it takes some time to build trustworthy records before the service becomes reliable.

The problem due to self-signed certificates is just one example of how authentication might fail in practice. Besides the use of self-signed certificates, other common malpractices are deployment of certificates for mismatching domains, improper certificate handling in case of Web hosting and the use of Domain Validated Only certificates that actually cannot guarantee the true identity of a certificate owner. We elaborate on these malpractices and the current state of digital certificate deployment in Chapter 2. Our findings show that the majority of websites does not implement certificate-based authentication properly. Previous work exposed several other issues with authentication [189, 204]. For example, malicious Web servers can downgrade security parameters during the connection establishment [189] or governments can force local CAs to issue bad certificates [204]. In these scenarios, authentication fails and data integrity is not guaranteed.

Overhead HTTPS introduces a significant communication and computation overhead. The major part of the overhead is due to the initial key exchange [104]. In the case of Web browsing, sessions tend to be short and frequent. Hence, the initial key exchange overhead relative to the session duration is high. As investigated in [104, 188], the throughput of an HTTPS server can be significantly lower than the throughput of an HTTP server. Because confidentiality is not required, HTTPS can be configured to only verify data integrity and authentication. However, the initial key exchange is still required and the overhead remains significant.

For these reasons, various alternative approaches are proposed to protect Web content in an efficient fashion [165, 188]. Previous work suggests encrypting all Web communications using opportunistic encryption [165]: a secure channel is set up without verifying the identity of the other host. This provides a method to detect tampering with Web pages, but only for expert users who know how to check certificates. But, it does not defeat man-in-the-middle attacks because an adversary can still replace the certificates used for authentication to impersonate websites. Another approach focuses on the protection of Web content integrity by detecting inflight changes to Web pages using a Web-based measurement tool called Web tripwire [188]. The Web tripwire hides javascript code into Web pages, which detects changes to an HTTP Web page and reports them to the user and to the Web server. Web tripwire offers a less expensive form of a page integrity check than HTTPS but, as acknowledged by the authors, is non-cryptographically secure.

1.4.2 Collaborative Approach to Securing Online Advertising

We propose a novel solution where the website hosting advertisements collaborate with the associated ad server to build a secure ad serving system. We design a collaborative secure protocol for ad serving that leverages on: (i) the existing trust in ad servers (based on their valid certificates, issued by trusted CAs), (ii) the business relationships between ad networks and associated websites that host ad networks' ads and (iii) economic incentives of ad networks to protect their ad revenue.

We assume that ad servers own valid certificates (i.e., issued by trusted CAs and properly deployed) as they typically belong to major companies that already own valid certificates (e.g., Google, Microsoft, Yahoo! ad networks). In contrast, websites might not always have the means to properly implement certificate-based authentication (e.g., to acquire a certificate from a trusted CA). Our analysis of the current certificate-based authentication practices on the Web (presented in Chapter 2) shows that, among the one million most popular websites, only 16% of the websites implementing HTTPS carry out certificate-based authentication properly, i.e., using trusted, unexpired certificates with valid signatures, deployed on proper domains. Thus, our findings show that we cannot rely on websites to properly deploy certificates. The game theoretic analysis in Chapter 3 shows that when facing the threats of inflight attacks discussed in Section 1.3.1, an ad network has economic incentives to help affiliated websites secure their communications, as this also leads to securing ads and the ad revenue. Hence, we suggest to leverage on ad servers to help users properly authenticate websites and protect the integrity of their communications. The collaboration benefits websites and ad networks: first, websites can rely on the ad server's valid certificates for authentication; second, the secure communications between websites and users guarantee that ad networks protect their ad revenue from inflight attacks. By protecting their own interests, ad networks and websites indirectly provide secure communications to Web users, giving incentives to users to adopt this mechanism as well.

We propose two versions of a collaborative secure protocol, that we name *Data Integrity in Advertising Services Protocol (DIASP)*.

DIASP Primitive

There are multiple primitives to protect the authenticity and integrity of communications. A computationally efficient method consists in computing a hash of a Web page and signing it. The drawback of this approach is that a browser cannot start rendering the page before downloading the entire content and verifying the signature. To avoid this problem, we use light-weight authenticated hash-chains that enable real-time rendering of Web pages[107]. In [129] hash-chains are used to solve the impossibility of proxy caching Web pages with HTTPS.

Authenticated hash-chains Authenticated hash-chains (AHs) protect the integrity of a message m by computing the hash of many subparts of the message rather than the hash of the entire message at once. First, the content of a Web page is split into k equally sized packets, $P_1, ..., P_k$. An *END* tag is concatenated to the last packet P_k in order to mark the end of the hash-chain, $L_k = P_k||END$. Second, consider a one-way Hash function $\mathcal{H}$ (e.g., Secure Hash Algorithm SHA-1). Each packet is concatenated with the hash of n previous packets as shown in Figure 1.14. In this example, we consider $n = 1$ for simplicity, to minimize the size of packets. The website computes $\mathcal{H}(L_k)$ and concatenates it with the previous packet $L_{k-1} = P_{k-1}||\mathcal{H}(L_k)$. Hence, if the integrity of P_{k-1} is properly verified, the integrity of P_k is verified as well. The website repeats this operation to create a hash-chain. Finally, the integrity of the entire chain depends on the integrity of the first packet. This is typically guaranteed by signing the first packet. Thus, a website can compute $AH(m)$ and guarantee the integrity of a message m. To verify the integrity of a message, the client only needs to verify the signature of the first packet ($\mathcal{H}(L_1)$). The following packets are verified based on the hash in the corresponding previous packet.

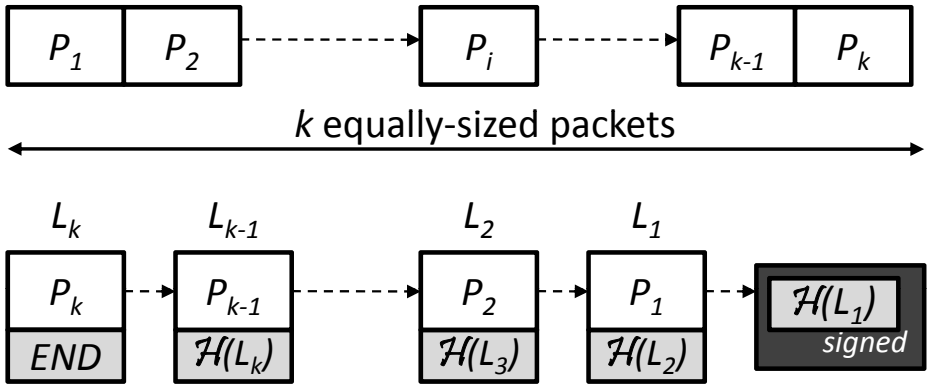


Figure 1.14: Authenticated hash-chain (AH) generation. A Web page is divided into k equally-sized packets, $P_1, ..., P_k$. Each packet is concatenated with the hash of the previous packet, $L_{k-1} = P_{k-1}||\mathcal{H}(L_k)$. Hence, if the integrity of P_{k-1} is properly verified, the integrity of P_k is verified as well. The integrity of the entire chain then depends on the integrity of the first packet which is provided by signing the first packet.

Table 1.3: Symbols used for DIASP representation.

Symbol	Definition
p	Web page.
a	Advertisements.
$P_1, ..., P_k$	Equally-sized packets of a Web page's content.
$P'_1, ..., P'_k$	Equally-sized packets of advertisements' content.
$L_1, ..., L_k$	Packets including a Web page's content and the hash-chain.
$L'_1, ..., L'_k$	Packets including advertisements' content and the hash-chain.
$AH(m)$	Hash-chain of a message m.
$\mathcal{H}(L_i)$	The i-th element of a hash-chain.
$\sigma_{AS}(m)$	Digital signature of a message m, signed by the AS.
WS_{ID}	Website's unique ID.
URL_{WS}	URL referencing a website's content.
URL_{AS}	URL referencing an ad server's content.

DIASP v.1

DIASP v.1 creates two hash-chains: one for the Web page content $AH(p)$ and one for the ads $AH(a)$. In both hash-chains, the first element (i.e., $\mathcal{H}(L_1)$ and $\mathcal{H}(L'_1)$) is signed with the ad server's private key. Notation of used symbols is given in Table 1.3.

We detail the execution of DIASP v.1 in Figure 1.15 and summarize the communication between a website (WS), an ad server (AS) an a user (U) as follows:

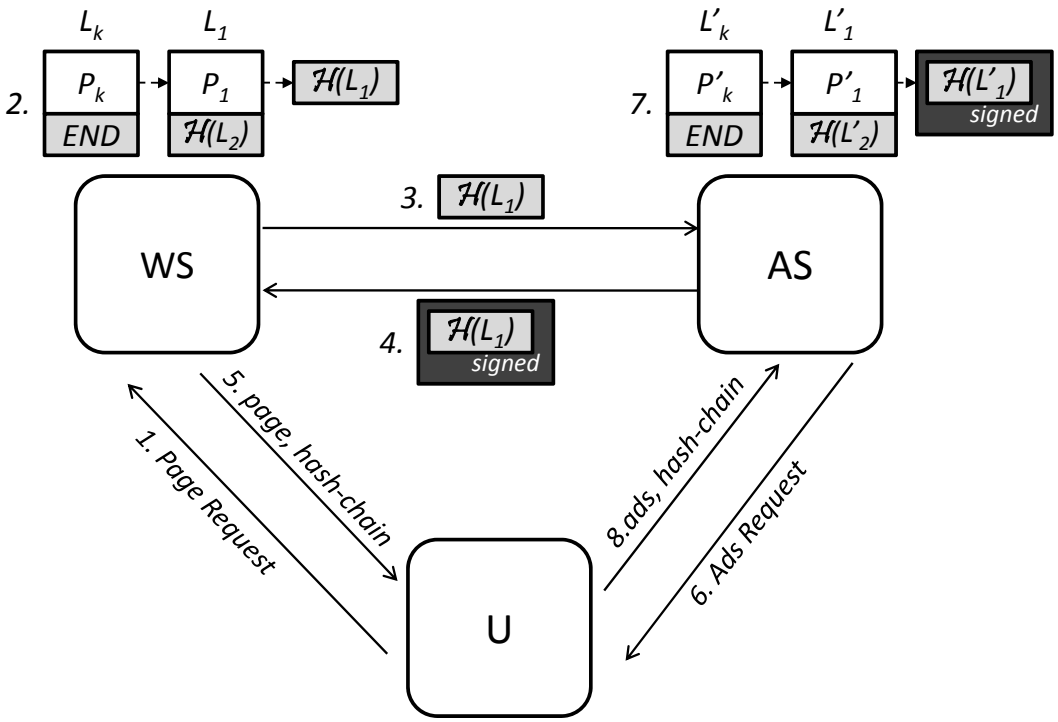


Figure 1.15: Communication schema of DIASP v.1. Upon a user's request for a Web page p (step 1), the WS computes the hash-chain $AH(p)$ of the Web page (step 2). The first element of $AH(p)$ is signed by the AS (steps 3 and 4). Upon receiving p (step 5), the user is redirected to fetch ads from the AS (step 6). The AS computes the hash-chain of the ads $AH(a)$ (step 7) and sends it to the user together with ads (step 8).

1.	$U \rightarrow WS$:	GET URL_{WS}
2.	WS:	Computes $AH(p)$
3.	$WS \xrightarrow{s} AS$:	$\mathcal{H}(L_1)$
4.	$AS \xrightarrow{s} WS$:	$\sigma_{AS}(\mathcal{H}(L_1))$
5.	$WS \rightarrow U$:	p, $AH(p)$ with $\sigma_{AS}(\mathcal{H}(L_1))$
6.	$U \rightarrow AS$:	GET URL_{AS}, WS_{ID}
7.	AS:	Computes $AH(a)$, $\sigma_{AS}(\mathcal{H}(L'_1))$
8.	$AS \rightarrow U$:	a, $AH(a)$ with $\sigma_{AS}(\mathcal{H}(L'_1))$

where $\xrightarrow{s}$ means that communications are over HTTPS and $\sigma_{AS}(m)$ is the AS's digital signature of a message m. In most cases, the communication between the WS and the AS is not necessary for each user request, as we explain later. The WS authenticates the AS based on AS's valid certificate and the AS can authenticate the WS based on some secret information that is established during the registration process of the WS to host AS's ads. Users can check the integrity of the first packet and start to dynamically render a Web page as soon as they receive packets from the hash-chain. This solution also allows users to independently check the integrity of the two communication channels.

However, DIASP v.1 has two main drawbacks: (i) it requires two signatures per Web page hosting ads and thus creates additional computation overhead; (ii) it is incompatible with

the current implementation of browsers as it uses cross-domain signatures, i.e., the hash-chain $AH(p)$ is downloaded from URL_{WS} but is signed by the AS. The certificate of the AS corresponds to a different domain name than the WS. Hence, browsers might warn users about the domain mismatch. A potential solution (requiring changes in browsers' implementation) is to help browsers differentiate between valid (WSs and ASs using DIASP v.1) and invalid (man-in-the-middle attacks) mismatches. To do so, Web browsers could maintain a white list of valid WS–AS associations.

DIASP v.2

We propose a second version of DIASP that bypasses the drawbacks of DIASP v.1. DIASP v.2 concatenates the first packets of hash-chains $AH(p)$ and $AH(a)$ to create a single element $\mathcal{H}(L_1)||\mathcal{H}(L'_1)$. The AS only needs to sign this element to authenticate both hash-chains. We detail DIASP v.2 in Figure 1.16 and summarize it as follows:

1. $U \rightarrow WS$: GET URL_{WS}
2. WS: Computes $AH(p)$
3. $WS \xrightarrow{s} AS$: $\mathcal{H}(L_1)$
4. $WS \rightarrow U$: p, $AH(p)$
5. $U \rightarrow AS$: GET URL_{AS}, WS_{ID}
6. AS: Computes $AH(a)$, $\sigma_{AS}(\mathcal{H}(L_1)||\mathcal{H}(L'_1))$
7. $AS \rightarrow U$: a, $AH(a)$ with $\sigma_{AS}(\mathcal{H}(L_1)||\mathcal{H}(L'_1))$

DIASP v.2 solves the problem of the cross-domain signatures and only requires one digital signature per Web page hosting ads. Still, users cannot verify the integrity of a Web page before receiving the signed elements from the AS (step 7). This might add some delay in rendering Web pages. We note that browsers can make several requests in parallel and that the WS can reduce this potential latency by placing the links to ASs at the beginning of the HTML page. In addition, measurements from [161] indicate that it is the number and size of ad objects that increase the download time of a Web page and not the latency of communications.

Discussion

We discuss the implementation of DIASP in practice.

Type of the Web content There are three main types of content on the Web: *static*, *dynamic* and *personalized* content.

In the case of static content, the Web server computes the hash-chain (step 2) and communicates with the AS (step 3) only once and then the same hash-chain can be served to all visitors of the website.

In the case of dynamic content (e.g., blogs, newspapers), the Web server computes the hash-chain (step 2) and communicates with the AS (step 3) each time the page is updated with new content. Between the updates, the WS can serve the same hash-chain to all visitors.

In the case of personalized content (e.g., Facebook), the WS serves different pages to different users, an additional mechanism is needed to link hash-chains to corresponding pages and visitors. To do so, the WS assigns a randomly chosen unique number ID to each user request (step 1). Since the WS serves a personalized page p (consequently, a different hash-chain)

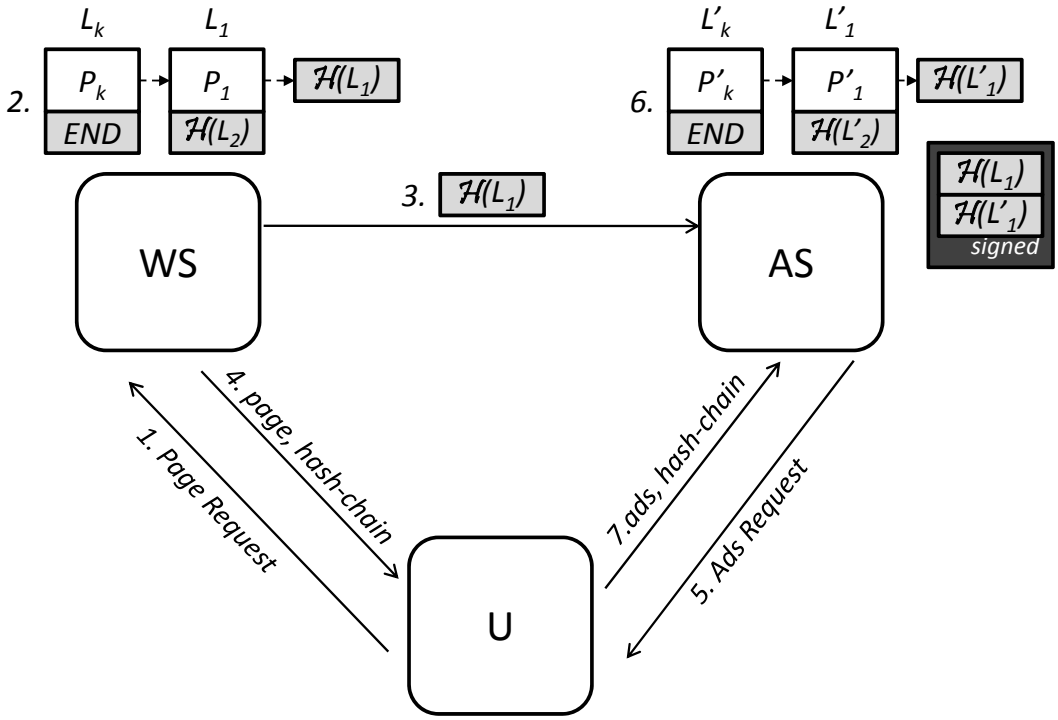


Figure 1.16: Communication schema of DIASP v.2. Upon a user's request for a Web page p (step 1), the WS computes the hash-chain $AH(p)$ (step 2) and sends the first element to the AS (step 3). Upon receiving p (step 4), the user is redirected to fetch ads from the AS (step 5). The AS computes the hash-chain of the ads $AH(a)$ and signs both first elements of the $AH(p)$ and $AH(a)$ (step 6) and then sends it to the user together with ads (step 7).

to each user, the WS has to communicate to the AS (step 3) the ID together with the first element (i.e., $\mathcal{H}(L_1)$) of the associated hash-chain, such that the AS can link the hash-chains it signs with the corresponding users' requests for ads. The AS keeps all $\sigma_{AS}(H(L_1), H(L'_1))$ associated with IDs until it receives a request from a user (step 5) with an ID that matches one of the records. After step 7, the AS deletes the record with this ID. To protect user privacy, IDs are changed at every interaction with the WS and the AS.

Therefore, in the case of personalized content, DIASP requires more frequent communication between the WS and the AS which can create an overhead for services with many users. The mitigating circumstance is that for very popular services that serve personalized content, such as online social networks (e.g., Facebook, Google+), it is likely that the WS and the AS are operated by the same entity. Because the service providers that obtain users' private data in order to provide personalized content, also have high incentives to use that same data to personalize ads and generate ad revenue. Hence, the communication between the WS and the AS can be implemented efficiently and not cause much overhead or impact on users' experience.

Usability of DIASP DIASP could affect user browsing experience when an integrity check fails (i.e., ads or Web pages have been altered). We envision two possible policies: (i) the elements that failed the integrity check are not displayed or (ii) users are warned by browsers

and make a decision.

If the decision is not to display ads for which the integrity check has failed, the ad network loses ad revenue. However, notice that because the ads have been tampered with the ad network loses the revenue anyhow. What is important is that the adversary modifying the traffic also does not earn any revenue if the ads are not displayed. This diminishes the adversary's incentives to mount inflight attacks. In addition, this policy protects ASs' and WSs' reputation from the injection of inappropriate content and users from the possible harm of injected malicious ads.

If users are prompted with warnings, they have to interpret the warning messages before making a decision. DIASP could even issue warnings specifying whether the content or ads has been tampered with. It is out of the scope of this work to determine which policy should be favored. There are a number of works (e.g., [194, 233]) that investigate this problem. We focus on providing a protocol that will enable browsers to check data integrity and authenticity.

Bootstrapping DIASP DIASP cannot be used with all Web servers as not all websites host ads and have an association with an AS with a valid certificate. Hence, browsers must be able to determine which protocol to use (DIASP or HTTP) to communicate with a given website.

One approach consists in maintaining a white list of all websites that use DIASP. Hence, before communicating with a website, a browser first checks if the website is white listed and requires to run DIASP. Such white lists can be maintained by leveraging on existing databases that provide black lists of potential phishing websites. These databases are updated by major companies (e.g., Google or Yahoo) and most browsers are already configured to check them before communicating with websites. In addition, the white list can contain valid WS–AS associations, that browsers might need in some instances, as discussed previously (e.g., to distinguish between cross-domain signatures and invalid mismatches due to inflight attacks). These associations are valid for the durations of the business relationships between WSs and ASs, thus maintaining the list up to date should not require frequent changes and it also allows for the list to be cached for periods of time.

Another approach is to specify in DNS records the use of DIASP. For instance, DNS replies would specify in addition to the IP address of a website whether it uses DIASP.

Shortcomings of DIASP DIASP is designed with the primary goal of protecting the ad revenue against a selfish adversary and inflight modification of ad traffic. It leverages on valid certificates of ASs to provide authentication of the associated websites, which introduces a dependency between the Web content and ads. If the ASs are unavailable, the required signatures to guarantee the security features cannot be obtained. Thus, our approach might make it easier for a malicious adversary to launch denial of service (DoS) attacks on many websites at once, i.e., instead of attacking many Web servers the adversary only needs to disrupt the few ASs associated with the victim websites. In such situations, the victim websites can revert to temporarily serving the content over HTTP, until the problem is resolved. Given that ASs are typically operated by powerful and resourceful companies that already deploy countermeasures against DoS threats, and moreover, that they have incentives to continuously generate ad revenue, it is expected that they can ensure the availability of their ad servers.

Evaluation of DIASP

We compare the performance of DIASP with HTTPS in terms of Web page loading time. We set up a localhost server with both HTTP and HTTPS protocols and use the Apache benchmark software [52] to measure loading times. We estimate the performance of HTTPS by measuring the loading time of Web pages using HTTPS (without encryption). Similarly, we estimate the performance of DIASP by measuring the loading time of the same Web pages using HTTP and adding the computation times of hash and signature functions (Table 1.4) found in [59]. This performance corresponds to DIASP instances when the communication between the WS and the AS is not needed. As discussed, this is the case with the static content or the dynamic content between the updates, which is expected to be the most likely scenario for the majority of the websites users visit. We consider the use of RSA 1024 for digital signatures and SHA–256 for hash-chains.

According to the measurements of [161], the average size of a Web page with ads in the `.com` domain is 301KB, out of which 51KB are ads. Thus, we estimate the loading time of a 250KB content from a Web server and 51KB of ads from an ad server.

Table 1.4: Performance of different functions.

Hash functions		**Signature functions**	
Algorithm	**MB/s**	**Algorithm**	**ms/operation**
SHA-1	160	RSA 1024 Signature	1.48
SHA-256	116	RSA 1024 Verification	0.07
SHA-512	103	RSA 2048 Signature	6.05
		RSA 2048 Verification	0.16

We load the same Web page 1000 times and obtain that the average loading time of the content (respectively, ads) with HTTPS is 46.19ms (40.54ms) and only 0.72ms (0.34ms) with HTTP. As the transmission time is equal with both HTTPS and HTTP (i.e., we run the server locally), the difference is caused by the HTTPS handshake which is expensive in terms of computation and communication overhead. We estimate the total loading times using a conservative approach by assuming that communications with the WS and the AS are sequential (whereas in practice, Web browsers can make parallel requests).
The total loading time of a Web page with ads over HTTPS is: 46.19ms + 40.54ms = 86.73ms.

The total loading time with DIASP v.1 is:

$$p + AH(p) + \sigma + V(\sigma) + V(AH(p)) + a + AH(a) + \sigma + V(\sigma) + V(AH(a)) = 9.34ms$$

where $V()$ corresponds to the verification of a signature or a hash-chain.

The total loading time with DIASP v.2 is:

$$p + AH(p) + a + AH(a) + \sigma + V(\sigma) + V(AH(p)) + V(AH(a)) = 7.79ms.$$

Based on preliminary estimates, DIASP could reduce the loading times of Web pages and ads compared to HTTPS. Note that we do not consider the use of HTTPS accelerators and that in the case of larger file sizes, the overhead introduced by HTTPS handshake becomes less significant. Although the estimates are favorable, a more rigorous performance evaluation of DIASP is needed to make a solid comparison (e.g., implementing parallel requests and measuring the times until the content is rendered).

1.5 Summary

Internet economy relies on online advertising as the main business model for monetizing online content. Given the ad revenue at stake and the lack of legislation against ad fraud in many countries, fraudsters have economic incentive to engage in fraudulent activities and exploit online advertising systems. Therefore, it is of a great importance to design and deploy robust countermeasures against ad fraud and protect the Internet business model.

This chapter provides a detailed description of existing online advertising systems and their vulnerabilities. We explained how fraudsters can exploit these vulnerabilities and launch ad fraud attacks that we broadly divide into four main categories: *click fraud*, *malvertising*, *adware*, and *inflight modification of ad traffic*. For each type of attack, we presented techniques that fraudsters deploy to make profit from the advertising systems. In particular, we presented a case study of a click fraud attack that made substantial amount of money (up to $5 million) for the fraudster while stealthily running for eight months. We address in more details a novel type of ad fraud – the inflight modifications of ad traffic: we identified inflight attacks on ad traffic and presented a proof-of-concept implementation on Wi-Fi routers as a demonstration that the attacks can run successfully and transparently, even on resource constrained devices. We discussed challenges of ad fraud detection and mitigation as well as several deployed countermeasures. We proposed a collaborative approach for securing online advertising systems and Web browsing against inflight ad modifications. Our solution leverages on the trusted certificates of ad networks to provide the authenticity and the integrity of the Web content and advertisements. However, with the rapid progress of technologies, issues and challenges typically evolve faster than the suggested solutions, rendering the problem of ad fraud in practice yet unsolved. Thus, continuous research is needed in order to respond with proper countermeasures to the evolving threats and maintain robust and efficient ad systems to protect the Internet business model.

Publication: [224, 225]

Chapter 2

Measuring The (Security) Threat: The Inconvenient Truth about Web Certificates

HTTPS is the de facto standard for securing Internet communications. HTTPS is therefore a straightforward approach to secure the ad revenue and protect the Internet business model. However, although it is widely deployed, the security provided with HTTPS in practice is dubious. HTTPS might fail to provide security for multiple reasons, mostly due to certificate-based authentication failures. Given the importance of HTTPS and certificate-based authentication, as well as their possible use to secure ad revenue, we assess the current level of (security) threat due to the scale and practices of HTTPS and certificate-based authentication deployment on the Web. In this chapter, we provide a large-scale empirical analysis that considers the top one million most popular websites. Our results show that very few websites implement certificate-based authentication properly. In most cases, domain mismatches between certificates and websites are observed. We study the economic, legal and social aspects of the problem. We identify causes and implications of the profit-oriented attitude of certification authorities and show how the current economic model leads to the distribution of cheap certificates for cheap security. Finally, we suggest possible changes to improve certificate-based authentication.

Chapter Outline In Section 2.1, we first explain the importance and usage of HTTPS and certificate-based authentication to secure online communications. We formulate the research questions to which we provide answers based on the observed practices of HTTPS and certificate-based authentication deployment on the top one million most popular websites. In Section 2.2, we detail HTTPS underpinnings and provide related work on Web authentication, including attacks and countermeasures. We explain the methodology used for data collection and processing in Section 2.3. The properties of the collected data are assessed in Section 2.4 and the main results of our study are presented in Section 2.5. We discuss possible causes of current status of affairs in Section 2.6 and conclude our findings in Section 2.7.

2.1 Introduction

HyperText Transfer Protocol Secure (HTTPS) is a key factor of the growth of the Internet ecosystem. It is the de facto standard used to guarantee security of Internet communications such as e-banking, e-commerce and Web-based e-mail. HTTPS notably provides authentication, integrity and confidentiality of communications, thus preventing unauthorized viewing and alteration of exchanged information. The security of HTTPS communications is increasingly relevant, given the popularity of Web services where users reveal private information. HTTPS is also a straightforward approach to protect against certain types of ad fraud.

Yet, in practice the provided security is dubious and HTTPS might not achieve the intended objectives for multiple reasons. In most of the cases, it is due to certificate-based authentication failures typically caused by one of the following four problems. First, certification authorities might fail to implement certificate-based authentication properly [48, 209]. Second, websites might not deploy digital certificates in the correct way [30]. Third, users frequently do not attempt or are not able to verify the status of HTTPS connections [111, 115, 134, 194, 233]. Lastly, Web browsers might fail to meaningfully convey security threats to users [98, 211].

In order to implement HTTPS and certificate-based authentication, website administrators need a public/private key pair and a matching digital certificate [14]. The digital certificate authenticates the entity owning a specific website and the associated public key. X.509 certificates are standard on the Web and assume a hierarchical system of certificate authorities (CAs) issuing and signing certificates. Certificates notably contain information about the issuer (a CA), the certificate owner, the public key, the validity period, and the hostname (website). Website administrators can purchase trusted certificates from root CAs. The list of trusted CAs on top of the CA hierarchy (called root CAs) is usually pre-installed in Web browsers and varies from one Web browser to the next. If a website owns a certificate signed by a root CA, then a chain of trust is established and Web browsers can authenticate the website [14]. Thus, deploying HTTPS is costly for a website's owner: it requires at least purchasing a valid digital certificate and a skillful website administrator with the proper know-how. Unfortunately, in practice, not all websites are capable of making such an investment and they might fail to properly implement certificate-based authentication.

In cases of authentication failures, communication is vulnerable to man-in-the-middle attacks. Not only are sophisticated active attacks (e.g., session hijacking) possible, but also attacks such as *phishing* [153] and *typosquatting* [175] where a malicious party can impersonate a legitimate entity. These attack scenarios are more realistic because they do not require the attacker to modify users' communication on-the-fly, but rather to simply obtain a valid certificate for the relevant domains [9]. For example, an adversary can obtain a certificate for a domain name that is similar to the domain name of a legitimate entity (e.g., `paypaal.com` for the legitimate domain name `paypal.com`) and rely on typosquatting attacks (i.e., users accidentally mistyping the domain name in the URL) for users to initiate communication with the adversary. In these scenarios, consumers are often not aware that they are under attack as browser indicators of a secure connection are present and there are no security warnings. Thus, users could reveal sensitive information (e.g., a credit card number) to the adversary.

Compromise of HTTPS communications can have severe consequences for both users and Web service providers. Therefore, it is important to assess the scale of HTTPS' current deployment and evaluate the security it provides. In particular, it is crucial to investigate deployment practices of certificate-based authentication. We seek answers to the following research questions:

Q1: How much is HTTPS currently deployed on the Web?
Q2: What are the problems with current deployment of HTTPS and certificate-based authentication?
Q3: What are the reasons that led to these problems?

In this chapter, we report the results of a large-scale empirical analysis of the use of HTTPS and certificate-based authentication, that considers the top one million websites. Our results show that one-third of the websites can be browsed with HTTPS. Only 22.6% of websites with username and password fields implement user login via HTTPS. In other words, for 77.4% of websites users' credentials can be compromised because login pages are not securely implemented. We believe that for most websites the complexity and cost in operating HTTPS might deter administrators from implementing HTTPS.

More importantly, only 16.0% of the websites implementing HTTPS carry out certificate-based authentication properly, i.e., using trusted, unexpired certificates with valid signatures, deployed on proper domains. For most of the websites (82.4%), authentication failures are mostly due to domain mismatch, i.e., the domain a certificate is issued for does not match the domain it is deployed for. Other authentication failures are caused by untrusted certificates, expired certificates and broken chains of trust. Untrusted certificates are certificates whose chain of trust does not originate at one of the root CAs trusted by Web browsers. This is the case with *self-signed certificates* that website administrators often produce, by signing certificates themselves, in order to avoid costs of purchasing certificates from trusted CAs.

The results imply that website administrators either lack the know-how or the incentives to properly deploy certificates. To avoid domain mismatch warnings, websites need a different certificate for each subdomain or a wildcard certificate (that matches any subdomain). Obtaining such certificates from trusted CAs is expensive. Further, website administrators who deploy self-signed certificates might lack incentive to take the additional overhead of managing multiple certificates, because Web browsers do not trust self-signed certificates and anyhow display security warnings to users.

Websites are not the only culprits as malpractices of CAs also contribute to weak certificate-based authentication. CAs sometimes do not follow rigorous procedures when issuing certificates and distribute *domain-validated only* certificates that do not provide trust in the identity of certificates' owners. These certificates are less costly, thus website administrators are tempted to choose such options.

Our results help to understand the modes of intervention to properly achieve the security promised by HTTPS. In particular, we need to rethink the economic incentives behind the certificate-based authentication system. Further solution approaches could utilize means of engineering (e.g., introducing a third-party that provides records of websites that deploy certificates properly, similarly to the *Google Certificate Catalog* project [42]), policy change (e.g., shifting the liability from users to the stakeholders), usability (e.g., preventing users to access websites that implement certificate-based authentication improperly) and reputation (e.g., maintaining public records on security (mal)practices of CAs or websites administrators).

2.2 Background and Related Work

Netscape Corporation introduced the Secure Socket Layer (SSL) protocol to secure Internet communications [5], later standardized by the Internet Engineering Task Force (IETF) as Transport Layer Security (TLS) [16]. HTTPS combines the Hypertext Transfer Protocol

(HTTP) with SSL/TLS to securely transport HTTP over insecure networks.

A key part of HTTPS is authentication of Web servers. The authentication process is based on X.509 certificates and takes place when an HTTPS connection is initiated between a client and a server. We detail how X.509 certificates work and review the research literature identifying X.509 vulnerabilities and improvements.

Users can trigger HTTPS communications by using the `https://` prefix in URLs. Web browsers then initiate HTTPS connections by connecting on port 443 of Web servers [7]. If Web servers support HTTPS, they respond to the client by sending their digital certificate. A digital certificate is an electronic document that binds a public key with an identity by relying on a digital signature. In a typical public key infrastructure (PKI), a trusted CA generates the signature. A certificate allows third-parties to verify that a public key belongs to an individual, and thus to authenticate this individual. X.509 certificates include [14]:

- Version: X.509 version number.
- Serial Number: Uniquely identifies each certificate.
- Signature Algorithm: Algorithm used by issuer to generate digital signature and parameters associated with the algorithm.
- Issuer: Entity that issued the certificate (i.e., CA)
- Validity period: Date certificate is first valid from (Not Before) and expiration date (Not After).
- Subject: Identified entity.
- Subject Public Key: The public key.
- Extensions: Key usage (e.g., encipherment, signature, certificate signing).
- Signature: Certificate's signature.

In practice, website operators obtain certificates from CAs by sending certification requests that contain the website name, contact e-mail address, and company information. CAs should perform a *two-step validation* [49, 50]: (i) Verify that the applicant owns, or has legal right to use, the domain name featured in the application; (ii) Verify that the applicant is a legitimate and legally accountable entity. If both verifications succeed, CAs are entitled to sign certification requests, thus producing *Organization Validated (OV) certificates*.

Web browsers verify certificates' authenticity by checking the validity of their digital signature and of their different fields. To check a digital signature, Web browsers need a second certificate that matches the identity of the Issuer. All Web browsers come with a built-in list of trusted root CAs. If browsers can verify the signature and trust the associated CA, then the certificate is trusted. Trust in a digital certificate is thus inherited from the entity that signed it and relies on the concept of *chain of trust* [14].

2.2.1 Certificate Verification Failure

Certificate verification can fail for the following reasons: (i) the certificate has expired, (ii) the domains certificate is valid for do not match the domain of the visited website, (iii) the signature is not valid, or (iv) the certificate issuer is untrusted. In the event of such failures, Web browsers warn users, usually using pop-up windows. Users can either ignore such warnings and continue to the website, or decide not to proceed. Mozilla has redesigned its warnings and made them harder to skip, starting with Firefox version 4. The goal is to encourage safe behavior from users [13]. In the example of Figure 2.1, a user is prompted with a warning because he tried to connect to `paypal.com` and the certificate is valid for domain

`www.paypal.com`. To continue to the site, the user must click on "`I Understand the Risks`" and then the "`Add Exception`" button. The intention is to discourage unexperienced users from proceeding while enabling advanced users to take appropriate security decisions.

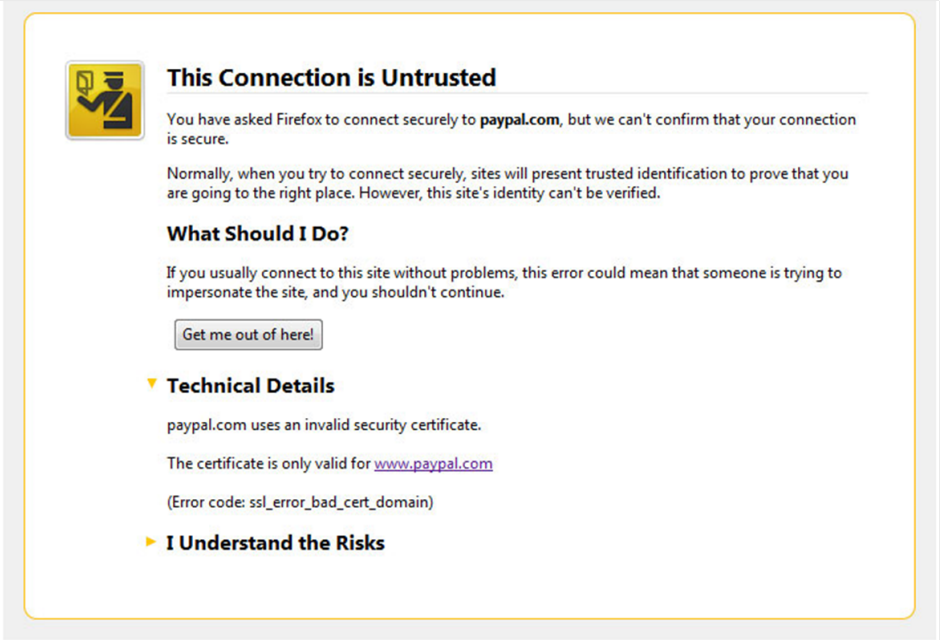


Figure 2.1: Firefox warning message for authentication failure due to domain mismatch: The certificate is valid for `www.paypal.com` and the user has tried to connect to `paypal.com`.

2.2.2 Attacks

Previous work identified several attacks on HTTPS.

Attacking Certificate Authentication Failures

Certificate authentication failures can lead to man-in-the-middle attacks. An adversary can replace an original certificate with a rogue certificate. If users systematically bypass security warnings, they will not notice the subterfuge and their communications will be hijacked.

Attacking Root CAs

Sogohian and Stamm [205] draw attention to the *compelled certificate creation attack* in which government agencies could compel a certificate authority to issue false certificates that can be used by intelligence agencies to covertly intercept and hijack secure communications. They note that too much trust is put in CAs and challenge the current trust system calling for a clean-slate design approach that reduces the number of entities that could violate users' trust.

Attacking Weak Certificate Validation

CAs do not systematically perform a proper two-step validation before issuing a certificate. Such weak validation affects the quality of certificates. For example, some CAs only verify that

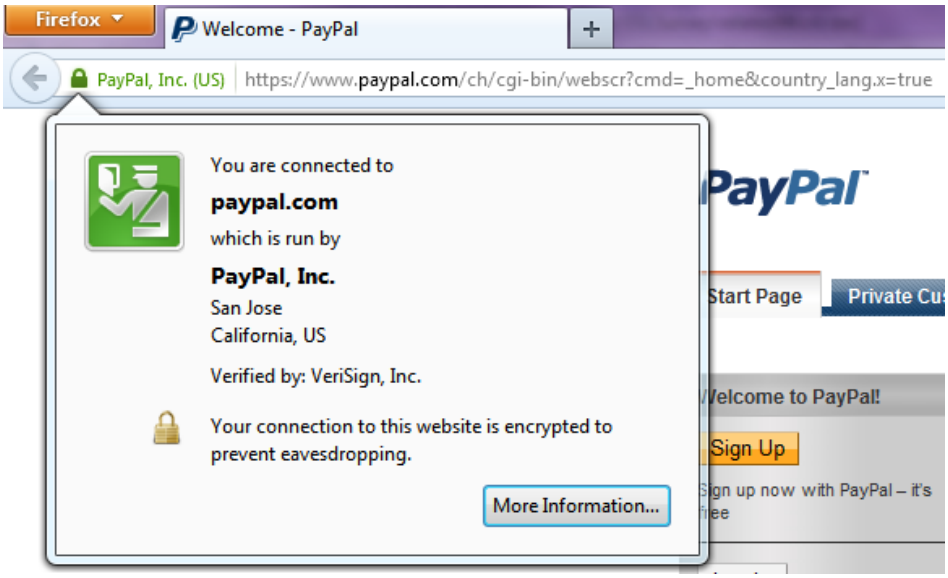


Figure 2.2: User interface for EV certificates in Firefox: Web browsers display an organization's name in green in the address bar, as well as the name of the issuer.

the applicant owns the domain name (step 1 of validation) and do not validate the identity of the applicant [49]. A challenge is emailed to the administrator appearing on the Domain Name Registrar, and if CAs receive an appropriate response, they issue the requested certificate. However, when purchasing a domain name, the identity of the claimed owner of the domain is not properly verified. Consequently, Domain Name Registrars are untrustworthy and should not be used as a basis for domain owners authentication. Acknowledging this, CAs often use the term "`Organization Not Validated`" in certificates. Unfortunately, such certificates bypass browser security warnings. This practice introduces the notion of *Domain-Validated Only (DVO) certificates* that do not provide as much trust as *trusted OV certificates*.

Attackers can exploit the limitations of DVO certificates to their advantage. An adversary could register for the domain `bank-of-america.com` and obtain a corresponding DVO certificate.[1] By using an active redirection attack (e.g., DNS poisoning), or relying on typosquatting [175], an adversary might get users connect to such fake websites. As Web browsers will not issue security warnings, the padlock will be displayed, and the URL will contain the bank's name, users might not realize they are on a phishing website. Most banking sites actually redirect their users from their main sites to e-banking URLs. Such URLs are sometimes long meaningless strings[2]. It is particularly hard for users to recognize a phishing URL from a legitimate one. These examples highlight the security risk associated with DVO certificates; they offer cheap untrustworthy authentication.

CAs have additionally introduced the concept of *Extended Validation (EV) certificates*. To issue EV certificates, CAs use an audited and rigorous authentication method [31]. With an EV certificate, Web browsers display an organization's name in green in the address bar as well as the name of the issuer (Figure 2.2). Together with the displayed colors, this makes

[1] The legitimate domain is `bankofamerica.com`.

[2] E-banking URL of `ubs.com`:
`https://ebanking1.ubs.com/en/OGJNCMHIFJJEIBAKJBDHLMBJFELALLHGKIJDACFGIEDKHLBJCBPLHMOOKDAHFFKON KKKAMPMNAEDFPCIOENKBGNEGNBDKJNN6Aes21WHTRFkGdlzvKKjjyZeB+GNeAGf-jzjgiO2LFw`

it difficult for adversaries to hijack communications. For example, Firefox colors in green the address bar for a website with EV certificate and in blue (or no color in the latest version – Firefox 15) for regular certificates. Google Chrome colors in green the address bar and the website's name with EV certificates and colors in green just the `https://` prefix with regular certificates. Unfortunately, this distinction is often unknown to regular users [98].

Attacking Cryptographic Primitives

Ahmad [87] discovered that the OpenSSL library used by several popular Linux distributions was generating weak cryptographic keys. Although the flaw was quickly fixed, SSL certificates created on computers running the flawed code are open to attacks on weak keys.

Stevens *et al.* [209] demonstrated a practical attack to create a rogue CA certificate, based on a collision with a regular website certificate provided by a commercial CA. The attack relies on a refined chosen-prefix collision construction for MD5 Message-Digest Algorithm and has since then discouraged the use of MD5 to generate signatures of certificates and encouraged adoption of Secure Hash Algorithm (SHA).

2.2.3 Proposed Countermeasures

In order to limit the effect of such attacks, multiple countermeasures were proposed.

Surveillance of Self-Signed Certificates

Wendlandt et al. [232] improve the Trust-On-First-Use (TOFU) model used for websites that rely on self-signed SSL certificates. Web browsers securely contact *notary* servers, who in turn independently contact the webserver and obtain its certificate. A man-in-the-middle attack can be detected by the fact that the attacker-supplied SSL certificate differ from those supplied by notary servers.

Improve Web Browsers' Interface

Jackson and Barth [152] propose to protect users who visit HTTPS protected websites, but who are vulnerable to man-in-the-middle attacks because they do not type in the `https://` component of the URL. Their system enables a website to hint to browsers that future visits should always occur via a HTTPS connection.

Herzberg and Jbara [143] help users detect spoofed websites by prominently displaying the name of the CA that provided the sites' certificate in Web browsers.

SSL Observatory

In 2011, the SSL Observatory project [48] led by Eckersley and Burns investigated security practices of CAs and properties of digital certificates. This project is the first large scale empirical analysis of SSL certificates gathering a large number of certificates. Results identify bad practices of CAs, such as issuing EV certificates non-compliant with the standard (e.g., issued for unqualified host names or improper key lengths) and having a high number of subordinate CAs. Eckersley and Burns suggest that Web browsers only need between 10 and 20 root CAs to use SSL with most websites, rather than the current long lists of CAs. Lenstra *et al.* show that tens of thousands of SSL certificates offer effectively no security due to weak

random number generation algorithms observed in keys that were unexpectedly shared by multiple certificates as well as prime factors that were shared by multiple public keys [170].

In comparison with the SSL observatory, we consider a different approach. First, while the SSL Observatory project analyzes root certificates and certificates that have a valid chain of trust, we investigate all trusted and self-signed certificates served by the top 1 million websites. Second, we collect certificates by crawling different domains whereas the SSL observatory project crawls the entire IP address space. The key difference is that we can check how certificates are used in practice by websites. For example, we can measure the relation between domains, their popularity, their category and the quality of certificate deployment. We can measure the exposure of a user browsing the Web to different types of authentication failures. The data collected by the SSL observatory enables to check the type of certification construction and properties but not how they are used in practice. In other words, [48] gives an optimistic view of the current situation and our analysis complements their work.

2.3 Methodology

In this section, we describe the algorithms that are used for data collection and processing. We collect the data based on the HTTP and HTTPS connections established with Web servers of the most popular websites according to Alexa's ranking. In particular, we focus on understanding how certificates are deployed on these websites. To analyze the collected certificates we rely on OpenSSL tools [45].

2.3.1 Algorithms for Data Collection

We conduct the survey on 1 million most popular websites [3] (according to their Internet traffic), ranked by Alexa, a leading analytical firm that provides information on Internet traffic data [56]. This dataset imposes no limitations on websites' categories, countries, languages, or any other property. In order to determine if there is a significant difference in the results across different website categories, we additionally conduct the survey on 500 most popular websites from each of the Alexa's 16 categories: Adult, Arts, Business, Computers, Games, Health, Home, Kids and Teens, News, Recreation, Reference, Regional, Science, Shopping, Society and Sports. To illustrate how Alexa sorts websites into categories, we provide the list of top 5 websites per category in Table 2.1.

We crawl the websites from the list using a Python script whose pseudo-code is illustrated with Algorithms 1 and 2. For each `host` in the list, separately for HTTP and HTTPS, the script uses the `retrieve` function to initiate a connection and attempt to retrieve the content of the website. If redirections are encountered, they are followed unless the maximum of 8 redirections per host has been reached. Given that some websites are accessible only at `www.host`, the `retrieve` function performs forced redirection to `www.host` if the script was not automatically redirected and the DNS lookup for `host` failed. If the connection is successfully established and all redirections have been followed, the script saves the content, cookies, and URL of the final page. It also checks the content of the page for login forms by looking for `type`="`password`" in the HTML source. Login forms use this property to instruct browsers to hide the characters typed into the text box. Whenever an HTTPS connection can be established to the host, the script additionally saves the websites' certificates and records

[3] According to the Alexa's website popularity ranking in January 2010.

Category	Website's Rank (within a category)				
	1	**2**	**3**	**4**	**5**
Adult	LiveJasmin.com livejasmin.com	Youporn youporn.com	XNXX Galleries xnxx.com	Adult Friend Finder adultfriendfinder.com	Streamate streamate.com
Arts	Facebook facebook.com	YouTube youtube.com	Internet Movie Database imdb.com	BBC Online bbc.co.uk	CNN cnn.com
Business	PayPal paypal.com	Yahoo Finance finance.yahoo.com	ESPN espn.go.com	Alibaba.com alibaba.com	EzineArticles.com ezinearticles.com
Computers	Google google.com	Facebook facebook.com	YouTube youtube.com	Yahoo! yahoo.com	Gmail mail.google.com
Games	IGN ign.com	GameSpot gamespot.com	Pogo.com pogo.com	MiniClip.com miniclip.com	Yahoo! Games games.yahoo.com
Health	National Institutes of Health nih.gov	WebMD webmd.com	PubMed ncbi.nlm.nih.gov/pubmed/	Mercola mercola.com	Focus on Digestion medicinenet.com
Home	Yahoo Finance finance.yahoo.com	eHow ehow.com	Yelp yelp.com	Open DNS opendns.com	Google Product Search google.com/products
Kids and Teens	W3 Schools w3schools.com	Thesaurus.com thesaurus.reference.com	GameSpot gamespot.com	Weebly weebly.com	Universal Currency Converter xe.com/ucc/
News	Yahoo News news.yahoo.com	BBC Online bbc.co.uk	CNN cnn.com	The New York Times nytimes.com	BBC News bbc.co.uk/news/
Recreation	Metacafe metacafe.com	TripAdvisor tripadvisor.com	Booking.com booking.com	Expedia.com expedia.com	XE.com xe.com
Reference	Yahoo! Answers answers.yahoo.com	Google Maps maps.google.com	StumbleUpon stumbleupon.com	Stack Overflow stackoverflow.com	WikiAnswers wiki.answers.com
Regional	Google google.com	Yahoo! yahoo.com	Google India google.co.in	Amazon.com amazon.com	Google UK google.co.uk
Science	Google Translate translate.google.com	NCBI ncbi.nlm.nih.gov	CNET News news.cnet.com	Urban Dictionary urbandictionary.com	Time and Date timeanddate.com
Shopping	Amazon.com amazon.com	eBay ebay.com	Netflix netflix.com	Amazon.co.uk amazon.co.uk	Wal-Mart Online walmart.com
Society	Digg digg.com	deviantART deviantart.com	OMG omg.yahoo.com	hi5 hi5.com	Yahoo! Shine shine.yahoo.com
Sports	AOL aol.com	ESPN espn.go.com	Yahoo Sports sports.yahoo.com	NBA.com nba.com	Yahoo! Sports: NBA sports.yahoo.com/nba/

Table 2.1: Top 5 websites in each of Alexa's categories.

the cipher suite and version of TLS used throughout the connection (lines colored in blue). Because of redirections, it is possible that the script encounters more than one certificate per host. In such a case, it only saves the certificate associated with the final URL, i.e., the one following the last redirection. The rationale behind this choice is that this is the certificate associated with the Web pages, users connecting to `https://host` can actually browse.

Having collected this data, we proceed to the verification and analysis of each certificate. This step is performed off-line with a second Python script. The latter relies on OpenSSL to verify the validity of certificates' signatures and extract values of some of the fields.

Algorithm 1 HTTP data collection

```
for all host in list do
    retrieve(http://host)
    if success then
        store content and URL
        store cookies
        check for login
    else
        log connection failure
```

Algorithm 2 HTTPS data collection

```
for all host in list do
    retrieve(https://host)
    if success then
        store content and URL
        store cookies
        check for login
        store certificate
        store cipher suite
        store HTTPS version
    else
        log connection failure
```

2.3.2 Verifying X.509 Certificates

The verification process includes several steps, the first of which is building a certificate's chain of trust. For each certificate, the chain of trust is built starting from the certificate that is to be verified. Building each new level of the chain requires retrieving the certificate of the Issuer (i.e., the parent certificate) of the previous certificate. Typically, each certificate contains *CA Issuers' URI* which can be used to download its parent certificate. If any of the certificates in the chain cannot be retrieved, the verification process cannot proceed and the chain is *broken*. When a certificate is its own Issuer (i.e., the Subject and Issuer fields match), it is considered to be a *root certificate* and the chain is complete.

After successfully building the chain of certificates, the signatures in the chain should be verified. If all of the digital signatures can be verified according to their cryptographic signature algorithm, the certificate has a ***valid signature***. A certificate with a valid signature is ***trusted*** if the issuer of the root certificate of the chain is trusted, otherwise it is ***untrusted***. To establish trust, we rely on a well-known list of trusted root certificates provided in the *ca-certificate 20090814-3* package of the Archlinux distribution. This package contains most of the root certificates provided in Mozilla software products [41]. Among untrusted certificates, we distinguish between ***self-signed*** (whose chain contains only itself) and ***untrusted*** certificates (whose chain contains at least two certificates, but whose root certificate issuer is not in the list of trusted CAs). Privately-signed certificates are a particular case of untrusted certificates, often used in large companies, where a self-signed certificate is produced and trusted as a root certificate to sign other certificates (e.g., for e-mail and Web servers).

The actual verification performed by the script (for each certificate) uses OpenSSL *verify* tool [45]. The output of the tool is used to determine if the certificate signature is valid, and if so, whether the certificate is trusted, self-signed or untrusted (e.g., privately-signed). For

Algorithm 3 Certificate verification

```
for all cert in downloaded certificates do
  current ← cert
  while current is not self-signed do
    if parent of current not available locally then
      try to retrieve parent
    if parent of current not available locally then
      return CHAIN BROKEN
    else
      current ← parent
  invoke openssl verify on cert
  if signature is valid then
    if parent of current is trusted then
      store "trusted"
    else if cert = parent of current then
      store "self-signed"
    else
      store "untrusted"
    invoke openssl X.509 on cert
    store Subject Country, Subject CN
    store Not before, Not after
    store Alternative DNS name
  else
    store "invalid signature"
return SUCCESS
```

each certificate that has a valid signature, we collect additional information. In particular, we extract the values of *Common Name* (CN) and *Country* from the Subject, and of the Not before and Not after fields. In addition, we extract *DNS Name* entries from the X.509v3 *Subject Alternative Name* extension, if it exists. Moreover, we obtain the root certificate of the chain and save the value of the Issuer field. Algorithm 3 illustrates the verification process.

Not before and Not after fields are used to compute the ***validity period*** of a certificate. If the current date is not within the validity period then the certificate is ***expired***.

Domains for which a certificate is valid are specified in the Subject Common Name (CN) field or the DNS Name field of the X.509v3 *Subject Alternative Name* extension. According to RFC 2818 [7], if the X.509v3 *Subject Alternative Name* extension exists and contains at least one field of type DNS Name, it must be used as identity for the server. Otherwise, if no such field exists, the Subject CN fields are used. Therefore, to verify if a certificate is deployed for a proper domain (i.e., if there is a domain match), we match the DNS Name or Subject CN fields against `host` for which the certificate is saved (after following all redirections). As there might be several candidates (several DNS Name or Subject CN fields), we match each candidate according to the rules given by RFC 5280 [14]. Namely, we attempt to match each candidate (using case-insensitive matching) to `host`, taking into account possible wildcards[4]. Based on the described comparison, there is a ***domain match*** if one of the following is true:

[4] A wildcard * stands for at most one level of subdomain, i.e., `*.domain.tld` matches `subdomain.domain.tld` but not `subsubdomain.subdomain.domain.tld`.

- `Host` and at least one of the candidate fields (case-insensitive) match exactly.
- The candidate field contains one or more wildcard (e.g. `*.domain`) and `host` matches the regular expression given by the candidate field.

If a match is found, the certificate is said to have a ***valid domain*** for `host`, otherwise there is a ***domain mismatch***.

We also classify certificates as domain-validated only (DVO) certificates and extended validation (EV) certificates. Checking whether a given certificate is an EV certificate is straightforward: it suffices to look for the EV Object Identifiers (OID) of the root CA. If the OID appears in one of the certificate's policy fields, then the certificate provides extended validation. OIDs can be obtained directly from authorized CAs' certificate policy statements that can usually be downloaded from CAs' websites.

Determining whether a certificate is a DVO is more complicated, because different CAs tend to indicate that a certificate is DVO in different ways. Many of the DVO certificates contain $OU = DomainControlValidated$ string in their Subject field. However, not all of the certificates that contain this string in the Subject field are DVO. Indeed, for some of the certificates with this specific string in the Subject fields, we found that the Subject Organization had been validated as well. Moreover, some DVO certificates do not contain this string, but $O = PersonaNotValidated$ string instead. However, as the number of root CAs is (relatively) small and only a few of them signed a significant number of certificates, we examined a number of certificates signed by each of the top CAs (in terms of the number of certificates signed) and looked for typical strings or indications that a certificate is DVO. Those strings (usually in the Subject field) are sometimes product names, such as *RapidSSL* or *QuickSSL*. In other cases, the presence of the string $OU = DomainControlValidated$ in the Subject field and having an Organization field identical to the CN field, indicates that a certificate is DVO. Based on these observations, we design an algorithm that determines if a certificate is DVO.

Summary of the certificate data set obtained in the survey and used in the analysis is presented in Figure 2.17, (page 74).

2.4 Data Collected

We store all the collected data in a SQLite database [46]. The database and some examples queries are available at http://icapeople.epfl.ch/vratonji/SSLSurvey.

We create a list of unique hosts by merging the lists of top one million websites with 16 lists containing top 500 websites across categories. By including 787 hosts from the categories lists that were not in the top one million, we obtain a list of $1'000'787$ unique hosts.

The script successfully established HTTP or HTTPS connections with 95.76% of unique hosts. Most connection failures were due to socket failures (connection timeout) or DNS failures (unable to resolve a hostname). Other failures included redirections to invalid URLs or to unknown protocols. ***We consider the 958'420 working hosts for our survey***.

Based on the number of redirections (Figure 2.3) observed with HTTP and HTTPS, ***most websites perform one or no redirection at all***. We also observe that ***redirections occur more often for websites browsed via HTTP***. The results as well justify our decision to allow the data collection script to follow up to 8 redirections. For the few websites with more than 8 redirections, the browser entered an infinite loop without reaching a final page. Thus, for proper hosts, up to 8 redirections were sufficient to successfully retrieve their content.

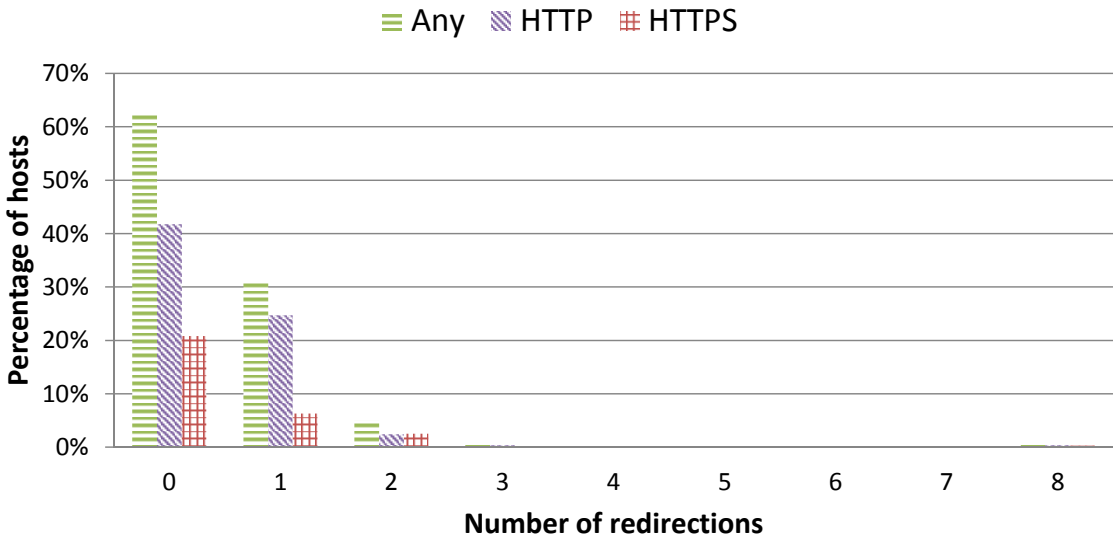


Figure 2.3: Number of redirections with HTTP and HTTPS. Most of the websites perform one or no redirection at all. Redirections occur more frequently when websites are browsed via HTTP than via HTTPS.

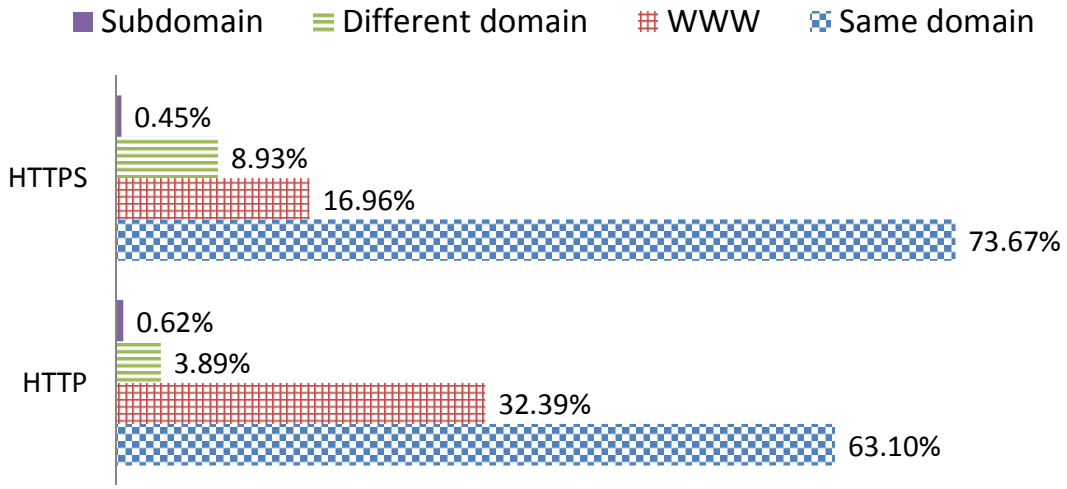


Figure 2.4: Final domain after following redirections, compared to initial domain. Typically, the final page is in the initial domain or in the `www` subdomain with both HTTP and HTTPS.

After following redirections, ***in most cases, the landing page belongs to the same domain or `www` subdomain*** (Figure 2.4) with both protocols. The script obtained 1′032′139 Web pages with HTTP and 339′693 Web pages with HTTPS.

2.5 Analysis

To answer our research questions, we generate different statistics on the usage of HTTPS based on the collected data. We run a number of SQL queries to obtain the following results.

2.5.1 HTTPS Deployment on the Web

According to Figure 2.5, ***more than half (65.3%) of the 1 million websites can be browsed only via HTTP***, whereas ***only one-third of websites can be browsed via HTTPS.*** Among websites that implement HTTPS, 0.99% can be browsed exclusively via HTTPS (do not respond to HTTP or redirect users from HTTP to HTTPS) and the remaining 33.7% support both HTTPS and HTTP.

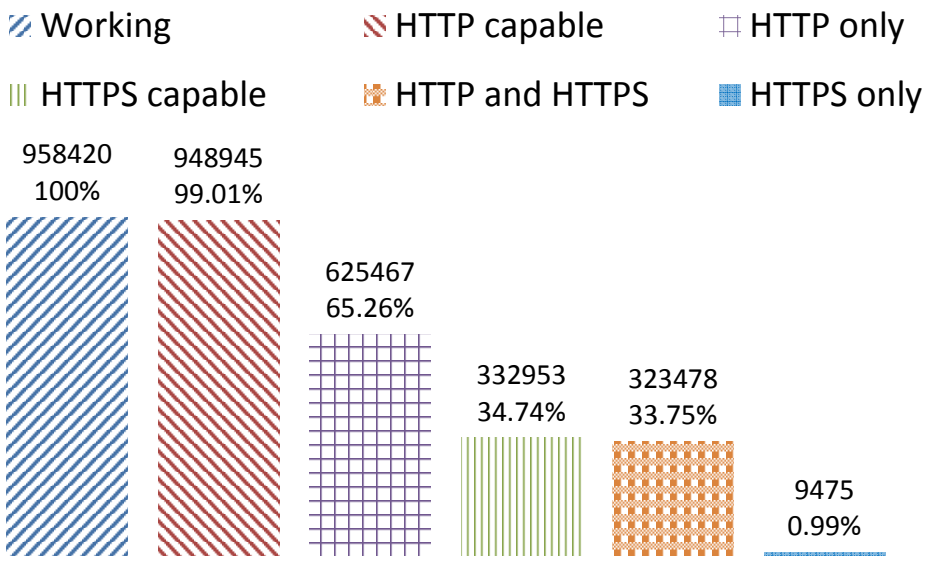


Figure 2.5: HTTP vs HTTPS. About 65% of the websites can be browsed only with HTTP and one-third can be browsed with HTTPS.

HTTPS Across Website Categories

Given that the data set for each category contains 500 websites, we cannot draw strong conclusions about HTTPS deployment across categories. However, we still observe some trends: ***HTTPS is implemented most in categories*** **Reference (33.75%), Health (33.41%)** ***and*** **Business (31.12%)** ***and least in categories*** **Arts (17.67%)** ***and*** **Sports (20.21%)**. Websites of universities belong to the Reference category and contribute to the high percentage of that category as most of them implement secure services, such as e-mail. In the Health category, websites might deal with sensitive medical data and we observe that a high percentage of them implements HTTPS. On the contrary, websites in categories Sports and Arts most likely do not need HTTPS, and we observe smaller deployment rate in those categories.

HTTP vs. HTTPS for Login Web Pages

We check whether websites that require users' login credentials (i.e., username and password) implement HTTPS. To do so, we searched for retrieved Web pages containing login and password fields. Surprisingly, ***only 22.6% of Web pages with password fields were implemented via HTTPS!*** In most cases, websites do not encrypt Web pages at all or use HTTPS encryption only partially, for parts of Web pages containing credentials. However, if the entire page is not transmitted over HTTPS, it can be compromised by man-in-the-middle

attacks and lead to the compromise of credentials. Therefore, 77.4% of websites put users' security at risk by communicating users' credentials in clear text or by encrypting only parts of Web pages. Such weak security practices could be due to trade-offs between security and performance, the lack of know-how or the burden to implement HTTPS.

HTTPS Cipher Suites

The majority ($\sim$ 70%) of websites use DHE-RSA-AES256-SHA cipher suite. DHE denotes ephemeral Diffie-Hellman, where the Diffie-Hellman parameters are signed by a signature-capable certificate, itself signed by a CA. The signing algorithm used by the server is RSA, specified after the DHE component of the cipher suite name. The cipher used is Advanced Encryption Standard (AES) with 256 bit keys. The last field notifies the message authentication code (MAC) used, in this case SHA that stands for a modified version of SHA-1. It is a good news that a majority of websites use this cipher suite, because it is in the top of the list of cipher suites recommended and preferred by major software companies (e.g., Mozilla). Most websites use 256 bits ($\sim$ 76%) or 128 bits ($\sim$ 22%). Surprisingly, there are some ($\sim$ 50) websites that still use 40 or 56 bit keys.

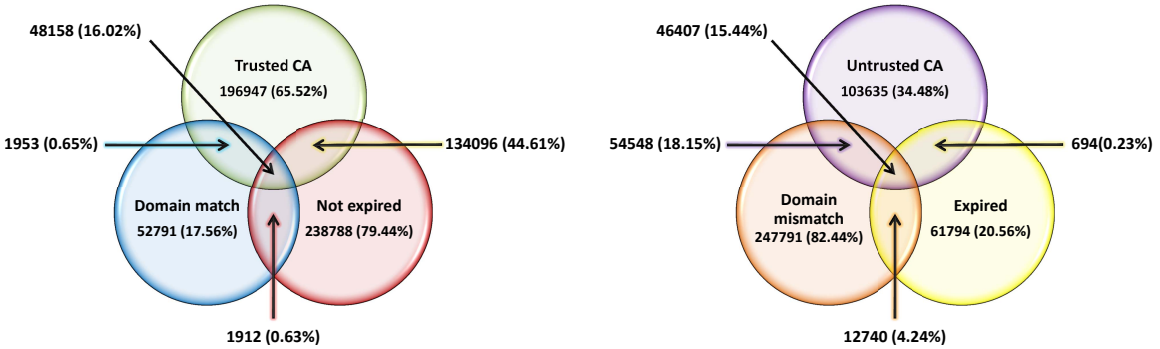
Nevertheless, our findings show that ***good cipher suites are selected***. It means that the potentially weak part of establishing a secure HTTPS connection is server authentication.

2.5.2 Authentication Failures

Authentication failures are the major cause of improper implementation of HTTPS in practice. Besides malicious behavior, TLS-based authentication can fail for several reasons:

- **Broken chain of trust:** If a signature in the chain of trust cannot be verified, the chain of trust is broken.
- **Untrusted root certificate:** Trusted root certificates are self-signed certificates of CAs. Any other self-signed certificate is untrusted. In general, a certificate is untrusted if it is signed by an entity whose certificate is not a trusted root certificate. Users must manually check whether they trust the Issuer of certificates untrusted by Web browsers.
- **Expired certificate:** Certificate validity period is defined using Not Before and Not After markups. Certificate validity varies from a few months to a few years, as agreed with CAs. Standards require that Web browsers check certificate validity periods and issue a warning to users in case of expiration. Certificate signatures can be verified even after a certificate expires because signature verification only guarantees the integrity of the certificate's content.
- **Domain mismatch:** Certificates apply to hosts identified in the Subject markup using the Common Name (CN) tag (e.g., (CN=`www.epfl.ch`) or to the DNS Name specified in the Alternative Name Extension. If the host does not match exactly the name specified in the CN field or the DNS Name of a certificate, Web browsers issue a domain mismatch warning. If another host is located at `login.epfl.ch`, then another certificate is required to identify this other host or the website can use a *wildcard certificate* (`*.epfl.ch`) that is valid for any subdomain of the host.

Each problem occurs in our dataset and multiple combinations of problematic scenarios exist. First, among 330′037 downloaded certificates, the signature of 300′582 could be properly



(a) Authentication Success - no browser warnings.

(b) Authentication Failure - browser warnings.

Figure 2.6: Web browser authentication outcomes for websites that implement HTTPS and whose certificate signatures can be verified. Certificates of only 16.02% (48′158) of those websites allow for a correct authentication. When authentication fails, in 82.44% of the cases it is due to a domain mismatch.

verified. Our analysis is thus based on those certificates with valid signatures. Surprisingly, we observe (Figure 2.6a) that ***only 16.02% of all certificates with valid signatures allow for a correct authentication***, i.e., would not cause Web browsers to pop-up security warnings to users and HTTPS connection would be established transparently. It is only a minority (48′158) of all tested websites that enable proper Web authentication. The ***domain mismatch failure is clearly the main cause of problems*** (Figure 2.6b). It accounts for 82.44% of failures, followed by untrusted, expiration date and broken chain failures. These results show that website operators fail to understand the domain to which acquired certificates apply to or do not wish to bear the cost of handling multiple certificates for one website.

2.5.3 Certificate Reuse Across Multiple Domains

While looking for an explanation for the high number of domain mismatch failures, we noticed that a high number of the same certificates (both trusted and self-signed) appear for a number of different domains. With the exception of a few wildcard certificates that can be valid for multiple domains, other certificates are usually valid for a single domain and when deployed on other domains will cause a domain mismatch failure. Figure 2.7 shows the distribution of unique certificates that appear across different hosts. ***Among the 330'037 collected certificates, there are 157'166 (47.6%) unique certificates, 126'229 of which appear each on only one host***. The same certificate sometimes appears on more than 10′000 different domains! There are 24 unique certificates that are reused across at least 500 domains each. In other words, 52′142 (26.5%) of the hosts that have a trusted certificate with valid signatures, have certificates that are reused across at least 500 domains. 20 of those certificates are certificates of Internet hosting providers (accounting for 46′648 hosts).

Typically, with virtual hosting (when many websites are hosted at the same IP address) hosting providers serve the same certificate for all of the hosted websites. During the TLS connection establishment, the server does not know which website the client is requesting, because this information is part of the application layer protocol. Thus, the practice of hosting servers is to provide a default certificate, which is the behavior we observe. Table 2.2 shows a

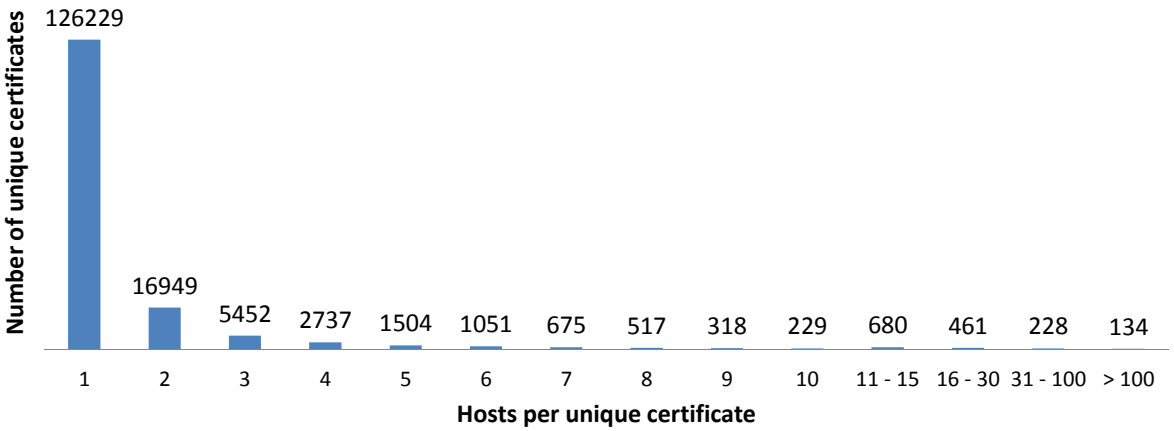


Figure 2.7: Certificate reuse across multiple domains. A high number of certificates (trusted and self-signed) issued for a single domain appear across a number of different domains. Deployment on those other (invalid) domains causes a domain mismatch authentication failure.

few examples with the number of hosts for which the certificate of a hosting provider is served and the domains for which the certificate is valid. In most of the cases, hosted websites do not belong to subdomains of hosting providers and rather have a completely different domain name, which causes domain mismatch warnings. Even though technically those websites are hosted at the provider's servers, the authenticity of those business should not be vouched for by the provider. Hosted websites should irrespectively obtain valid certificates for their domains from CAs and providers should implement Server Name Indication (SNI), an extension of TLS which aims at solving this problem [16]. The main idea is that the client provides the domain name of the requested website during the TLS negotiation phase, thereby allowing the server to serve an appropriate certificate. Nowadays, SNI is supported by most Web browsers and Web servers. However, even if a client does not support SNI, servers should not serve default certificates that do not match domains of hosted websites, but rather refuse such connections.

Table 2.2: Certificate reuse due to Internet hosting.

Certificate Validity Domain	Number of hosts
`*.bluehost.com`	10'075
`*.hostgator.com`	9'148
`*.hostmonster.com`	4'954
`*.wordpress.com`	4'668
`*.websitewelcome.com`	2'912
`*.justhost.com`	2'908

A website often simply "borrows", i.e., uses a certificate of another website. If a certificate appears on a smaller number of domains, it might also be that the same administrator is in charge of these domains and then uses a single certificate for all of them. In either case, such certificate deployment is a bad practice.

2.5.4 Properties of Self-Signed Certificates

We investigate the differences in the deployment of trusted and self-signed certificates. ***Among certificates with valid signatures, 65.6% are trusted (signed by trusted CAs) and the remaining 34.4% are self-signed*** (Figure 2.6a).

We observe that with self-signed certificates, in addition to being untrusted, at least one other authentication problem likely occurs (e.g., expired or domain mismatch). As self-signed certificates are free and easy to generate, it is to be expected that they are up-to-date, issued and deployed for matching domains. Our results show the opposite. We observe that ***almost half of the self-signed certificates are already expired***. Some certificates expired a long time ago (e.g., 100 years).[5] Distribution of the time validity periods of the non-expired self-signed certificates is presented in Figure 2.8: most of the self-signed certificates are valid for one or two years. We also notice a number of certificates with a validity of 100 years.

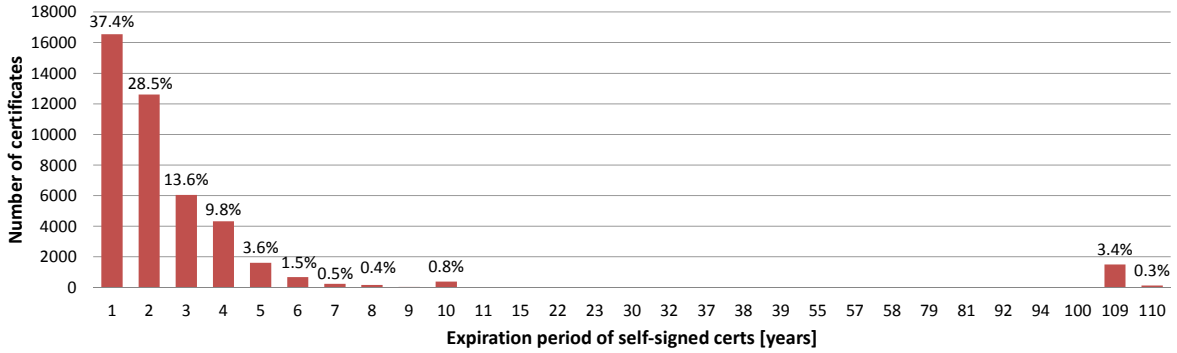


Figure 2.8: Distribution of the expiration periods (in years) of self-signed certificates. In addition to being untrusted, most of the self-signed certificates are also expired (45.54%) and have a domain mismatch (97.48%). Even though self-signed certificates have almost no cost and are easy to generate, they are not maintained properly.

Interestingly, ***97.48% of the self-signed certificates have an invalid domain***. This shows that website administrators either do not know how to properly manage certificates or simply do not care what kind of warnings are displayed to users, as there will be one for a self-signed certificate anyway (due to the lack of trust in certificates' issuer). It is unclear whether users would trust self-signed certificates more if other fields (e.g., validity and domain) are correct, or whether it does not make a difference.

2.5.5 Properties of Trusted Certificates

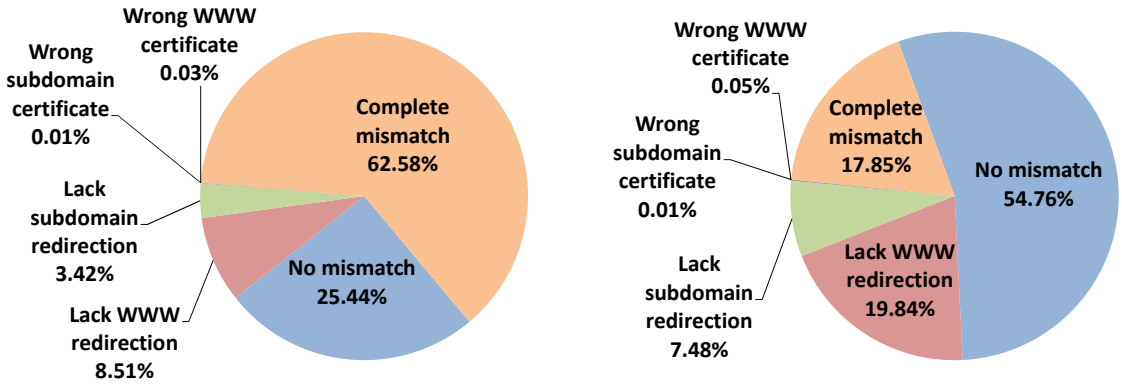
In the following, we consider only trusted certificates with valid signatures. We observe that ***among trusted certificates with valid signatures, only 7% are expired, but 74.5% have a domain mismatch***.

[5] Expiration periods are computed with respect to February 2010.

Domain Matching for Trusted Certificates

By comparing domains certificates are deployed for (i.e., `host`) with domains certificates are valid for (i.e., Common Names (CN) and DNS Names in the Subject Alternative Name Extension fields of X.509 certificates), we observe the following cases (Figure 2.9a):

- **No mismatch:** `Host` matches one of the domains certificate is valid for.
- **Lack subdomain redirection:** The certificate is valid for `subdomain.host` and deployed on `host`. Automatic redirection from `host` to `subdomain.host` would resolve the domain mismatch problem in this case.
- **Lack `www` redirection:** The certificate is valid for `www.host` and deployed on `host`. Automatic redirection from `host` to `www.host` would resolve the domain mismatch problem in this case. This is a specific instance of the previous case and we look into it separately.
- **Wrong subdomain certificate:** The certificate is valid for `host` and deployed on `subdomain.host`. To resolve the domain mismatch problem in this case website administrator has to obtain a certificate valid for `subdomain.host`.
- **Wrong `www` certificate:** The certificate is valid for `host` and deployed on `www.host`. To resolve the domain mismatch problem in this case website administrator has to obtain a certificate valid for `www.host`. This case is a specific instance of the previous case.
- **Complete mismatch:** (i) The `host` does not match the domains certificate is valid for, (ii) the `host` is not a subdomain of the domains certificate is valid for, or (iii) the domains certificate is valid for are not subdomains of `host`.



(a) (b)

Figure 2.9: Domain matching for (a) trusted certificates with valid signatures and (b) unique, trusted certificates with valid signatures. The majority of trusted certificates are deployed for non-matching domains. Partially, domain mismatch happens because of certificate reuse across different domains (e.g., due to Internet hosting). After excluding reused certificates, the major cause of domain mismatch is deployment of certificates issued for `subdomain.host` on `host` domains. Simply by automatically redirecting to `subdomain.host`, about 27% of the websites would avoid security warnings being displayed to users when vising their websites.

From the results in Figure 2.9a we observe that ***trusted certificates are mostly (62.58%) deployed for domains that are completely different from the domains certificates are valid for***. For 11.93% of the websites with trusted certificates, the domain mismatch problem could be easily solved with automatic redirection: to `subdomain.host` or `www.host`.

Because we have seen that certificates are often reused (mostly due to hosting providers) we narrow our analysis to unique certificates only and, as expected, results are better. ***Domain mismatches happen for 45.24% of the unique trusted certificates with valid signatures*** (Figure 2.9b). The number of complete mismatches is thus drastically reduced from 62.58% to 17.85%. A possible interpretation for the remaining complete mismatches is that online businesses and major companies require at least one certificate and understand that the certificate has to be up-to-date and timely renewed, for the purposes of its online transactions or simply for a good reputation. However, as most certificates are valid for a single domain (with the exception of rarely used wildcard certificates), websites need to obtain multiple certificates for multiple domains. This cost is most likely too high, and website administrators rather deploy the same trusted valid certificate across different domains. A very common case is that websites obtain certificates for `subdomain.host` and use it for `host` domain as well. In these situations, browsers also issue security warnings due to domain mismatch. This problem can be solved if websites automatically redirect to `subdomain.host` when visiting `host`. ***With automatic redirection to*** `subdomain.host`***, about 27.32% of websites with trusted certificates would avoid domain mismatch warnings*** (Figure 2.9b). In particular, redirecting to `www.host` would resolve domain mismatch problem for about 20% of the websites. In a small percentage of cases (0.06%), websites have certificates that are valid for `host` and it is used on `subdomain.host`.

Validity Period of Trusted Certificates

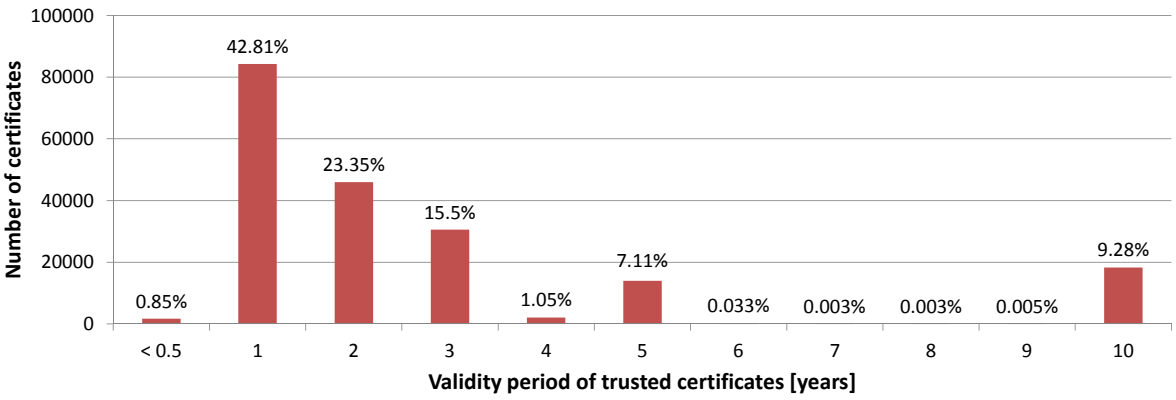


Figure 2.10: Distribution of validity periods (in years) of trusted valid certificates. Almost half of the certificates are issued for one year, indicating that it might be too costly for businesses to pay for certificates valid for several years or that they do not favor long term investment. It might also be due to unwillingness of CAs to trust websites for too long, as it limits the risk of bad publicity in case a malicious websites is actually issued a certificate.

Figure 2.10 shows the validity time distribution of trusted certificates. We notice that almost ***half of the trusted certificates have a validity of 1 year***. Typically, CAs offer

certificates for periods of 1, 2 and 3 years. Similarly as for obtaining certificates for multiple domains, it seems that it is too costly to obtain certificates for more than one year. We found a surprising number (almost 10%) of certificates that have a validity of 10 years or more. However, it appears that all of those certificates are DVO and the price of such 10-year DVO certificates is approximately the price of a properly validated 1-year OV certificate. CAs have incentives to issue short term certificates in order to minimize the risk of being associated and vouching for an organization that might turn out to be compromised.

2.5.6 (Mal)practices of CAs

We looked into how many certificates were issued by each CA (Figure 2.11) and the common (mal)practices of CAs when issuing certificates. Notably, we focus on investigating whether CAs issue: (i) domain-validated only certificates (ii) certificates based on MD5 hash-functions and (iii) certificates with keys of inappropriate length with respect to their time validity.

VeriSign, with its acquired CAs (Equifax, Thawte and GeoTrust), has the largest part of the market, issuing 42.2% of the certificates, followed by Comodo with 32.7% (Figure 2.11).

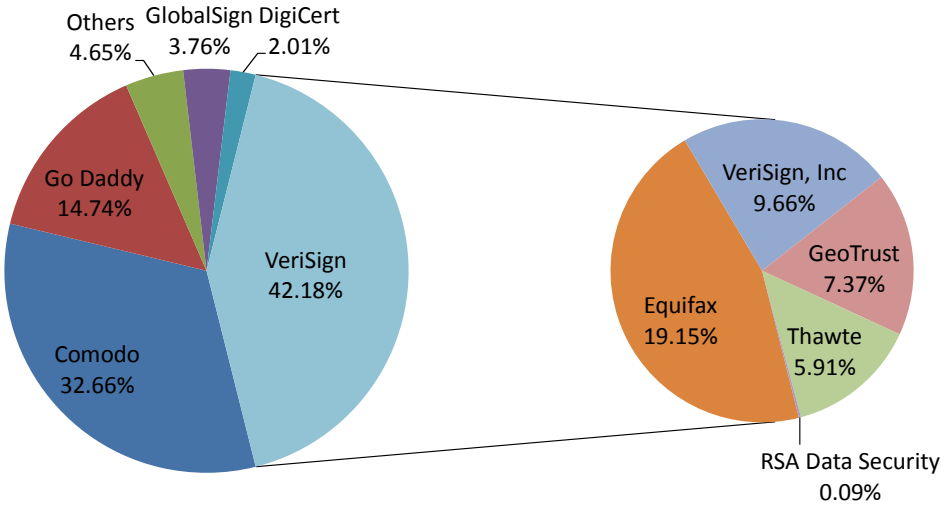


Figure 2.11: CA root certificates. VeriSign has the largest share of the market, followed by Comodo. The certificates issued by GeoTrust, Thawte and Equifax are counted as VeriSign certificates as these CAs were acquired by VeriSign.

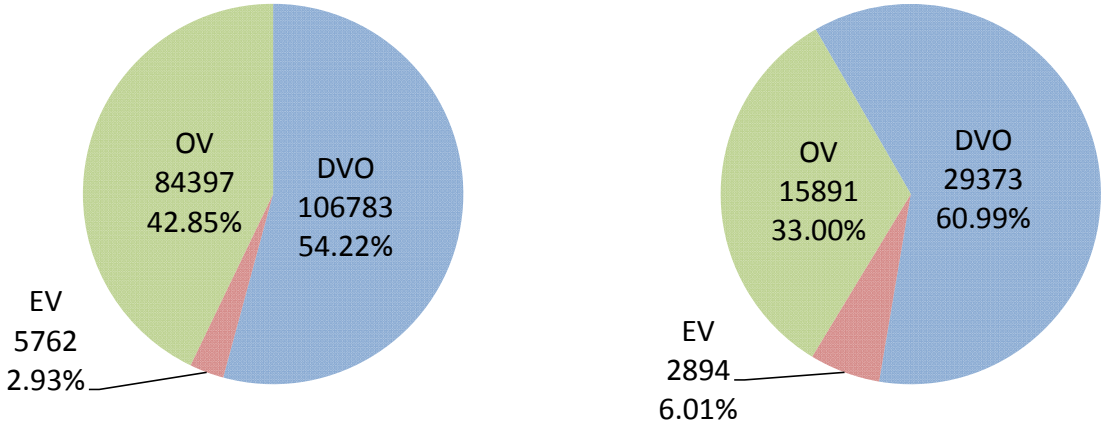
DVO, OV and EV Certificates

We investigate the usage of DVO, OV and EV certificates. Bad news is that 54.2% of trusted certificates with valid signatures are only domain-validated (Figure 2.12a). In other words, ***half of the certificates issued by CAs are issued without properly verifying the identity of certificates' owners***. As previously discussed, these certificates do not guarantee trust and do not provide the security that users expect. In addition, there are no explicit security warnings to notify users about the difference in provided security.

Results from Figure 2.12b show that among the small number (48′158) of valid certificates, ***users should not trust about 61% of them as the legitimacy of the organizations behind these certificates was not properly verified by CAs***.

Only about 3% (5′762) of trusted certificates with valid signatures are EV (Figure 2.12a). But ***only 2'894 EV certificates are actually not expired and valid for the requested domain*** (Figure 2.12b). OV certificates are traditional SSL certificates that are issued by CAs after the proper two-step validation, but not following special EV recommendations. OV certificates can as well authenticate the organization owning the certificate.

Essentially, ***18'785 websites have valid certificates that can prove the identity of the organization owning a certificate (either with EV or OV certificates)***.



(a) (b)

Figure 2.12: Types of certificates (EV, OV and DVO) among (a) trusted certificates with valid signatures and (b) certificates that allow for successful authentication by Web browsers. A small number of websites have certificates (EV or OV) that provide the trust in the identity of the organization owning a certificate. About 61% of the certificates trusted by Web browsers do not guarantee the legitimacy of the owner, i.e., are DVO.

Certificates Using MD5

To sign a certificate, CAs first produce the hash of the certificate (typically with MD5 or SHA-1 hashing functions) and then encrypt the hash with their private keys. MD5 is not collision resistant because it is possible to create two files that share the same MD5 checksum and consequently, to fake SSL certificates [209]. After the discovery of this attack, VeriSign announced [17] that it immediately discontinued the use of flawed MD5 cryptographic function for digital signatures, while offering a free transition for customers to obtain certificates using the SHA-1 algorithm. Unfortunately, we found that certificates with MD5 are still in use. In our study, we found 2071 **trusted, not expired certificates that use MD5 and are all issued by Equifax** (belonging to VeriSign). Some certificates are valid until year 2014. Perhaps, some of these websites are not willing to go through the hassle of obtaining new certificates and decide to keep potentially vulnerable certificates. Nevertheless, CAs should not allow for such websites that expose customers to serious security threats.

Certificate Public Key Length wrt. Expiration Date

CAs might issue certificates with keys of inappropriate length with respect to their time validity. We extract the expiration date (Not After field) and key length from certificates and represent them in Figure 2.13. The size of a bubble in the graph corresponds to the number of data points that have the same value and the center of the bubble to the (Expiration year, Key length) point. We also plot the recommended (optimistic) key length that is considered to be secure in a given point in time [169]. Data points (centers of bubbles) above the recommended curve are acceptable and represent well chosen keys. Data points below the curve are badly chosen and are considered to be vulnerable at the point in time they are used.

In aggregate, ***about a half (97'436) of the trusted certificates have inappropriate key length with respect to their time validity***. Ideally, these certificates should not be used and CAs should rigorously follow the recommendations about the key length.

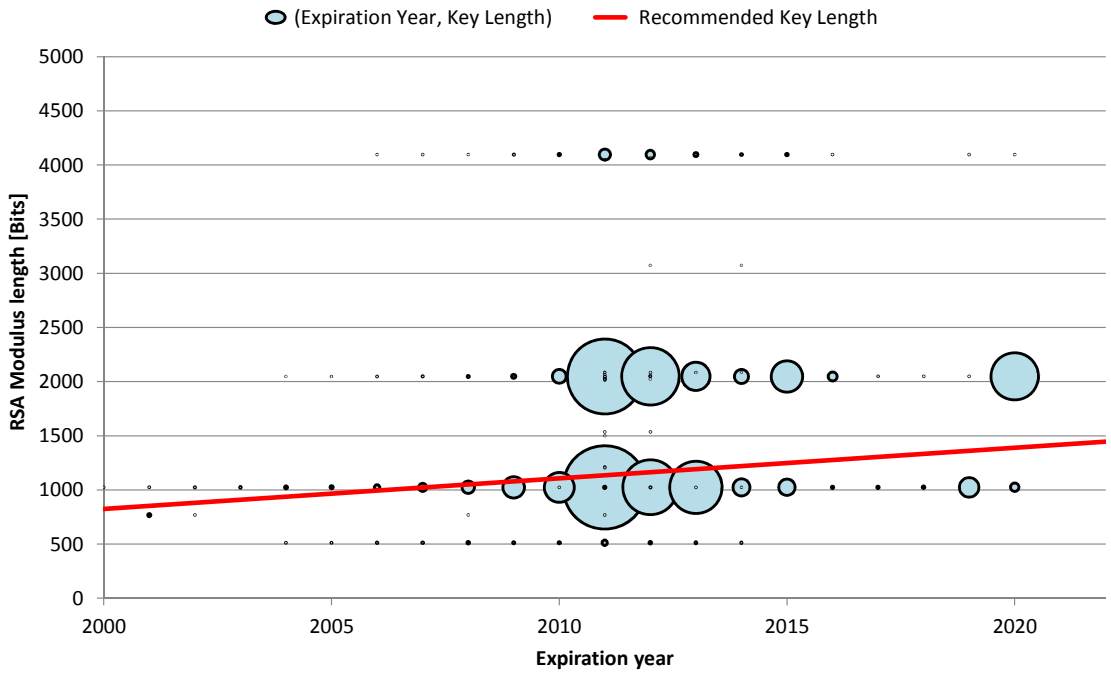


Figure 2.13: Appropriateness of the key length wrt. expiration time. Data point (Expiration Year, Key Length) corresponds to the center of a bubble and the size of the bubble represent a number of data points with the same value. Data points above the recommended key length curve are well chosen, the ones below are not considered to be secure at the time they are used. About half of the trusted certificates have inappropriate key length with respect to their time validity.

2.5.7 Correlation of the Authentication Failure Rate with Other Parameters

To better understand the underlying reasons for the observed certificate deployment, we correlate the authentication failure rate with other parameters such as issuing CAs, subjects' countries, website categories and rank.

Authentication Failure Rate wrt. CAs

Since CAs are only responsible for issuing certificates, not for managing how they are deployed, it might not be fair to correlate authentication success rate to certificates' issuing CAs. Given that the authentication success rate mostly depends on whether a certificate is deployed on a matching domain, it is a responsibility of the organizations who purchased the certificates to properly maintain them and make sure that they allow proper authentication. Nevertheless, it is interesting to compare authentication success rate that is achieved with certificates issued by different CAs (Figure 2.14). We limit our results to those CAs for which we collected at least 4′000 trusted valid certificates.

We observe that certificates issued by GoDaddy, GlobalSign and VeriSign obtain a higher authentication success compared to others. Interestingly, certificates that are signed by root certificates belonging to smaller and perhaps less famous CAs (Equifax, Thawte and UserTrust)[6] have a smaller success rate.

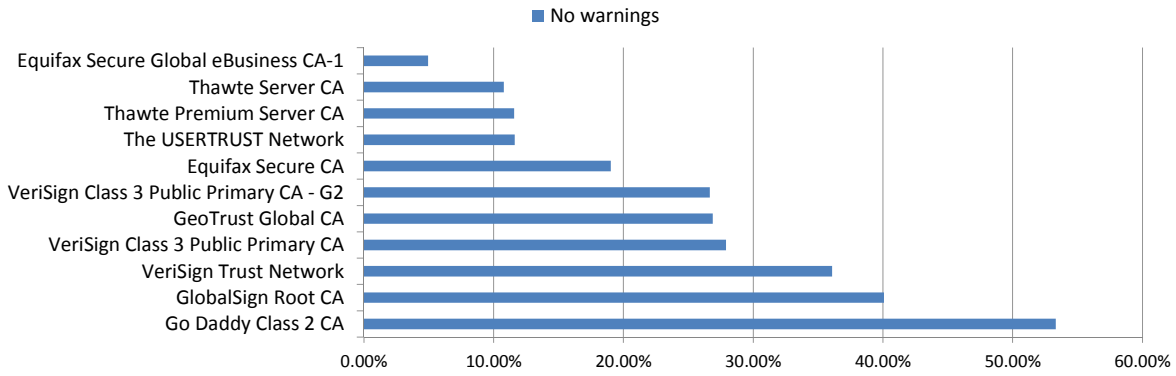


Figure 2.14: Authentication success rate across CAs. Certificates issued by GlobalSign, GoDaddy and VeriSign achieve higher authentication success rate. Either they help their clients manage certificates properly or their customers are more security conscious and resourceful and take better care of their certificates.

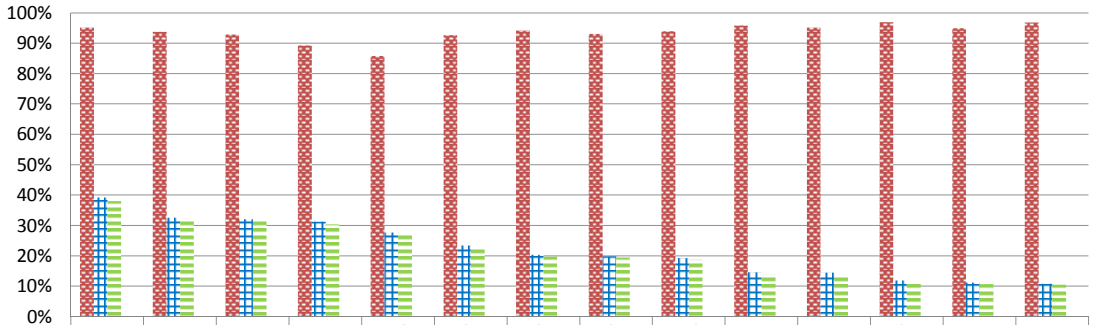
There are different hypotheses to explain this. GlobalSign, GoDaddy and VeriSign are well-established and trusted CAs with major clients. Their certificates are typically more expensive than competitors'. Hence, only resourceful companies can afford to purchase such certificates and these organizations might care more about properly deploying certificates in order to provide good security. On the contrary, less security-conscious website administrators would opt for inexpensive and easier to obtain certificates, that are typically issued by other CAs. Given their lack of incentives, it follows that they might not bother deploying certificates properly. Another possibility is that GlobalSign, GoDaddy and VeriSign only issue certificates after a proper two-step validation process or that they make sure that their customers know how to properly deploy certificates.

Authentication Failure Rate wrt. Countries

We investigate whether organizations from different countries differ in the way they deploy certificates. In Figure 2.15, we show properties of trusted certificates with valid signatures

[6]Even though some CAs (e.g., Equifax and Thawte) were acquired by VeriSign, we refer to them as separate CAs as they offer different products and services and have different policies.

for organizations across several countries. We consider countries for which we observed more than 1′000 certificates. We compute the statistics based on the total number of trusted valid certificates we have collected for each country (the last row in Fig. 2.15). The results confirm that the major reason for authentication failure is due to domain mismatch, as most of the certificates are not expired. Therefore, the total percentage of certificates that do not cause any certificate warnings is dictated by the certificates being properly deployed for the domain they are issued for. We observe that organizations from Japan are most successful in the proper certificate deployment, having successful authentication with 38.1% of certificates. Second best are organizations from Germany with 31.8% of their certificates leading to successful authentication, followed by Netherland with 31.5%. The US is in the middle, having a percentage 18.7% that is closer to the average number observed across the top 1 million websites (16.02%). Poorest deployment practices are in France, Brazil and Switzerland. The major factor for a low authentication success rate among Swiss websites is due to the fact that many of them are hosted by an Internet hosting provider that serves its certificate for each hosted website.



	JP	DE	NL	AU	ES	CA	GB	SE	US	PL	RU	CH	BR	FR
Not expired	95.2%	93.7%	93.0%	89.3%	85.8%	92.7%	94.2%	93.1%	94.0%	95.9%	95.2%	97.0%	95.0%	96.9%
Domain match	39.2%	32.6%	32.1%	31.3%	27.7%	23.5%	20.4%	19.8%	19.3%	14.6%	14.5%	11.9%	11.2%	10.8%
No warnings	38.1%	31.8%	31.5%	30.4%	27.1%	22.5%	19.7%	19.5%	18.7%	14.0%	13.5%	11.4%	10.7%	10.5%
Certificates	11016	8357	1759	2604	1582	3746	12465	1210	65490	2753	1360	1821	3241	4700

Figure 2.15: Certificate validity across countries. Organizations from Japan, Germany and Netherlands have the best, whereas France, Brazil and Switzerland have the poorest practices in deploying certificates. The major reason for authentication failure is due to domain mismatch, as most of the certificates are not expired.

Authentication Failure Rate wrt. Website Categories

If we look at the authentication success across different categories of websites (Table 2.3), firstly we observe that websites from Computer category have a remarkably high percentage 70.25%. Typically sites of technological companies belong to this category and it seems that they have a good know-how and understand the relevance of properly deploying certificates. Reference, Regional and expectedly Business category are also significantly better than the average with more than 40%. It is understandable as Reference sites include University sites, Business websites have e-commerce services and Regional include tech companies such as Google, Yahoo, and Apple. Sports, News, Home and Adults category have the lowest number.

Table 2.3: Certificate deployment across website categories.

Category	Total	Trusted	No Warnings
Computers	121	109 (90.08%)	85 (70.25%)
Reference	133	116 (87.22%)	70 (52.63%)
Business	130	122 (93.85%)	57 (43.85%)
Regional	99	93 (93.94%)	43 (43.43%)
Shopping	129	126 (97.67%)	50 (38.76%)
Recreation	129	105 (81.39%)	45 (34.88%)
Kids and Teens	87	71 (81.60%)	29 (33.33%)
Games	113	87 (76.99%)	35 (30.97%)
Society	126	97 (76.98%)	39 (30.95%)
Arts	75	50 (66.67%)	23 (30.67%)
Science	131	101 (77.09%)	40 (30.53%)
Health	146	115 (78.77%)	41 (28.08%)
Adult	100	61 (61.0%)	26 (26.0%)
Home	103	73 (70.87%)	26 (25.24%)
News	85	64 (75.29%)	18 (21.18%)
Sports	93	71 (76.34%)	13 (13.9%)

Authentication Failure Rate wrt. Websites Ranks

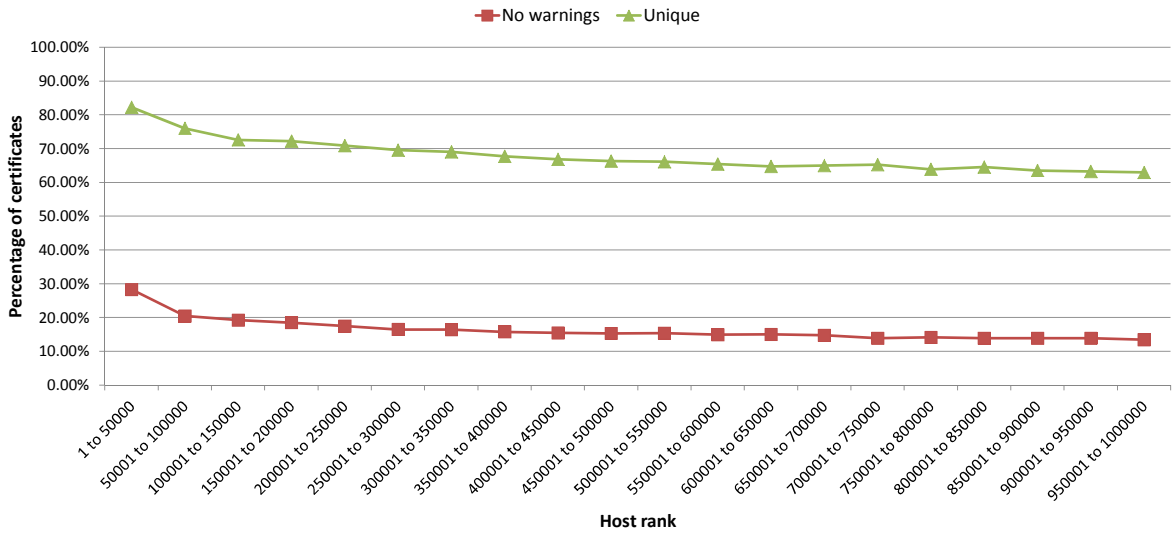
We looked at how the authentication success changes with respect to websites' rank. We divide the ranked 1 million websites into bins of 50′000 websites each, and compute the number of certificates found among those 50′000 websites that allow for a proper authentication and the number of unique certificates (the two plots in Figure 2.16a). The number of certificates with a certain property is expressed in percentages with respect to the total number of certificates in the corresponding bin. We observe that the authentication success is significantly better for the first 50′000 websites and then it decreases for lower ranks. This is expected as popular websites generate more revenue from users' traffic and thus can afford better security practices (or perhaps because better security practices attract more users to these websites). We provide in Table 2.4 a few examples of well ranked websites that suffer from authentication failures.

Table 2.4: Top websites' certificate-authentication implementation failures.

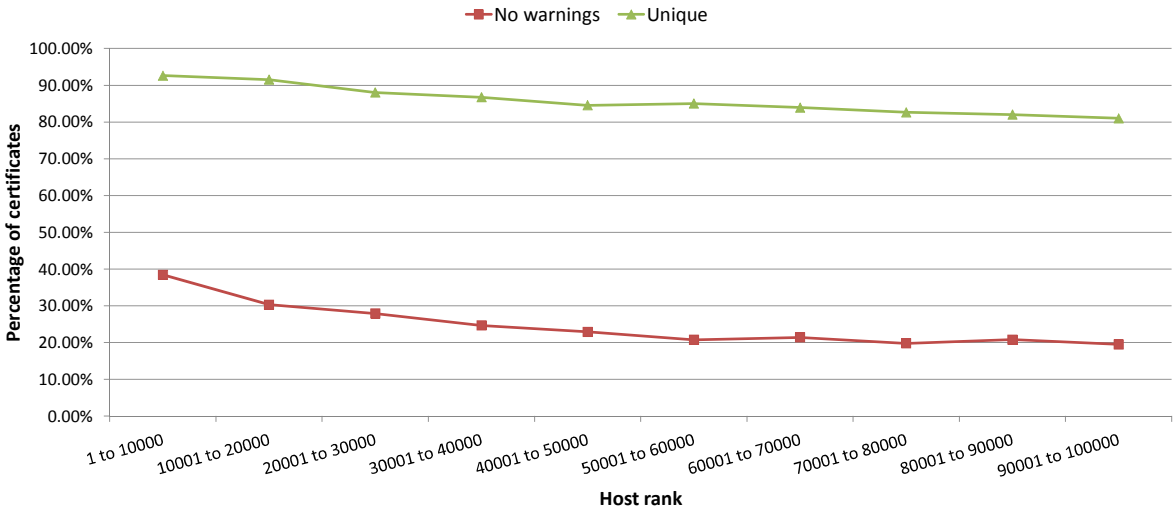
Rank	Host	Cause of failure
31	`fc2.com`	Domain mismatch (CN=`fc2server.com`)
269	`techcrunch.com`	Domain mismatch (CN=`*.wordpress.com, wordpress.com`)
322	`nfl.com`	Domain mismatch (CN=`a248.e.akamai.net, *.akamaihd.net`)
336	`stackoverflow.com`	Domain mismatch (CN=`stackauth.com, *.stackauth.com`)
377	`39.net`	Self-signed & Domain mismatch (CN=`cms.39.net`)
394	`www.informer.com`	Expiration

Given that certificate reuse across domains contributes to domain mismatch and leads to authentication failure, we also found the number of unique certificates. One may notice a strong correlation between the shapes of the two curves, authentication success and unique certificates, which might confirm that indeed certificate reuse across domains is a significant contributor to authentication failure. Since we observe higher dynamics for the highest ranks,

we zoom into the highest 100′000 ranked websites (Figure 2.16b). We draw the same conclusions as for 1 million websites and observe correlations between all the rank, the authentication success rate and the usage of unique certificates.



(a)



(b)

Figure 2.16: Certificate deployment properties vs. website rank: (a) Top 1 million websites and (b) Top100′000 websites. It appears that the proper certificate deployment, in terms of authentication success and use of unique certificates, is correlated to the rank. Higher ranked websites have better practices in implementing certificates properly.

Summary of the Certificate Data Set

Summary of the certificate data set obtained in the survey and used in the analysis is presented in Figure 2.17.

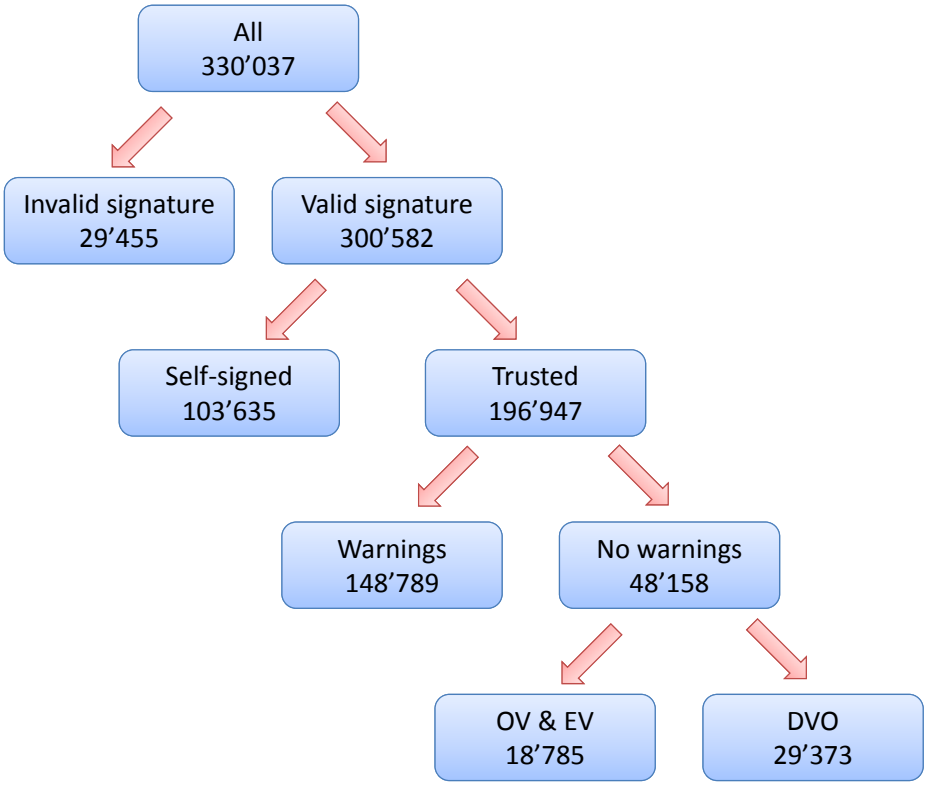


Figure 2.17: Data set of certificates used in the survey.

2.6 Discussion

We outline and interpret most interesting results of Section 2.5 where we obtained several weaknesses of certificate-based authentication leading to security failures. Economic, legal and social reasons might explain these issues.

2.6.1 Failures

Out of top one million websites, about 35% can be browsed via HTTPS. Unfortunately, most of them poorly implement certificate-based authentication and generate authentication problems. *Only about* 48'158 *websites (*16.02% *of the ones with verifiable certificate signatures) have valid certificates*, i.e., certificates issued by trusted CAs, not expired, deployed on domains they are issued for, and with verifiable signatures.

Successful authentication does not necessarily mean that users should trust authenticated websites. CAs increasingly issue domain-validated only certificates, for which they only verify that the applying entity has registered for the requested domain. Such validation process might not guarantee the legitimacy of certificates and lead to man-in-the-middle attacks

(putting users' security at risk). Consequently, users should not systematically trust all websites that their browsers trust. Our results show that 61% of valid certificates are DVO. This reduces the number of websites that users can fully trust to 18′785. Essentially, *only* 5.7% *of the websites that implement HTTPS properly implement certificate-based authentication* and enable users to securely establish HTTPS connections.

Economics

Our investigations showed many domain mismatches were due to improper handling of certificates by websites. Several reasons can explain this, mostly boiling down to *misaligned incentives*. Websites typically offer several services on different subdomains and should obtain a certificate for each of them. This is complex – as it requires technical understanding, and expensive – as it requires obtaining several certificates. Hence, website operators often prefer to reuse a single-domain certificate across a number of (sub)domains, leading to domain mismatches. For example, if people lacking technical know-how (e.g., business managers) were responsible for obtaining certificates, they might focus on cost reduction, whereas people with technical know-how (e.g., engineers) might not invest sufficient time to carefully design certificate-based authentication systems. Certificate management is a cost and does not directly generate revenue compared to other services. Hence, *most website operators have an incentive to obtain cheap certificates*.

CAs are also culprits for authentication failures. CAs' business model depends on the price and the number of certificates sold. From an economic point of view, *CAs have an incentive to distribute as many certificates as possible* in order to increase profit. CAs segment their market and apply differentiate pricing. Consequently, they created different forms of certificates: Domain-validated only certificates, EV certificates and regular certificates.

In our results, we observed that *most website operators choose cheap certificates leading to cheap Web authentication*. Domain-validated only certificates are popular amongst small websites because they are easy and fast to obtain. They require minimum effort from CAs. Several CAs even offer free trials where websites can be certified for free for short periods of time. EV certificates differ from regular certificates and domain-validated only certificates in that they require rigorous verifications. They are a preferred option for large websites dealing with important user information. Information asymmetry plays a large role in pushing cheap certificates. As website operators cannot tell the difference between good and bad security (i.e., market for lemons), they might as well take the cheaper option, thus pushing race to the bottom price.

A positive result is the low number of expired certificates we observed. This is probably because CAs strongly encourage renewal to increase revenue. This shows that *CAs could provide incentives to push proper adoption of certificates*. Yet, most trusted certificates were not deployed properly showing that CAs do not make that investment.

Liability

Liability should be assigned to the party that can best manage risk. Unfortunately, *most CAs transfer their liability onto their customers*. This reduces their risk and involvement in providing security. For example, Verisign License agreement version 5 states that Verisign "Shall not be liable for (i) Any loss of profit, business, contracts, revenue, or anticipated savings, or (ii) any indirect or consequential loss" [37]. It caps to $5000 for total liability for damages

sustained. This exhibits a serious flaw of the system: although CAs distribute an essential element of Web security, they do not take responsibility for it.

In this three-body problem, CAs pass liability to websites that in turn transfer it to users through their website policies. This tendency to push liability to others is another example of misaligned incentives reinforced by information asymmetry. CAs are not encouraged to protect users and rather focus on risk-reducing strategies. This problem appears in other situations, such as security economics of banking [88]. It might be difficult to expect full liability from CAs but a reasonable involvement could dramatically improve the current situation.

Lack of liability is known by economists to generate a moral-hazard effect [88]: As CAs know that customers cannot complain, they tend to be careless in the distribution of certificates, leading to erroneously distributed certificates such as [47].

Reputation

Man-in-the-middle attacks caused by the use of weak certificates can harm websites' and CAs' reputation. Hence, CAs should have an incentive to provide good security. Our results show that *well-established CAs tend to properly issue certificates and rely on a number of less prominent subsidiaries to issue certificates far less rigorously.* This helps preserve their reputation, while not missing good business opportunities. In addition, our results show that CAs tend to provide short-lived certificates, e.g., for one year. This limits the risk of bad publicity in case a malicious website is actually authenticated.

For websites, we observe that mostly large corporations get EV certificates in order to limit risk for their customers. Even if they could afford the cost of a man-in-the-middle attacks, they wish to protect their own reputation and provide good security. *Most less exposed websites select domain-validated only certificates.* In other words, they are fine with cheaper certificates. This could be because website administrators underestimate the value of the data they handle and wish only to reduce security costs. In addition, peer influence from other websites adopting similar weak security practices, could encourage websites administrators to choose domain-validated only certificates.

Usability

For most users, security is secondary as they seek offered services. The variety of options of certificate-based authentication (e.g., domain validated, EV certificates, self-signed certificates and notion of certificates) actually makes it difficult for users to understand the system. Users might misinterpret security warnings as annoyances that prevent them from using Web services. Bad certificate management leads to more security warnings. *The more interruptions users experience, the more they learn to ignore security warnings.* This is counter-productive. Regardless of how compelling, or difficult to ignore SSL warnings are, users could think they are of little consequence because they also see them at legitimate websites [115]. A recent study about SSL warnings' effectiveness shows that users' attitudes and beliefs about SSL warnings are likely to undermine certificates' effectiveness and it suggests to avoid warnings altogether and make security decisions on behalf of users [211].

Finally, *Web browsers have little incentive to limit access to websites* whose certificates are issued by untrusted CAs and thus stop users from accessing websites they could access from other browsers. Firefox currently tries to discourage users from communicating to websites with self-signed certificates by showing users complex warnings. Such approach spurred

agitated debates on the usability of Firefox for Web authentication. In addition, we have seen that unfortunately there are situations (with domain-validated certificates) where users cannot entirely rely on browsers to help them decide whom to trust.

2.6.2 Countermeasures

We observe that proper incentives to secure the certificate-based authentication are missing. The current deployment of digital certificates, mostly based on self-regulation, is moving towards a business model that does not put the emphasis on security and needs a change. We suggest multiple regulation options to modify incentives and improve the situation.

New Third-Parties

An independent third-party could change the current equilibrium of the system. This third-party could be managed by users with an open website (e.g., wiki), by an association of CAs or by Web browsers directly. Basically, such third-party could interfere with the current free-market approach to introduce information related to performances of CAs, and steer the system in a better direction.

This independent third-party could provide transparency by providing information similar to our results about security performances of CAs (Fig. 2.14). This could stimulate competition among CAs to provide better security. CAs would actually have to worry about how certificates are used by websites. Similarly, it could agree with a small set of trusted root CAs, more transparent, hierarchical and localized. Finally, it could also monitor how well websites use certificates and rate websites based on the security they provide.

Users could also run themselves a third-party to form groups of users sharing information with each other. This could reduce the problem of asymmetric information.

New Policies

Changing legal aspects is a difficult and slow process, but can be very effective. It is important that CAs take responsibility for certificate-based authentication. They should be liable for the security of the system as responsibility should follow those that earn revenue. In order to tackle the asymmetric information problem, previous work suggests the use of certification schemes in order to guarantee the quality of provided certificates [164]. Such certificates could be operated by governments (e.g., Orange book) or commercial companies (e.g., Common criteria). However, regulation is costly. One-model-fits-all approach is hard to put in place, especially for smaller companies [130].

Another option is to force websites to be responsible for properly implementing certificate-based authentication. However, websites are customers of the system and it is difficult to blame them for not understanding how to invest money in security.

Finally, Web browsers could pressure CAs in order to improve the quality of CAs' practices. For example, Web browsers could have the policy to trust only the top performing root CAs in terms of provided security.

In general, even though websites generate most authentication failures, we believe that policies should focus on certification authorities and Web browsers.

2.7 Summary

We crawled the top 1 million popular websites and investigated whether they use HTTPS and how they deploy certificate-based authentication. Our results show that nearly one-third of websites can be browsed with HTTPS, but only 18′785 (5.7%) of them properly implement certificate-based authentication. In other words, only 5.7% of the websites that implement HTTPS, do so without causing security warnings in Web browsers and with providing trust in the identities of certificates' owners.

We discuss multiple reasons that might have led to the failure of the current model for Web security. We argue that the current free market approach, where utility-optimizing entities try to maximize profits at minimum cost, is the root of the problem. We can compare the current situation to a *market for lemons*: information asymmetry occurs because CAs know more about certificates and their security features compared to websites and users. Consequently, most website administrators acquire cheap domain-validated only certificates and poorly implement them on their servers. Only a fraction of elite website administrators achieves high security standards by obtaining EV certificates and installing them properly. We also observe strategic behavior of CAs that rely on subsidiaries to sell less trustworthy certificates and maximize profits. This situation is not satisfactory as it affects the global security of the Internet ecosystem. We believe that the right incentives are not in place and suggest multiple policy changes to solve this issue. Notably, we suggest to make CAs liable for the proper certificate usage, Web browsers to trust only top performing CAs, and the creation of an open-source community checking root CAs.

Publication: [223]

Part II

Economics of Online Advertising (In)Security

Chapter 3

Security Games in Online Advertising: Can Ads Help Secure the Web?

Internet Service Providers (ISPs) are an important part of the Internet ecosystem, providing Internet connectivity to the end users. Traditionally, their business model is not based on online advertising. Nevertheless, some ISPs are trying to become part of the online advertising market. Such ISPs either: (i) cooperate with online advertising entities (e.g., ad networks) by providing users' private information to achieve better ad targeting in exchange for a share of the revenue, or (ii) modify the ad traffic on-the-fly such that they divert part of the online advertising revenue for themselves. This is a very important issue because online advertising is at the core of today's business model and it fuels many "free" applications and services. In this chapter, we study the effect of strategic ISPs on the Web using game theory as a tool to analyze mutually dependent actions of ISPs and the current participating entities in online advertising systems, notably ad networks. Our results show that if the users' private information can improve ad targeting significantly and if ad networks do not have to pay a high share of revenue to the ISPs, ad networks and ISPs will cooperate to jointly provide targeted online ads. Otherwise, ISPs will divert part of the online ad revenue for themselves. In that case, if the diverted revenue is small, ad networks will not react. However, if their revenue loss is significant, the ad networks will invest into improving the security of the Web and protecting their ad revenue.

Chapter Outline In Section 3.1 we elaborate on the strategic behavior of ISPs attempting to become part of the online advertising market and the possible consequences for the security of the Web. After a brief presentation of the related work in Section 3.2, we present the system model in Section 3.3 and the various threats and countermeasures in Section 3.4. We present a game-theoretic model with two players, the ISP and the ad network and identify equilibrium outcomes of that game in Section 3.5. In Section 3.6, we provide further analytical refinements of our model and a numerical example to study the practical impact of the obtained results in Section 3.7. We summarize our findings in Section 3.8.

3.1 Introduction

The traditional role of ISPs is to provide Internet access to end users. ISPs are supposed to provide this service by only faithfully forwarding end users' communication, in compliance with the Network Neutrality Policy [108]. Since 2008, several cases of ISPs meddling with users' traffic and violating the Network Neutrality policy have been reported [15, 43, 68, 90]. Reis et al. [188] show that more than 1% of Internet traffic is modified on-the-fly between Web servers and end users. The majority of the modifications are performed on the ad traffic (e.g., ad injection, ad blocking) by ISPs.

Due to their topological position between end users and the Internet, ISPs can observe all the traffic of their end users. Based on the observed traffic, ISPs can extract users' private information, their preferences and interests, and can profile their online behavior. In the EU, to comply with data retention legislations [10, 145], ISPs have to obtain and keep records of their users' activities for a period between six months and two years, and upon request provide them to law enforcement agencies. This directive has imposed a significant burden on ISPs as it increases their storage costs and it requires investing into new technologies for packet inspection (e.g., Deep Packet Inspection [174]). There is no clear answer on how ISPs will obtain a return on that investment.

One possibility for ISPs to generate additional revenue is to take part in the online advertising business. Online advertising is the main business model on the Web today and it generates huge revenues (e.g., $31.7 billion in the US in 2011 [148]). However, ISPs are not part of the traditional online advertising systems. The online ad revenue model includes ad networks, advertisers and Web publishers. In this revenue model, ISPs are bypassed because the only service they provide is to forward the traffic to and from end users. Hence, ISPs might be tempted by the high online ad revenues and might try to become participants in the online advertising business, especially because the user information in their possession could have high commercial value (e.g., due to its unavailability to other online entities). According to observed cases in practice, the behavior of ISPs can be either *cooperative* or *non-cooperative*.

A *cooperative* ISP collects and provides information about users' online behavior with the goal of improving ad targeting. This rich data about users can help better matching ads to users' interests, resulting in higher click-through rates on ads and consequently increasing the ad revenue [71]. Cooperative ISPs generate revenue by charging ad networks for user profiles. There are several examples in practice of ISPs that shared their users' data with ad companies (e.g., Phorm [76]), despite many concerns about the users' privacy [25].

A *non-cooperative* ISP diverts part of online advertising revenues for its own benefit by performing some of the attacks on ads described in Chapter 1. For example, it injects ads into the content of Web pages on-the-fly [43, 188] or replaces legitimate ads with its own [90].

To the best of our knowledge, this is the first in-depth quantitative analysis of ISPs becoming strategic in the online advertising business. We study the effect of strategic ISPs on the Web using game theory as a tool to analyze mutually dependent actions of ISPs and the current participating entities in online advertising systems (e.g., ad networks). Our analysis shows that the outcome of the game between ISPs and ad networks mostly depends on: (i) the value of the users' private information and (ii) the share of the revenue that ad networks offer to ISPs. If the collected users' private information improves ad targeting significantly and ad networks do not have to pay a high price for it to the ISPs, the latter tend to be *cooperative* and they improve the quality of ad targeting jointly with ad networks. Otherwise, ISPs tend

to be *non-cooperative*. Non-cooperative ISPs can divert a very small fraction of clicks from all the websites without causing any reaction from ad networks. However, if ISPs become greedy and divert a high fraction of clicks, ad networks will secure the high value websites first (e.g., by paying for SSL certificates and thus enabling the use of HTTPS), i.e., the websites that generate high volumes of clicks on their Web pages. This means that the significance of the threat creates incentives for ad networks to protect their ad revenues, which could result in improved Web security. Improved Web security would not only benefit ad networks, but websites and users as well, because the security of all the online content, not only ads, would be improved. The results also show that ISPs will probably never try to divert a very high fraction of clicks from very popular websites, as that would cause a higher loss for ad networks, which would then promptly secure the websites and prevent ISPs from obtaining any revenue from those websites.

3.2 Related Work

A strategic role that ISPs can take in online advertising market with a goal of protecting users' privacy has been proposed in [190]. In a mechanism called "transactional privacy", users decide what personal information about themselves is released and put on sale while receiving an adequate monetary compensation for it. Aggregators can purchase access to exploit this information when serving ads to a user. In this approach, a trusted third party is needed to manage a market of personal information: acting as the legal mediator for the users and the aggregators, preventing leakage of users' information, allowing users to put information for sale in a transparent manner, running auction mechanisms, enforcing payments, and handling any issues from users and aggregators. This can be done for a small percentage of the users' revenues and ISPs are obvious candidates for the role. The advantages of ISPs are that they are highly regulated, and users sign a legally binding contract with ISPs for connectivity that can be extended to cover consent and potential exploitation of personal information. ISPs can also control which information goes through the network. In our work, we consider cooperative ISPs that are potentially willing to collect and trade users' personal information in exchange for a fraction of ad revenue, remunerated by ad networks. Our model can be extended to include ISPs offering a new service to its customers, notably withholding or trading only user-approved personal information on behalf of its customers.

A line of research relevant to our work is research on fraud in online advertising which is mostly focused on click fraud [110, 128, 154]. Many problems that stem from online advertising and security gaps, especially the consequences for the end users, are addressed in [117]. The context, mechanisms and processes associated with the click-fraud industry are analyzed from an economics point of view in [163]. Economics of click fraud are also briefly addressed in [154]. In [95], the economic analysis based on a game-theoretic model of the online advertising market, shows that ad networks that deploy effective algorithms for click fraud detection gain a significant competitive advantage. Similarly, [92] presents a game-theoretic model of click fraud in a publisher network that sheds light on the economic trade-offs search engines face and shows that search engines have incentives to invest in technology that filters invalid clicks. Monitoring and filtering tools are found to have a central role in advertisers' perceived benefits of online advertising which influences their attitude and trust towards search engine providers and their intention to advertise online [112]. If it is the case that some ad networks do not fight click fraud, mechanisms are proposed in [118] to protect online advertisers from being

charged for fraudulent clicks. In comparison, our model does not address click fraud but focuses on the economics of fighting ad fraud and introduces a new strategic player - the ISP - in addition to the traditional players in online advertising (i.e., ad networks, advertisers and publishers). Our results show that this player can yield significant implications for the security of the Internet.

Also related to our work is research on finding the right incentives to increase the security of the Internet. There are several contributions in the literature. Part of the research focuses on how risk management and cyberinsurance could be used as a tool for security management [99, 138, 198]. The game-theoretic approach of [139] on strategic security investment models how users choose between investments in security (e.g., firewalls) or insurance (e.g., backup) mechanisms. The positive effect of cyberinsurance on the investment of agents in self-protection is analyzed using a game-theoretic model in [168]. The main conclusion of this work is that cyberinsurance is not a good incentive for self-protection without regulation. Another line of work proposes a centralized certification mechanism to encourage ISPs to secure their traffic and analyzes the resulting scheme using game theory [239]. In contrast to these works, our analysis shows that Internet security can be increased, under given conditions, without any central oversight and thanks to self-interested decisions by only a few key players (namely, the ad networks).

3.3 System Model

We consider a system consisting of the online advertising system and an access network (i.e., an ISP), as depicted in Figure 1.1 (page 9).

3.3.1 Online Advertising Systems

We briefly overview the ad serving systems that were presented in detail in Chapter 1. To have their ads appear with the appropriate Web content, Advertisers (AV) subscribe with an Ad Network (AN) whose role is to automatically embed ads into Web pages. Ad networks have contracts with publishers (e.g., websites (WS)) that want to host advertisements. When a User (U) visits such a website, while downloading the content of the Web page, the user's browser will be directed to communicate with one of the Ad Servers (AS) belonging to the ad network. The ad server chooses and serves the most appropriate ads to the user, such that users' interests are matched and the potential revenue is maximized. Throughout the rest of the chapter, we use the terms "ad network" and "ad server" (that belongs to the ad network) interchangeably. We also use the terms "user" and "user's browser" interchangeably.

A user-generated click on an advertisement directs the user's browser to the advertised website and is called a *clickthrough*. The event of a click-through being followed by a predefined users' action on the advertiser's website (e.g., online purchase or registration for a newsletter) is called a *click conversion*. We consider the pay-per-click and pay-per-action revenue models, in which an advertiser pays a certain amount of money to the ad network whenever a clickthrough or a click conversion, respectively, on an advertisement occurs. The ad network gives a fraction of the ad revenue to the website that hosted the ad on which a clickthrough or a click conversion occurred. Throughout the rest of the chapter, we use the term "clicks" to refer to the user-generated clicks on ads that create ad revenue for the ad network and the associated website.

The ad network values an associated website based on the volume of clicks and ad traffic generated by the website's visitors clicking on hosted ads. Popular websites that attract a great number of visitors generate more clicks on ads, thus also create a high ad revenue for the associated ad network and themselves.

3.3.2 Internet Service Providers (ISPs)

Traditionally, an ISP provides Internet access to end users and is topologically placed between users and the Internet. We say that the system operates in the *nominal mode* when the ISP only faithfully forwards users' traffic. However, to capture the recent behavior of ISPs, in our system model the ISP can also either take advantage of the users' private information and operate alone as an ad network offering higher quality clicks to the set of its advertisers (*non-cooperative behavior*) or cooperate with ad networks by sharing users' private information to jointly improve ad targeting (*cooperative behavior*).

3.4 Threats and Countermeasures

Given that ISPs are in the position to observe all the traffic of their subscribers and that recently they had to invest in technologies that enable profiling of their subscribers' online behavior, ISPs can collect a high volume of users' private data. Such a rich data would be of immense value for ad networks as it can improve the quality of matching ads to users' interests [71]. Consequently, ad networks could generate even higher ad revenues. Ad networks are already deploying mechanisms (e.g., third-party cookies) to track users' interests. However, the collected information cannot be as rich as the ISPs are able to obtain, because ISPs have access to all the users' traffic (unless it is encrypted). Thus, ad networks might be willing to subsidize ISPs to profile users' online behavior in exchange for a share of ad revenue. When the ISP and the ad network are cooperative the system operates in the *cooperative mode*.

Some ISPs might gain more revenue when being non-cooperative. A non-cooperative ISP plays a role similar to the role of ad networks: it uses the obtained information about users' interests and performs advertising services for a set of its own advertisers. As the ISP is the last hop in forwarding the traffic towards its subscribers, it can free-ride on the existing traffic to deliver ads of its choice to the end users. The ISP can simply perform inflight modifications of the content of Web pages between servers and users with the goal of modifying the original ads or injecting new ads. Another technique is for the ISP to replace entire Web pages by modifying users' DNS traffic on-the-fly and redirecting users to servers of the ISP's choice.[1] Thus, the affected users would see altered ads, which are different from the original ads embedded into the webpages by a legitimate ad networks associated to the browsed website. When users click on the altered ads, the clicks generate revenue for the ISP instead of the ad network and we say that the ISP has diverted the clicks from the ad network. Consequently, the non-cooperative ISP diverts a part of the ad revenue from the ad network. When the ISP is non-cooperative and diverts clicks (i.e., ad revenue) from the ad network the system operates in the *non-cooperative mode*.

Depending on the ad network's loss of ad revenue caused by the ISP diverting clicks, the ad network might decide to deploy a countermeasure and prevent exploits by the non-cooperative ISP. A straightforward solution to prevent inflight modifications is to deploy HTTPS instead of

[1] However, in this case the websites might detect the decrease in the number of visits and become suspicious.

HTTP to deliver Web content and ads. HTTPS provides data integrity and in case encryption is used would also reduce the amount of information ISPs can collect about users. Given the system architecture (Figure 1.1, page 9), data integrity is necessary in both communication channels[2]: (i) between users and websites and (ii) between users and ad servers. So far, HTTPS and certificate-based authentication is used properly only by a small fraction of websites, as presented in detail in Chapter 2. Therefore, if the ad network wants associated websites to properly deploy HTTPS and certificate-based authentication, it should help the websites and cover the costs itself. The major part of costs of implementing HTTPS at a Web server is the cost of obtaining and properly deploying a valid X.509 authentication certificate[3]. Deploying HTTPS at ad servers is easy as they typically belong to major companies that already have valid authentication certificates and the know-how to properly deploy them.

Typically, users ignore security warnings related to certificate-based authentication failures because a high number of websites does not deploy it properly. However, if websites associated with an ad network agree to all use valid certificates and the ad network helps with a proper deployment, browsers can differentiate between: (i) the case of an authentication failure due to a website's improper certificate deployment and (ii) the case when an adversary tampers with a valid certificate or the content of a website. Consequently, Web browsers can deploy more sophisticated policies in handling associated security risks in these two cases and can display specifically targeted security warnings that alert users to not accept the content that has been altered by the adversary.

Each website maximizes its revenue by choosing an ad network whose ads it will host. A website can be associated with the ad network or with the ISP. This association is known, as the website has a contract with the associated ad network. If the website has willingly decided to associate with the ISP then the website's ad revenue is not affected by the deviating behavior of the ISP. The concerned websites are the ones that have chosen to host the ads of the ad network, but due to the actions of the non-cooperative ISP, the website's Web pages are displayed with ads of the ISP. Consequently, the website loses the ad revenue.

When the website that is originally associated with the ad network is affected by the non-cooperative behavior of the ISP, it can only decide whether to accept to deploy HTTPS or not. As explained, the major cost of deploying HTTPS instead of HTTP at the Web server is the cost of a certificate. If this cost is paid by the ad network, then the remaining costs (e.g., per transaction computational and communication overhead of HTTPS compared to HTTP) are negligible compared to the ad revenue. Thus, in the presence of the non-cooperative ISP, if the ad network is willing to bear the costs, the website's revenue is maximized when it accepts to deploy HTTPS together with the ad network. Since the ad network bears the costs, we say that the ad network *secures* the website.

3.5 Game-theoretic Model

We propose a game-theoretic model of the relationship between an ISP and an ad network. The strategic decision facing an ISP is to be cooperative or not with the ad network. In the case of a cooperative ISP, an ad network can offer a share of its revenue in exchange

[2]Only data integrity property of HTTPS is necessary, encryption is optional.

[3]Data integrity can be provided with Message Authentication Codes which are cheap in terms of computation and communication overhead. Thus, the per transaction cost of serving content over HTTPS instead of HTTP is negligible compared to the ad revenue.

for the users' private information based on which it improves ad targeting. In the case of a non-cooperative ISP, an ad network can deploy security mechanisms to prevent the ISP from diverting the revenue. We study within this model the possible outcomes of this tension between the ISPs and ad networks.

3.5.1 Actions

We denote the two entities, an ISP and an Ad Server (representing an ad network), as players ISP and AS, respectively. We model the behavior of ISP with the following three actions:

Divert (D): ISP diverts from AS a fraction m of the clicks generated at a website WS associated with AS. In practice, this means that ISP modifies the traffic on-the-fly. ISP diverts the revenue from AS because the diverted clicks are not associated with AS, thus it cannot charge advertisers for those clicks. This action models the *non-cooperative* behavior of ISP.

Cooperate (C): ISP shares with AS the collected private information about users in order to help AS improve the quality of ad targeting. In return, it receives from AS a share of the generated revenue. This action models the *cooperative* behavior of ISP.

Abstain (A): ISP takes no action. This models the traditional behavior of ISP when it operates in the nominal mode.

The player AS can choose between the following three actions:

Abstain (A): AS does not react to the changed behavior of ISP. This models the traditional behavior of AS operating in the nominal mode.

Cooperate (C): AS cooperates with ISP by providing a share of its revenue in exchange for the users' private information.

Secure (S): AS secures a given website to prevent the ISP from diverting clicks. The one-time cost (C_{ss}) of securing the website depends on the secure solution that is implemented. Our model applies, in general, to all solutions in which the ad network pays a per website one-time cost (C_{ss}) to secure ad serving. In the case of HTTPS, AS can buy a digital certificate from a Certification Authority (e.g., VeriSign) thus enabling the WS to communicate with users over the HTTPS protocol. HTTPS provides integrity and authenticity of the content, hence preventing ISP from meddling with users' traffic.

3.5.2 The Game

We model the problem as a dynamic, finite multi-stage game with perfect and complete information between AS and ISP. We assume that AS can detect inflight modifications of the ad traffic using mechanisms such as Web tripwires [188] and ISP can observe if HTTPS has been deployed at a given WS or not, hence it is a game with perfect information. The game consists of n stage games, where each stage game is an extensive-form game in which ISP plays first and AS plays second. This models the behavior observed in practice, where ISPs act first by taking part in the online advertising business and then the AS can react. We model the game as a finite game because business relationships usually have a finite duration. The length of the business relationship, known to the players, determines the value of n. If the website is not secured, in each stage game ISP chooses among the actions $\{D,C,A\}$ and then AS chooses among the actions $\{A,C,S\}$, as illustrated in Figure 3.1a. If AS secures the website at some stage of the multi-stage game, ISP cannot divert clicks until the end of the

game and AS cannot secure the website again. Thus, in all of the following stages, if the website is secured the single stage game is as illustrated in Figure 3.1b.

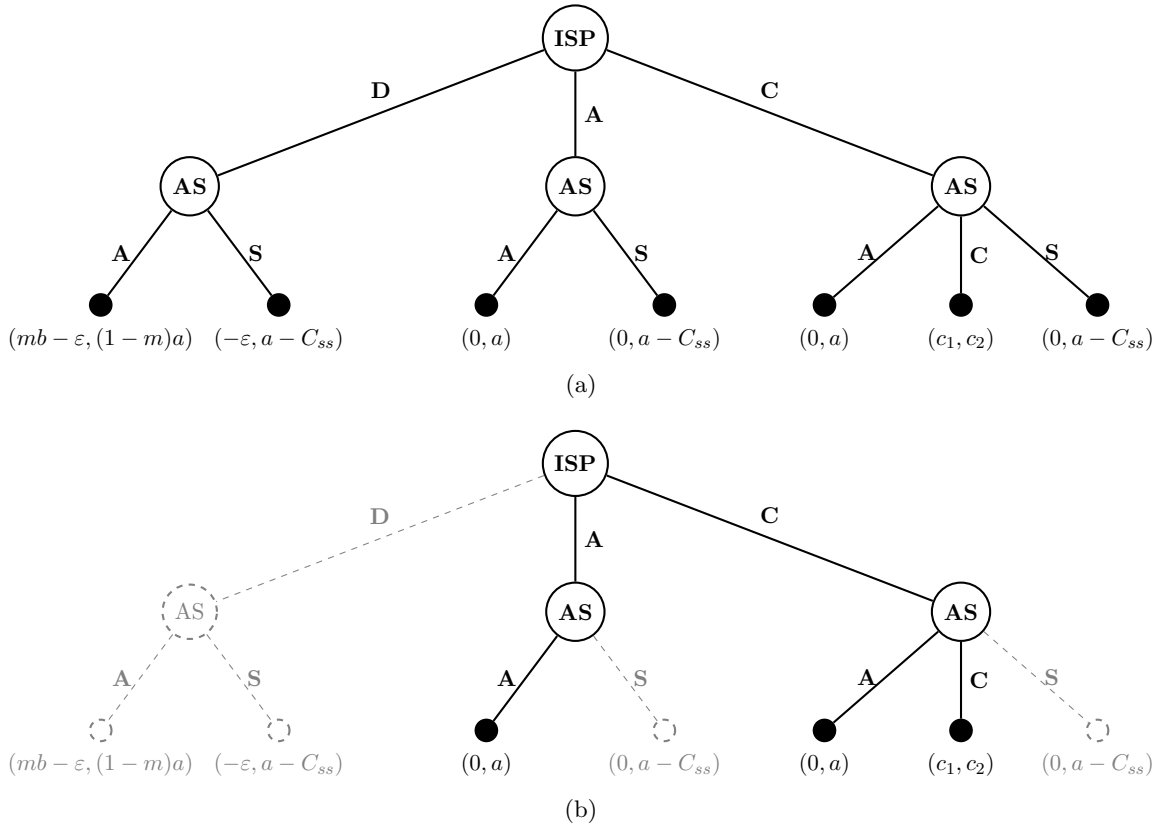


Figure 3.1: Extensive-form single stage games. ISP always plays first. (a) Single stage game if a website is not secured. (b) Single stage game if a website is secured: actions colored gray cannot be played anymore.

Note that in the model, we consider clicks on ads generated by ISP's subscribers at a single website. The results are extended to the case of multiple websites in Section 3.7. The website is not modeled as a player in the game because its revenue is maximized when the ad network's revenue is maximized. As explained in Section 3.4, the website always complies with a decision (to deploy HTTPS or not) that is made by the associated ad network. The symbols used in the model are given in Table 3.1.

3.5.3 Analytical Analysis and Results

In this section, we first explain the single stage games presented in Figure 3.1 and then we present the outcome of the multi-stage game.

In a stage game, when ISP plays A and it is not part of the online advertising system, AS earns the nominal revenue a and ISP earns nothing. This corresponds to the case when both players play A and it represents the system operating in the nominal mode (i.e., when ISP only faithfully forwards the traffic). Thus, the payoffs of ISP (u_{ISP}) and AS (u_{AS}) are $(u_{ISP}, u_{AS}) = (0, a)$.

Cooperation only emerges if both players are willing to cooperate, i.e., if they both play

Table 3.1: Table of symbols for the game-theoretic model.

Symbol	Definition
m	Fraction of clicks ISP diverts
ε	Cost of diverting clicks
u_{AS}	Ad Server's total payoff
u_{ISP}	ISP's total payoff
a	Ad Server's total payoff in the nominal model
b	ISP's per fraction revenue when diverting clicks
c_1	ISP's total payoff in the cooperation model
c_2	Ad Server's total payoff in the cooperation model
C_{ss}	One-time cost of securing a website
n	Number of stages of the multi-stage game
k_1	Stage at which AS secures the website
k_2	Stage at which ISP starts diverting clicks

C. Therefore, AS can choose action C only if ISP has played C. Let the corresponding payoffs in case of cooperation be $(u_{ISP}, u_{AS}) = (c_1, c_2)$.

If ISP plays D followed by AS playing A, a fraction m of the clicks is successfully diverted, which brings revenue mb to ISP. ISP has to pay a small cost (ε) in every stage to divert clicks due to resources invested in mounting and performing attacks (e.g., parsing the code of a Web page, identifying ads and replacing or injecting ads).[4] Therefore, ISP's payoff is $mb - \varepsilon$. When the diversion of a fraction m of clicks is successful, AS loses a part of its revenue proportional to the fraction of clicks being diverted, ma. Thus, the payoffs when ISP successfully diverts clicks from AS are $(u_{ISP}, u_{AS}) = (mb - \varepsilon, (1 - m)a)$.

AS can decide to play S to prevent the loss of its revenue. AS has to pay a one-time cost C_{ss} which makes its payoff $a - C_{ss}$ in the stage when it secures the WS. After securing the website, AS does not have to pay any other costs and it secures its nominal revenue a in all future stages. Depending on whether ISP has tried to divert clicks or not in the stage game when AS implements security, it either has a cost ε or not, which corresponds to payoffs $(u_{ISP}, u_{AS}) = (-\varepsilon, a - C_{ss})$ and $(u_{ISP}, u_{AS}) = (0, a - C_{ss})$, respectively.

To solve the finite multi-stage game with perfect information, we apply *backward induction* to determine the *Subgame Perfect Nash Equilibrium (SPNE)* of the game [126]. A strategy profile is a SPNE if it represents a Nash equilibrium of every subgame of the original game. The game outcome depends on the values of several parameters of the model. We perform an exhaustive analysis for all the possible values of the model parameters. There are five cases:

[4] In practice, the cost ε might not be exactly the same in each stage of the game. However, the variations are insignificant and since ε is negligible compared to the ad revenue, assuming a constant cost per stage does not influence the results.

$$\text{Case 1} : ma \geq C_{ss} \text{ and } c_2 > a$$
$$\text{Case 2} : ma \geq C_{ss} \text{ and } c_2 \leq a$$
$$\text{Case 3} : ma < C_{ss} \text{ and } c_2 \leq a$$
$$\text{Case 4} : ma < C_{ss} \text{ and } c_2 > a \text{ and } c_1 \geq mb - \varepsilon$$
$$\text{Case 5} : ma < C_{ss} \text{ and } c_2 > a \text{ and } c_1 < mb - \varepsilon$$

In practice, the values of the parameters can be estimated by each of the players and they determine to which of the five cases of the model the system corresponds to.

Next, we present the results for each of the cases. We first focus on the outcomes of the SPNE and then present the full SPNE strategy sets and proofs.

Result 1*: In Case* 1*, there is a unique SPNE where the outcome is (Cooperate,Cooperate) in every stage game and the corresponding total payoffs, summed over n stages, are:*

$$\begin{aligned} u_{ISP} &= nc_1 \\ u_{AS} &= nc_2 \end{aligned} \tag{3.1}$$

In Case 1, if ISP diverts a large fraction ($m \geq \frac{C_{ss}}{a}$) of clicks, the best response of AS is to implement security because the cost of deploying a secure protocol (C_{ss}) is smaller than the loss of revenue due to the diversion of clicks ($ma \geq C_{ss}$). If AS implements security, ISP does not earn any revenue and it only pays the cost of mounting the attack, $u_{ISP} = -\varepsilon$. Therefore, it is better for ISP either to abstain, in which case its payoff would be $u_{ISP} = 0$, or to offer cooperation, in which case its payoff would be $u_{ISP} = c_1$ if AS accepts the cooperation. Thus, in Case 1, cooperation is the best action for ISP. Whether ISP and AS cooperate now depends on the action of AS. In Case 1, cooperation is also more profitable for AS ($c_2 > a$), hence AS accepts cooperation.

Result 2*: In Case* 2*, there is a unique SPNE where the outcome is (Cooperate,Abstain) in every stage game and the corresponding total payoffs, summed over n stages, are:*

$$\begin{aligned} u_{ISP} &= 0 \\ u_{AS} &= na \end{aligned} \tag{3.2}$$

As $m \geq \frac{C_{ss}}{a}$ holds in Case 2 as in Case 1, the best action for ISP is to offer cooperation, as explained for Case 1. However, in Case 2 AS obtains a higher revenue when operating alone than when cooperating with ISP ($a \geq c_2$), thus AS does not accept cooperation and the system operates in the nominal mode in every stage game.

Result 3.1*: In Case* 3*, if* $m < \frac{C_{ss}}{na}$*, there is a unique SPNE where the outcome is (Divert,Abstain) in every stage game and the corresponding total payoffs, summed over n stages, are:*

$$u_{ISP} = n(mb - \varepsilon)$$
$$u_{AS} = n(1-m)a \tag{3.3}$$

If ISP diverts such a small fraction $m < \frac{C_{ss}}{na}$ of clicks as in Result 3.1, the loss of revenue it imposes to AS is not significant enough to cause AS to secure the website, i.e., the cost of a secure solution exceeds the revenue loss. Therefore, ISP diverts a fraction m of clicks in all stages and AS does not react.

Result 3.2: *In Case* 3, *if* $\frac{C_{ss}}{na} \leq m < \frac{C_{ss}}{a}$, *there are two SPNE that result in two different outcomes. The first outcome is (Divert,Abstain) in the first* k_1 *stage games, where* $k_1 = \lfloor \frac{\varepsilon}{mb-\varepsilon} \rfloor$ *and* $0 < k_1 < n$, *(Divert,Secure) in the stage game* $k_1 + 1$ *and (Abstain,Abstain) till the end. The corresponding total payoffs, summed over* n *stages, are:*

$$u_{ISP} = k_1(mb - \varepsilon) - \varepsilon$$
$$u_{AS} = k_1(1-m)a + a - C_{ss} + (n - k_1 - 1)a \tag{3.4}$$

The second outcome is (Abstain,Abstain) in the first k_2 *stage games, where* $k_2 = \lceil \frac{nma - C_{ss}}{ma} \rceil$ *and* $0 < k_2 < n$, *and (Divert,Abstain) in the last* $n - k_2$ *stage games. The corresponding total payoffs, summed over* n *stages, are:*

$$u_{ISP} = (n - k_2)(mb - \varepsilon)$$
$$u_{AS} = k_2 a + (n - k_2)(1-m)a \tag{3.5}$$

Result 3.2 means that if ISP wants to divert a high fraction of clicks, i.e., $\frac{C_{ss}}{na} \leq m < \frac{C_{ss}}{a}$, it cannot do so in all stages but only in a limited number of stages of the game. The two outcomes show that ISP has two options to divert clicks. In the first outcome, ISP diverts clicks in the first k_1 stage games, which causes AS to secure the website in the stage game $k_1 + 1$ because the loss of revenue for AS is higher than the cost of deploying the secure protocol. In the remaining stages, ISP cannot divert clicks and there is no cooperation, as AS earns more when operating alone ($a \geq c_2$), hence the system operates in the nominal mode. The second outcome shows that ISP has another possibility to divert clicks and avoid AS securing the website. If ISP abstains in the first k_2 stage games, it can then divert clicks in the remaining $n - k_2$ stage games till the end, with a fraction $m < \frac{C_{ss}}{(n-k_2)a}$. Intuitively, ISP can divert clicks in a larger number of stage games but with a smaller fraction, or for a smaller number of stage games but with a larger fraction.

Result 4: *In Case* 4, *there is a unique SPNE where the outcome is (Cooperate,Cooperate) in every stage game and the corresponding total payoffs are given by (3.1).*

In Case 4, as both AS and ISP earn more when cooperating than in any other mode ($c_2 > a$ and $c_1 \geq mb - \varepsilon$), their best actions are to always cooperate.

Result 5.1: *In Case* 5, *if* $m < \frac{(n-1)(a-c_2)+C_{ss}}{na}$, *there is a unique SPNE where the outcome is (Divert,Abstain) in every stage game and the corresponding total payoffs are given by (3.3).*

The result shows that when ISP diverts a small fraction of clicks the loss of revenue for AS is not significant enough to invest in securing the WS.

***Result 5.2**: In Case 5, if $\frac{(n-1)(a-c_2)+C_{ss}}{na} \leq m < \frac{C_{ss}}{a}$, there are two SPNE that result in two different outcomes. The first outcome is (Divert,Abstain) in the first k_1 stage games, where $k_1 = \lfloor \frac{\varepsilon+c_1}{mb-\varepsilon-c_1} \rfloor$ and $0 < k_1 < n$, (Divert,Secure) in the stage game $k_1 + 1$ and (Cooperate,Cooperate) till the end. The corresponding total payoffs, summed over n stages, are:*

$$\begin{aligned} u_{ISP} &= k_1(mb - \varepsilon) - \varepsilon + (n - k_1 - 1)c_1 \\ u_{AS} &= k_1(1 - m)a + a - C_{ss} + (n - k_1 - 1)c_2 \end{aligned} \tag{3.6}$$

The second outcome is (Cooperate,Cooperate) in the first k_2 stage games, where $k_2 = \lceil n - \frac{C_{ss}-a+c_2}{ma-a+c_2} \rceil$ and $0 < k_2 < n$, and (Divert,Abstain) in the last $n - k_2$ stage games. The corresponding total payoffs, summed over n stages, are:

$$\begin{aligned} u_{ISP} &= k_2c_1 + (n - k_2)(mb - \varepsilon) \\ u_{AS} &= k_2c_2 + (n - k_2)(1 - m)a \end{aligned} \tag{3.7}$$

Result 5.2 shows that, as in Case 3, if ISP wants to divert a higher fraction of clicks it has two possibilities: (i) divert in the first k_1 stage games (the first outcome), or (ii) divert in the last $n - k_2$ stage games (the second outcome). The difference between the outcomes in Cases 3 and 5 is that when in Case 3 the system operates in the nominal mode, in Case 5 AS and ISP cooperate. For ISP, cooperation is always better than operating in the nominal mode when it earns nothing. However, AS benefits more when operating alone than when cooperating ($a \geq c_2$) in Case 3, so it does not agree to cooperate. In Case 5 cooperation is more profitable ($c_2 > a$), hence AS agrees to cooperate.

The obtained outcomes of the multi-stage game for all the possible cases of parameters are presented in Table 3.2. Each column corresponds to a SPNE of the multi-stage game and each row corresponds to the achieved outcomes in each stage of the multi-stage game. Note that stages k_1 and k_2 are different in Case 3.2 and Case 5.2 and can be calculated with the expressions presented in Result 3.2 and Result 5.2. For the simplicity of presentation we abstract this in Table 3.2 and use the same symbols k_1 and k_2 for the both cases.

Proof. We use induction to prove that the payoff expressions in Section 3.5.3 hold for any $n \geq 1$. Next, we apply backward induction to these payoffs to solve the multi-stage game of n stages. The backward induction algorithm constructs a SPNE in finite games of perfect information [126]. We only present proofs of the results for Case 3 as they are more complex. Results for Case 5 can be proven in the same way as for the Case 3. Proofs for Cases 1, 2 and 4 are trivial.

Applying backward induction to the single stage game (Figure 3.1a) in Case 3 results in a unique SPNE with the strategy (D,AAA). The corresponding total payoffs in the game outcome, (Divert,Abstain), are: $((mb - \varepsilon), (1 - m)a)$.

To prove the payoff expressions for the n stage game, we prove that they hold for a single stage, we assume they are true for j stages and prove that they hold for $j + 1$ stages. We assume the relevant subgames (denoted by SG) and the respective payoffs in the multi-stage game with j stages:

Table 3.2: Outcomes of the multi-stage game.

Stage	**Case 1**	**Case 2**	**Case 3.1**	**Case 3.2**		**Case 4**	**Case 5.1**	**Case 5.2**	
1	(C,C)	(C,A)	(D,A)	(D,A)	(A,A)	(C,C)	(D,A)	(D,A)	(C,C)
2	(C,C)	(C,A)	(D,A)	(D,A)	(A,A)	(C,C)	(D,A)	(D,A)	(C,C)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
k_1	(C,C)	(C,A)	(D,A)	(D,A)	(A,A)	(C,C)	(D,A)	(D,A)	(C,C)
k_1+1	(C,C)	(C,A)	(D,A)	***(D,S)***	(A,A)	(C,C)	(D,A)	***(D,S)***	(C,C)
k_1+2	(C,C)	(C,A)	(D,A)	***(A,A)***	(A,A)	(C,C)	(D,A)	***(C,C)***	(C,C)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
k_2	(C,C)	(C,A)	(D,A)	(A,A)	(A,A)	(C,C)	(D,A)	(C,C)	(C,C)
k_2+1	(C,C)	(C,A)	(D,A)	(A,A)	***(D,A)***	(C,C)	(D,A)	(C,C)	***(D,A)***
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	(C,C)	(C,A)	(D,A)	(A,A)	(D,A)	(C,C)	(D,A)	(C,C)	(D,A)

SG_1 : $(j(mb-\varepsilon), j(1-m)a)$, which corresponds to ISP successfully diverting clicks in all stage games;

SG_2 : $(k_1(mb-\varepsilon)-\varepsilon, k_1(1-m)a+a-C_{ss}+(j-k_1-1)a) = (k_1(mb-\varepsilon)-\varepsilon, (j-k_1m)a-C_{ss})$, $0<k_1<j$, which corresponds to ISP successfully diverting clicks in the first k_1 stage games, resulting in AS implementing security in the stage game k_1+1, followed by the system operating as in the nominal mode till the end;

SG_3 : $((j-k_2)(mb-\varepsilon), k_2a+(j-k_2)(1-m)a) = ((j-k_2)(mb-\varepsilon), (k_2m+j(1-m))a)$, $0<k_2<j$, which corresponds to the system operating as in the nominal mode in the first k_2 stage games, followed by ISP diverting clicks till the end.

If we set $j=1$ in SG_1 (SG_2 and SG_3 do not exist in this case), we obtain the outcome of a single stage game. Now let us extend the j stage game with an additional stage game and solve the multi-stage game of $j+1$ stage games. For all subgames where the security was not implemented, in the unique SPNE in the $j+1$st stage game the outcome is (Divert,Abstain). Therefore, we add the payoffs $(\Delta u_{ISP}, \Delta u_{AS}) = (mb-\varepsilon, (1-m)a)$ to the payoffs of ISP and AS obtained after j stage games. In the subgames where security has been implemented, in the unique SPNE in the $j+1$st stage game the outcome is (Abstain,Abstain). We add the payoffs $(\Delta u_{ISP}, \Delta u_{AS}) = (0, a)$ to the payoffs of ISP and AS obtained after j stage games.

The obtained payoffs after $j+1$ stage games are:

SG_1 : $(j(mb-\varepsilon)+(mb-\varepsilon), j(1-m)a+(1-m)a) = ((j+1)(mb-\varepsilon), (j+1)(1-m)a)$;

SG_2 : $(k_1(mb-\varepsilon)-\varepsilon+0, k_1(1-m)a+a-C_{ss}+(j-k_1-1)a+a) =$
$(k_1(mb-\varepsilon)-\varepsilon, (j+1-k_1m)a-C_{ss})$, $0<k_1<j+1$;

SG_3 : $((j-k_2)(mb-\varepsilon)+(mb-\varepsilon), k_2a+(j-k_2)(1-m)a+(1-m)a) =$
$((j+1-k_2)(mb-\varepsilon), (k_2m+(j+1)(1-m))a)$, $0<k_2<j+1$.

Observe that the payoffs of the game with $j+1$ stages can be obtained by replacing j with $j+1$ in the payoffs of the game with j stages. As this also holds for $j=1$, these payoffs hold for any j by induction.

Now we can solve the game with n stages. Applying backward induction to this game, we obtain three SPNE:

- For $m < \frac{C_{ss}}{(n-k_1)a}$, there is a unique SPNE that corresponds to the outcome of the SG_1, where ISP always diverts clicks. The SPNE strategy set is (D,AAA) in every stage game.

- For $\frac{C_{ss}}{(n-k_1)a} < m < \frac{C_{ss}}{a}$, there are two SPNE. In the first SPNE, that corresponds to SG_2, ISP diverts clicks for k_1 stage games, where $k_1 > (n-k_2) + \frac{\varepsilon}{mb-\varepsilon}$ and $0 < k_1 < n$, AS secures the website in k_1+1st stage game and the system operates as in the nominal mode till the end. The SPNE strategy set in the first k_1 stage games is (D,AAA), in the stage game k_1+1 the strategy set is (D,SAA), and in every stage game till the end the strategy set is (A,AAA). In the second SPNE, that corresponds to the outcome of the SG_3, the system operates as in the nominal mode for the first k_2 stage games, where $k_2 = \lceil \frac{nma-C_{ss}}{ma} \rceil$ and $0 < k_2 < n$, followed by ISP diverting clicks till the end. The SPNE strategy set in the first k_2 stage games is (A,SAA) and (D,AAA) in the last $n-k_2$ stage games.

To obtain the results in Section 3.5.3, we derive the values of k_1 and k_2 as follows. In SG_1 and SG_3, AS does not implement security, therefore $k_1 = 0$. In SG_2, as ISP does not divert clicks after AS implements the secure solution, we need to set $k_2 = n$. In SG_3, the choice of k_2 is determined by the threat of AS implementing security. In a given stage game, AS compares the cost of securing a website:

$$(n-k_1)a - C_{ss}$$

to the revenue loss due to diverted clicks:

$$k_2 a + (n-k_2)(1-m)a$$

For $m < \frac{C_{ss}}{(n-k_2)a}$ the revenue loss is smaller than the cost of securing a website and AS lets ISP divert a fraction of clicks in every stage game from k_2 till n. □

3.6 Refinement of the Game-theoretic Model

In order to understand the implications of this game-theoretic model in practice, we apply the analysis of Section 3.5.3 to the real data set. Thus, we must first refine the game-theoretic model by estimating the values of the parameters using the data that characterize an online ad system in practice. We consider three different modes of operation: (i) *Nominal* (Figure 3.2), (ii) *Non-cooperative* (Figure 3.3) and (iii) *Cooperative* (Figure 3.4), that capture possible interactions between entities of the system. The symbols used below are given in Table 3.3.

Table 3.3: Table of symbols for the numerical analysis.

Symbol	Definition
$\mathcal{K}$	Set of Advertisers
$\mathcal{K}_1$	Set of Advertisers associated only with the Ad Server
$\mathcal{K}_2$	Set of Advertisers associated both with the Ad Server and the ISP
h	Fraction of revenue paid by the Ad Server to websites
l	Fraction of revenue paid by the Ad Server to ISP when cooperating
s	Fraction of revenue paid by the ISP to a third party for targeted advertising
β_j	Fraction of clicks that become conversions
Q	Volume of clicks
$v_{k,j}$	Advertiser k valuation of j's clicks

3.6.1 Nominal Mode

The system operating in the nominal mode is depicted in Figure 3.2. It corresponds to the case when ISP is faithfully forwarding the traffic and does not try to take part in the online advertising system. A number of clicks, Q, is generated by users at the website WS. The clicks are registered by AS that distributes them among associated advertisers (AVs). We assume that AS distributes clicks uniformly at random among the AVs. In practice, the volume of clicks given to each advertiser is typically determined in an auction based on advertisers' bids on given keywords. Modeling the auction process would add complexity to the problem and is out of the scope of our work, therefore we assume that all advertisers receive the same amount of clicks. We also assume that there is no click fraud, i.e., all the clicks from one ad network have the same conversion probability. Let the conversion probability of a click from AS be β_1.

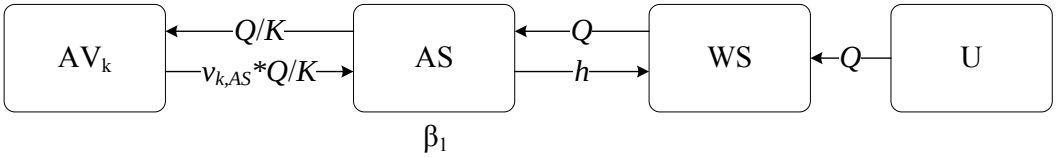


Figure 3.2: Nominal Mode. ISP faithfully forwards the traffic and AS distributes the clicks to its set of advertisers AV_k.

Advertiser $AV_k \in \mathcal{K}$, where $\mathcal{K}$ is the set of all AVs associated to AS and $K = |\mathcal{K}|$, selects its valuations $v_{k,AS}$ on clicks such that its revenue from AS is maximized. The valuations are directly proportional to the conversion probability of the clicks (i.e., the quality of the clicks) received from AS [95].

For the clicks that turn into conversions, AVs pay AS, and AS pays a fraction h of that amount to WS where the clicks were generated. We assume that AVs pay AS an amount of money equal to their valuations of clicks (i.e., bids). Therefore, $AS's$ nominal payoff, a, is:

$$u_{AS} = \frac{Q}{K}(1-h)\sum_{k\in\mathcal{K}} v_{k,AS} = a \tag{3.8}$$

3.6.2 Non-cooperative Mode

If ISP chooses to become part of the online advertising system and to divert clicks from AS, the system can be modeled as in Figure 3.3. ISP diverts a fraction m of Q clicks generated at WS and distributes it uniformly at random among the set of its own associated advertisers.

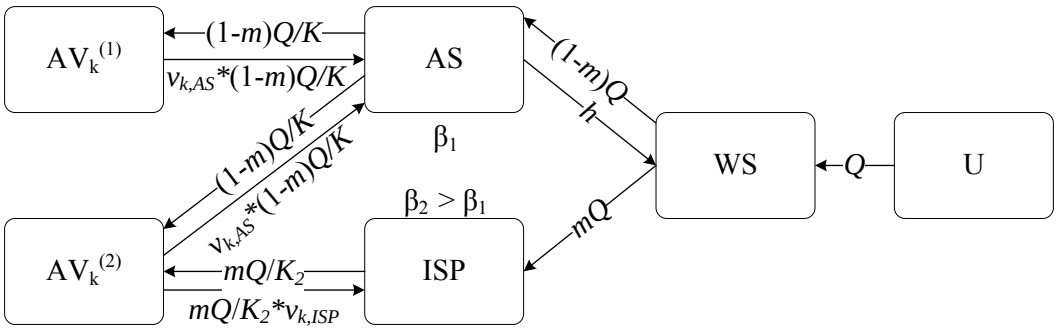


Figure 3.3: Non-cooperative Mode. ISP diverts fraction m of clicks from the AS and distributes them to its set of advertisers $AV_k^{(2)}$. AS distributes the remaining fraction of clicks to its own set of advertisers $AV_k^{(1)}$.

In the non-cooperative model, we assume two types of AVs:

Advertisers of type 1, $AV^{(1)}$, are associated only with AS because they care about their reputation and they do not associate with ISP even if it would increase their revenues. The set of $AV^{(1)}$ is represented by $\mathcal{K}_1$, where $K_1 = |\mathcal{K}_1|$.

Advertisers of type 2, $AV^{(2)}$, are associated with both ISP and AS. $AV^{(2)}$ are willing to associate with ISP, because working with both AS and ISP generates more revenue. The set of $AV^{(2)}$ is represented by $\mathcal{K}_2$, where $K_2 = |\mathcal{K}_2|$.

There are no advertisers associated only with ISP, because advertisers that do not care about their reputation have higher revenue when associated with both ISP and AS than in the case when they are associated only with ISP. Therefore, we have $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2$ and $K_1 + K_2 = K$.

An advertiser $AV_k^{(2)}$, associated with both AS and ISP, selects its valuations $v_{k,AS}$ and $v_{k,ISP}$ on clicks such that its revenues from AS and ISP are maximized.

The conversion probability of clicks coming from ISP (β_2) is higher than the conversion probability of clicks coming from AS (β_1), i.e., $\beta_2 > \beta_1$, due to ISP's better ad targeting based on users' private information. Therefore, an advertiser places higher valuations on clicks from ISP than on clicks from AS, i.e., $v_{k,ISP} > v_{k,AS}$. The difference in valuations on clicks from two different ad networks can be expressed as [95]:

$$v_{k,ISP} = \frac{\beta_2}{\beta_1} v_{k,AS} \tag{3.9}$$

Given that ISPs do not necessarily have the resources to perform ad targeting themselves, we assume that they rely on a third-party entity, as observed in practice [76]. The partnering entity provides ad targeting technology and in return, ISP gives the partner a fraction s of its revenue. The payoffs of AS and ISP in the non-cooperative model are:

$$u_{AS} = \frac{(1-m)Q}{K}(1-h)\sum_{k\in\mathcal{K}} v_{k,AS} = (1-m)a \tag{3.10}$$

$$u_{ISP} = \frac{mQ}{K_2}(1-s)(\sum_{k\in\mathcal{K}_2} v_{k,ISP}) - \varepsilon = mb - \varepsilon \tag{3.11}$$

where

$$b = \frac{Q}{K_2}(1-s)\sum_{k\in\mathcal{K}_2} v_{k,ISP} \tag{3.12}$$

3.6.3 Cooperative Mode

When cooperating with AS (Figure 3.4), ISP provides users' private information P that AS uses to improve ad targeting, i.e., to improve the conversion probability of a click. The benefit for AVs is that they receive clicks that have higher probability of conversion (β_2) which is why they offer higher valuations ($v_{k,ISP}$) for those clicks. Thus AS earns more money for Q clicks when cooperating than when operating alone. In return for users' private information, AS gives a fraction l of the revenue to ISP.

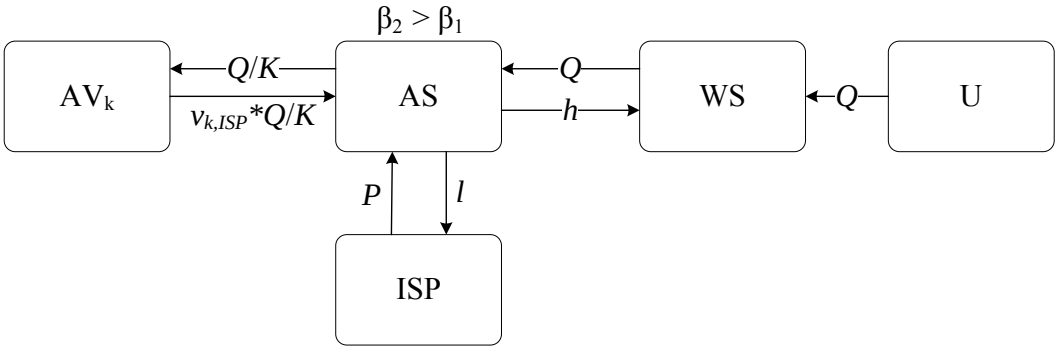


Figure 3.4: Cooperative Mode. ISP provides users' profiles P to AS and obtains a fraction l of the ad revenue in return. Based on P, AS improves the quality of clicks and thus obtains higher ad revenue.

In the cooperative model, based on (3.8) and (3.9), the payoffs of AS and ISP are:

$$u_{AS} = \frac{Q}{K}(1-h-l)\sum_{k\in\mathcal{K}} v_{k,ISP} = \frac{\beta_2}{\beta_1}\frac{1-h-l}{1-h}a = c_2 \tag{3.13}$$

$$u_{ISP} = \frac{Q}{K}l\sum_{k\in\mathcal{K}} v_{k,ISP} = \frac{\beta_2}{\beta_1}\frac{l}{1-h}a = c_1 \tag{3.14}$$

Cooperation is good for AS when $l \leq (1-h)(1-\frac{\beta_1}{\beta_2})$, i.e., when the cooperation revenue (c_2) is greater than the nominal revenue (a), based on (3.13).

3.7 Numerical Analysis

In this section, we evaluate the impact of the results in Section 3.5.3 on the Web using the above equations and a real data set. We extend the analysis to multiple websites. Note that the outcome of the game can be different for different websites, e.g., AS can decide to secure only some of the websites while cooperating with ISP for the others. We are interested in the outcomes of the game for the most popular 1000 websites.

3.7.1 Evaluations on a Real Data Set

The exact values of parameters that characterize the system in practice are difficult for us to obtain. Many of them are kept confidential (e.g., Q and h) and some are difficult to quantify (e.g., the value of users' private information). However, this information is available to the players of the game, i.e., ad networks and ISPs, thus our model is applicable in practice.

We use the following estimated values of system parameters in our analysis: (i) AS pays $h = 10\%$ of the revenue to its referrers per click conversion [54]; (ii) ISP gives $s = 30\%$ of the revenue to a third-party ad targeting company (varying the values of s has no significant effect on the results); (iii) the cost of a certificate is \$399 [83]; (iv) the cost of mounting an attack is $\varepsilon \leq \$100$ (writing and deploying scripts to perform inflight modifications of the ad traffic have a negligible cost, especially compared to the ad revenue and hence, the value of ε has no effect on the results in practice) and (v) advertisers pay \$0.5 per click conversion [26].

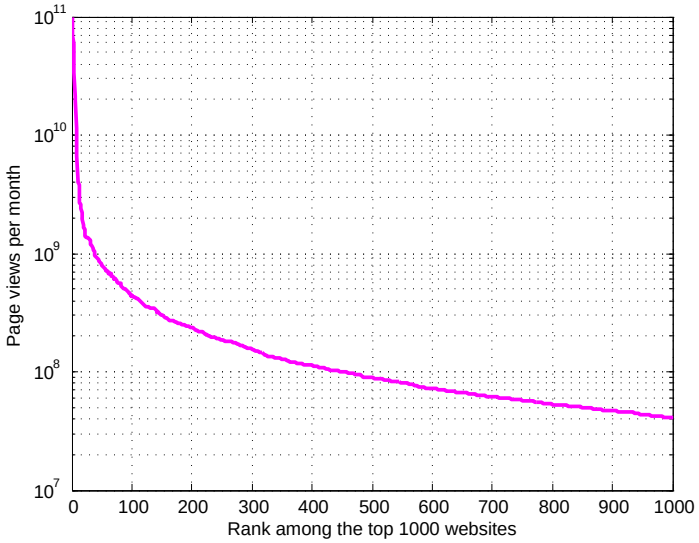


Figure 3.5: Popularity of the top 1000 websites based on page views per month.

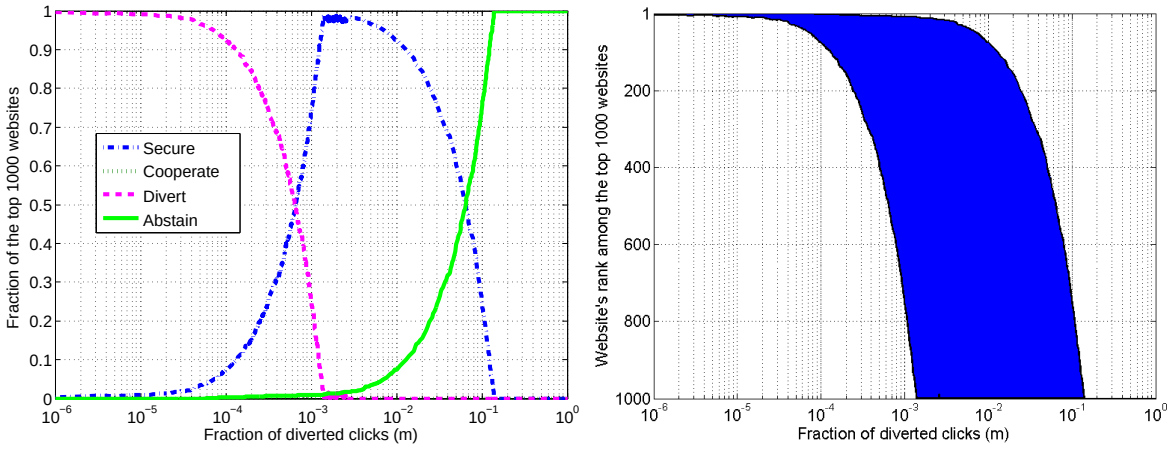
We infer the generated volume of clicks on ads on the 1000 most popular websites on the Web, based on the data of page views on each website in June 2009 (Figure 3.5), obtained from *Compete.com*. Based on the measurements reported in [161], 58% of the top websites host ads and there are 8 ads per page on average. The probability that a click occurs on an ad is around 0.1% [18]. Consequently, to convert the number of page views into the number of clicks on ads on each website, we use the following formula:

$$Q_i = (\text{Page views on the website } i) \cdot 0.58 \cdot 8 \cdot 0.001. \tag{3.15}$$

There are two system parameters that influence the outcomes of the game: (i) the fraction of shared revenue when cooperating (l) and (ii) the improvement of ad targeting ($\frac{\beta_2}{\beta_1}$). Thus, we take into account different values of the two parameters and analyze their effects. The fraction m of clicks diverted by non-cooperative ISP is also kept as a parameter of the analysis. We vary this parameter, and then consider the equilibrium outcome for each of the 1000 most popular websites, as predicted by the analysis in Section 3.5.3. Our numerical results show that the outcomes are mostly determined by the values of the three parameters: l, $\frac{\beta_2}{\beta_1}$ and m. By varying values of other system parameters we conclude that they only insignificantly change the absolute values of the results but not the main observations.

3.7.2 Numerical Results

In the case of a non-cooperative ISP, the outcomes of the multi-stage game for the 1000 most popular websites are depicted in Figure 3.6a. To obtain the non-cooperative scenario, we consider that the fraction of shared revenue when cooperating is high ($l = 0.4$) and ad targeting is not significantly improved ($\frac{\beta_2}{\beta_1} = 1.75$). The AS is not willing to cooperate and pay such a high price for not so valuable user profiles, thus we observe the non-cooperative behavior. Outcomes are represented with the four curves in Figure 3.6a. Each curve represents a fraction of websites for which the outcome of the game is the same.[5]



(a) (b)

Figure 3.6: Outcomes of the game in the non-cooperative scenario applied to real data. (a) Outcomes for the top 1000 websites (*Cooperate* curve is equal to zero). (b) Ranks of the websites that should be secured.

All the values of the *Cooperate* curve are equal to zero, which shows that, in this scenario, cooperation will not be established in any stage of the multi-stage game, for any of the websites.[6] The *Divert* curve represents the fraction of websites from which ISP successfully diverts a fraction m of clicks during all stages of the multi-stage game.[7] The *Secure* curve represents the fraction of websites that AS will secure at some stage of the multi-stage game,

[5]For a given m, the sum of the values of the four curves is always equal to one.

[6]The SPNE that correspond to Result 1, Result 4 and Result 5.2 are not achieved in the non-cooperative scenario.

[7]The SPNE that correspond to Result 3.1 and Result 5.1.

due to ISP diverting clicks.[8] The fraction of websites for which ISP will abstain during all stages of the multi-stage game is represented with the *Abstain* curve.[9]

Results show that ISP can divert a small fraction ($m < 0.001\%$) of clicks from all of the 1000 websites (*Divert* curve equal to one) without causing AS to react (*Secure* curve equal to zero). This amount of click diversion could be done in practice either by a very small ISP modifying all the traffic of its subscribers or by a large ISP selectively modifying only a tiny portion of the traffic it forwards.

If ISP starts diverting a higher fraction of clicks, it causes AS to deploy security and protect the concerned websites. Thus, we observe that the fraction of websites that will be secured among the top 1000 websites (*Secure* curve) is increasing for higher values of m. Consequently, the fraction of websites from which ISP successfully diverts clicks (*Divert* curve) is decreasing. When ISP diverts $m = 0.14\%$ of clicks, almost all (98.7%) of the 1000 websites should be secured.

If ISP is to divert a higher fraction ($m > 0.14\%$) of clicks, it would try do so only for the websites for which the condition $ma < C_{ss}$ holds, i.e., for which the revenue that ISP would divert from AS is smaller than AS's cost of deploying the security mechanism. Otherwise, AS would secure the websites in the first stage of the game, which would cause ISP to only pay the cost of mounting the attack without any gain. Thus, if the condition $ma < C_{ss}$ does not hold for a given website, ISP abstains during all stages of the multi-stage game. The fraction of such websites for which ISP abstains during all stages of the multi-stage game is higher for higher values of m, as represented with the increase of the *Abstain* curve following the increase of m. This implies that the fraction of websites from which ISP will try to divert clicks becomes smaller, thus resulting in fewer websites that need to be secured by AS (corresponding decreasing values of the *Secure* curve). Further, results show that ISP will not try to divert a high fraction ($m > 14\%$) of clicks from any of the websites, but rather choose to abstain (*Abstain* curve equal to one).

The *Secure* curve in Figure 3.6a only shows the fraction of websites that will be secured, but we are also interested in which websites are those. The colored area in Figure 3.6b corresponds to the popularity ranks of the websites that should be secured for a given value of m. Intuitively, since the ISP diverts the same fraction m of clicks from all the websites and more popular websites generate higher ad revenue, the loss of ad revenue for AS is higher for more popular websites. As the cost of securing a website is the same for all the websites, it is better for AS to first secure the most popular websites (i.e., those that generate highest ad revenue) among the ones ISP tries to divert clicks from. In this way, AS protects more ad revenue at the same cost.

Based on the results in Figure 3.6a, for small values of m ($m \leq 0.14\%$) ISP tries to divert clicks from all of the websites and the fraction of websites to be secured increases with the increase of m. The colored area in Figure 3.6b shows that for $m \leq 0.14\%$ AS secures the fraction of websites starting from the most popular ones, i.e., the highest ranked websites according to their popularity.

However, as m increases ($m > 0.14\%$) ISP stops diverting clicks from the most popular websites. We concluded earlier that ISP will not try to divert a given fraction m of clicks from websites for which the condition $ma < C_{ss}$ does not hold, as it would obtain a negative payoff. For a given m, this becomes true first for the most popular websites that generate

[8] The SPNE that corresponds to Result 3.2.
[9] The SPNE that corresponds to Result 2.

high ad revenue a. Therefore, ISP would only try to divert clicks from the less popular websites. Consequently, the threat exists only for the less popular websites and the most popular among those are the ones that will be secured by AS. For example, for $m = 5\%$, 60% of the websites will be secured by AS (*Secure* curve equal to 0.6 in Figure 3.6a) that correspond to websites ranked from 400 to 1000 (Figure 3.6b). For the highest ranked 40% there is no need to implement security as ISP would abstain from diverting clicks from those, knowing that AS would immediately implement security because $ma > C_{ss}$.

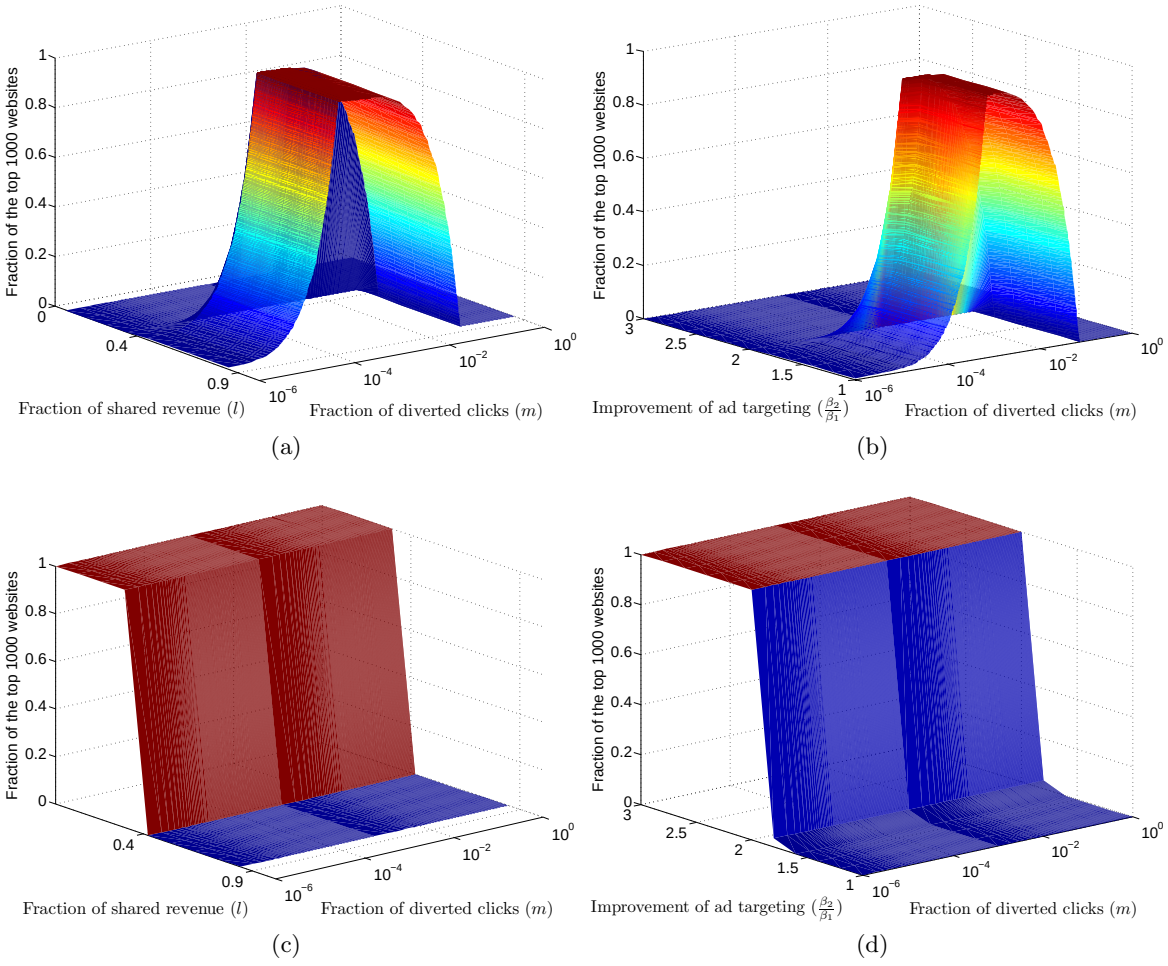


Figure 3.7: Effects of the parameters l and $\frac{\beta_2}{\beta_1}$ on the game outcomes. Fraction of the top 1000 websites that should be secured depending on: (a) l and (b) $\frac{\beta_2}{\beta_1}$. Fraction of top the 1000 websites for which cooperation is achieved depending on: (c) l and (d) $\frac{\beta_2}{\beta_1}$.

Next, we analyze the effect of the parameters l and $\frac{\beta_2}{\beta_1}$ on the results. Figures 3.7a and 3.7b represent the *Secure* curve for different values of parameters l and $\frac{\beta_2}{\beta_1}$, respectively. The graphs show that non-cooperative behavior occurs when ISP demands a high share $(0.4 \leq l \leq (1-h))$ for the users' profiles and when ad targeting cannot be significantly improved ($\frac{\beta_2}{\beta_1} < 2$). Observe that the fraction of the websites to be secured follows the same pattern as in Figure 3.6a. Thus, following our analysis of the non-cooperative behavior, the

threat of ISP diverting ad revenue can lead to improved Web security. In our example, if ISP modifies around 0.14% of the clicks, almost all of the websites should be secured.

The graphs in Figures 3.7c and 3.7d represent the fraction of the 1000 most popular websites for which ISP and AS cooperate during all stages of the multi-stage game.[10] The results show that if AS does not have to give a high share of its revenue to ISP ($0 < l < (1-h)(1-\frac{\beta_1}{\beta_2})$) or if the users' private information can significantly improve ad targeting ($\frac{\beta_2}{\beta_1} \geq 2$), ISP and AS cooperate for all of the websites.

We do not show the equilibrium outcomes *Abstain* and *Divert*, as they also follow the patterns in Figure 3.6a.

3.8 Summary

In this chapter, we have investigated the problem of ISPs becoming strategic participants in the online advertising business either by cooperating with ad networks to improve ad targeting (i.e., sharing users' private information in exchange for a share of the ad revenue) or by non-cooperatively diverting a fraction of the ad revenue from ad networks (i.e., implementing inflight modification of ad traffic). We have proposed a game-theoretic model of this problem to study the behavior and interactions of the ISPs and ad networks. We have applied our model to the real data of the 1000 most popular websites to understand the meaning of the results in practice. Our analysis shows that whether an ISP will be *non-cooperative* or *cooperative* mostly depends on the value of the users' private information obtained by ISPs and on their share of the advertising revenue. The effect on the Web is positive in both cases: When ISPs are cooperative, users receive better targeted ads and both ISPs and ad networks earn higher revenues; when ISPs are non-cooperative, Web security can be improved as a side effect of protecting the ad revenue.

Publication: [221]

[10] The SPNE that correspond to Result 1 and Result 4.

Chapter 4

ISPs and Ad Networks Against Botnet Ad Fraud

It is widely accepted that botnets are one of the most serious threats on the Internet since they are predominantly used for nefarious activities. Thwarting botnets requires huge resources. ISPs are in the best position to fight botnets and there are a number of recently proposed initiatives that focus on how ISPs should detect and remediate bots. However, it is very expensive for ISPs to do it alone and they would certainly welcome some external funding. Among others, botnets severely affect ad networks, as botnets are increasingly used for ad fraud. Thus, ad networks have an economic incentive, but they are not in the best position to fight botnet ad fraud. Consequently, ad networks might be willing to subsidize the ISPs to do so. In this chapter, we provide a game-theoretic model to study the strategic behavior of ISPs and ad networks and we identify the conditions under which ad networks are likely to solve the problem of botnet ad fraud by themselves and those under which they will subsidize the ISPs to achieve this goal. Our analytical and numerical results show that the optimal strategy is determined by the ad networks' ad revenue loss due to ad fraud and the number of bots participating in ad fraud.

Chapter Outline In Section 4.1, we discuss economic incentives of ad networks to thwart botnet ad fraud and how they can do so by subsidizing ISPs. After a brief presentation of the state-of-the-art research on the economics of botnets, click fraud and investments in online security in Section 4.2, we describe the impact of botnets on the online advertising business in Section 4.3. We then address the various threats and countermeasures in Section 4.4 and provide a case study of a botnet ad fraud in Section 4.5. In Section 4.6, we present a game-theoretic model with two players, the ISP and the ad network, and identify equilibrium outcomes of that game. We provide a numerical example to study the practical impact of the obtained results in Section 4.7 and present concluding remarks in Section 4.8.

4.1 Introduction

Today, botnets are a very popular tool for perpetrating distributed attacks on the Internet. Botnets are a serious threat for a number of entities: end users, enterprises with online businesses, websites, Internet Service Providers (ISPs), advertisers and ad networks (ANs). Botnets usually consist of compromised end users' PCs. Thus, depending on the malware, the consequences for end users can be severe (e.g., stolen credentials). Very often botnets are used for sending spam, which creates problems for ISPs, enterprises and end users. Botnet operators (aka bot masters) also use botnets to extort money from websites' owners under the threat of Distributed Denial of Service Attacks (DDoS). Lately, it is becoming more and more popular to use botnets for ad fraud [22], which creates a loss of ad revenue for advertisers, associated websites and ad networks and security threats for end users (e.g., fraudulent ads that lead to phishing attacks).

Consequently, thwarting botnets would benefit everyone and would reduce the level of online crime on the Internet. However, the problem of botnets in general cannot be solved exclusively by users (lack of know-how), ISPs (too expensive to fight botnets alone), ad networks, advertisers, websites and enterprises (lack of tools and resources).

Recent initiatives propose that ISPs perform the detection of botnets and remediation of the infected devices [151, 172]. Indeed, it is the ISPs that are in the best position to detect the presence of a botnet and to take measures against it. Yet, the revenues of ISPs are not (directly) affected by the botnets and ISPs would certainly welcome some external funding in the efforts to fight botnets. One possible approach is a government-sponsored program, as in Australia [28] and Germany [57]. In the case governments are unwilling to fund these initiatives, ISPs need to find a way to make them, at the very least, cost neutral if not cost positive.

Over the last decade, online advertising has become a major component of the Web, leading to annual revenues expressed in tens of billions of US Dollars (e.g., $31.7 billion in the US in 2011 [148]). The business model of a fast growing number of online services is based on online advertising and much of the Internet activity depends on that source of revenue. Unsurprisingly, the ad revenue has caught the eye of many ill-intentioned people who have started abusing the advertising system in various ways. In particular, click fraud has become a phenomenon of alarming proportions [22]. Recently, a new type of ad fraud attack has appeared, consisting in the inflight modification of the ads themselves. A prominent example is the *Bahama botnet*, in which malware causes infected systems to display altered ads, as well as altered Google or Yahoo search results to the end users [20]. Other examples of such botnets are Gumblar [27], Xpaj [159] and Ghost Click [74]. If the modification of ads is successful, users see ads that are different from what they would otherwise be. Consequently, users' clicks on the altered ads generate a revenue for the bot master instead of the ad network. Thus, the modification of the ads negatively affects the revenues of the "legitimate" advertisers and undermines the business model of the ad networks.

Considering the increasing trend of botnet ad-fraud attacks and the consequently increasing loss of ad revenue for ad networks, ad networks have economic incentives to fight botnets. However, ad networks are not in the best position to thwart botnets themselves and thus they might be willing to subsidize the ISPs to achieve that goal. In this chapter, we investigate whether ad fraud botnets alone are a reason enough for ISPs and ad networks to cooperate. Such cooperation would help ISPs deploy detection and remediation mechanisms and would be a first step towards fighting all botnets.

The contributions of this chapter are threefold. First, we identify two potential countermeasures that ad networks could use to address the problem of botnet ad fraud and we propose a cooperation scheme in which ISPs and ad networks jointly fight botnets. Second, we provide a game-theoretic model to study the interactions between ISPs and ad networks, as well as to identify optimal countermeasure strategies of ad networks and ISPs under different conditions. Finally, we apply the results to a real data set to study the practical impact. To the best of our knowledge, this work is the first to model the behavior of ISPs and ad networks facing botnet ad fraud.

4.2 Related Work

There are three main categories of literature that are relevant to our work: research on economics of botnets, online advertising fraud and security investments on the Internet.

A white paper from Kaspersky Lab [178] provides an in-depth analysis of the economics of botnets and estimates the revenue that can be generated from botnets perpetrating DDoS attacks, theft of confidential information, phishing attacks, click fraud, sending spam, distributing malware and from leasing a botnet. A botnet master can earn a significant amount of revenue from each of these activities (e.g., estimated $20 million from botnet-originated DDoS attacks in 2008). Moreover, all these activities can be performed at the same time.

Designing botnet-disabling mechanisms from an economic perspective is proposed in [122] for the case of botnet ad fraud and in [171] for the case of botnet DDoS attacks. In [122], authors propose a business model attack-generator called Multihost Adware Revenue Killer (MARK) that operates by constructing a distributed network of machines capable of controlling ad impression numbers, clickthrough rates and software package installs. They demonstrate that it is possible to change the economics of online advertising, in particular, to reduce the ad revenue generated by adware and botnets. However, the use of MARK in practice is highly questionable because it can be targeted at legitimate online marketing models. An economic model of botnet-related cybercrime is proposed in [171] to understand the effective rental size and the optimal botnet size that can maximize the profits of botnet masters and attackers (who rent botnets). The model considers uncertainty in the level of botnet attacks which is introduced by virtual bots (honeypots running on virtual machines that are to be compromised by the botnet masters). Introducing virtual bots in botnets reduces the probability of launching a successful attack and thus reduces the profitability of the botnet market. Consequently, the model predicts that botnet-related crimes will decrease with failing profit margins. In our work, we also propose a botnet-disabling solution from an economic perspective, but we introduce a new strategic player (the ISP) and we focus on the collaborative efforts of ad networks and ISPs.

Related work on online advertising fraud and security investments on the Internet are surveyed and discussed in Chapter 3. In Chapter 3, we investigate the problem of ISPs becoming strategic participants in the online advertising business. We propose a game-theoretic model of this problem to study the behavior and interactions of the ISPs and ad networks and we show that, when facing ad fraud, ad networks are willing to collaborate with ISPs in order to protect their ad revenue. In this chapter, we consider a distributed threat (in contrast to the centralized model in Chapter 3) and we propose a new collaborative approach that takes into consideration the economic incentives of the ad networks and ISPs.

4.3 System Model

We consider a system consisting of an *online advertising system*, a number of *bots* that attempt to exploit the online advertising system and an *ISP*, as depicted in Figure 4.1.

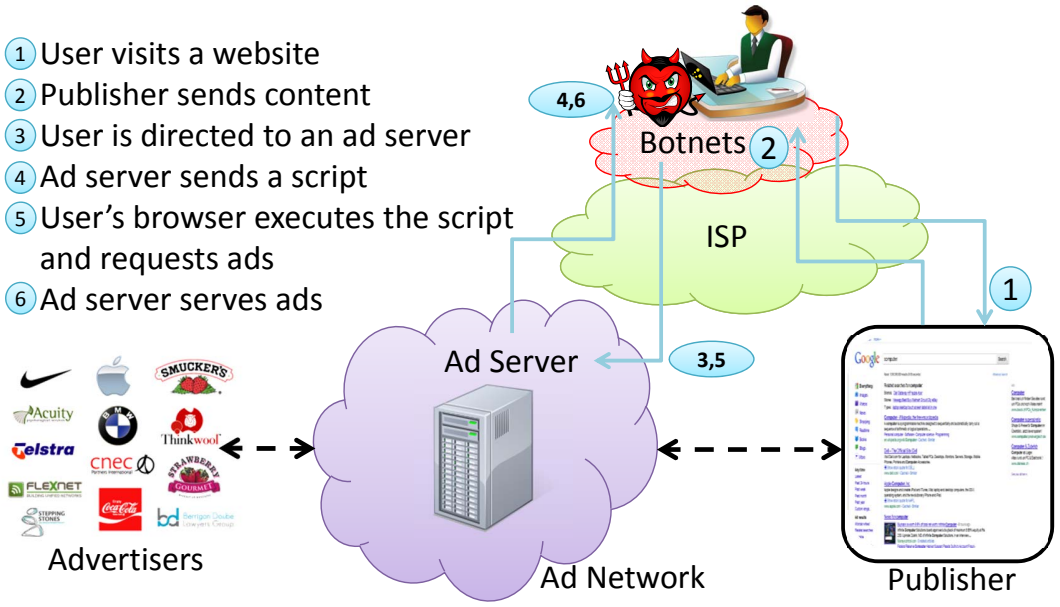


Figure 4.1: System model: Online ad system, an ISP and bots exploiting the ad system.

4.3.1 Online Advertising Systems

We consider the most prevalent model of serving online ads to end users, depicted in Figure 4.1. The process of delivering ads is presented in detail in Chapter 1 and summarized in Figure 4.1. We consider the pay-per-click ad revenue model in which advertisers pay a *cost-per-click* (CPC) to the ad network for each user-generated click that directs the user's browser to the advertised website. The ad network gives a fraction of the ad generated revenue to the website that hosted the ad. Popular websites that attract more visitors create more traffic towards advertised websites, thus generating more revenue for themselves and for the associated ad networks. Since we consider a single ad network in our system model, we assume that all the websites that host online ads are associated to that ad network. Throughout the rest of the chapter, we use the terms "user" and "user's browser" interchangeably.

4.3.2 Botnets

A botnet is a collection of software robots, or *bots*, that run autonomously and automatically. Bots are typically compromised computers running software, usually installed via drive-by downloads exploiting Web browser vulnerabilities, worms, Trojan horses or backdoors, under a common command-and-control infrastructure. Recently, a botnet of compromised wireless routers has been detected [24]. Such a botnet has the advantage of having the bots almost always connected to the Internet (compared to the typical end-user machine that is connected to the Internet only from time to time). In addition, it is more difficult to detect that a device

has been compromised, due to the lack of security software for such devices (e.g., no anti-virus software) or by a user.

A bot master controls the botnet remotely, usually through a covert channel (e.g., Internet Relay Chat) and usually for nefarious purposes. According to Click Forensics, a company that produces tools to detect and filter fraudulent clicks, for the third quarter of 2009, 42.6% of fraudulent clicks came from bots (Figure 4.2) [22]. For the same period in 2008, botnets accounted for 27.5% of fraudulent clicks. The data shows that using botnets for ad fraud is becoming more and more popular. This creates a problem for advertisers, ad networks and websites as they lose a part of the ad revenue. In the system model, we consider a number of compromised devices that run a malware that causes infected machines to participate in an advertising fraud.

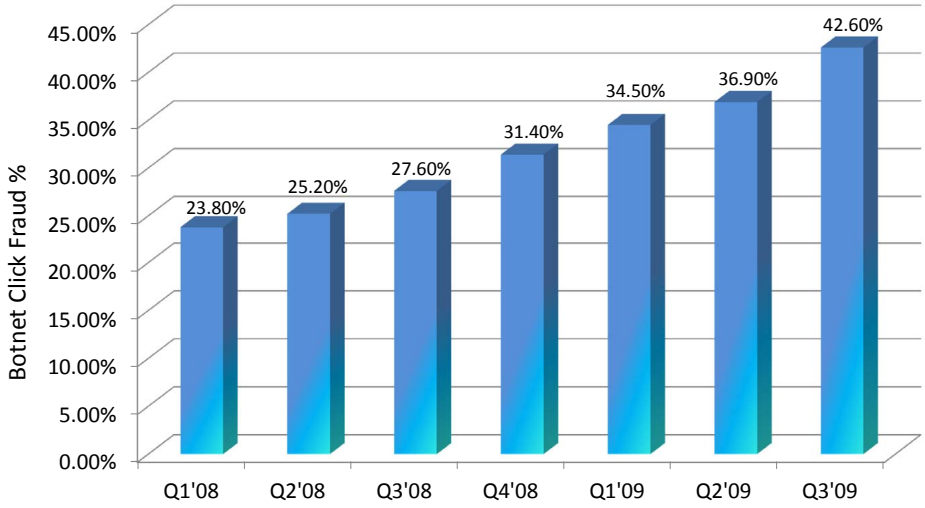


Figure 4.2: Significance of botnet ad fraud: Botnet Click Fraud by Quarter.

4.3.3 Internet Service Providers

The traditional role of an ISP is to provide Internet access to end users and to forward users' traffic in compliance with the Network Neutrality policy [108]. However, recently, ISPs have begun taking on additional roles. In the EU, ISPs have to obtain and keep the records about their users' online activities and provide them upon request to law enforcement agencies [10].

A new IETF intitiative focuses on how ISPs can manage the effects of devices used by their subscribers, detect those that have been infected with malicious bots, notify the subscribers and remediate the infection via various techniques [151]. The Internet Industry Association (IIA) has also drafted a new code of conduct that suggests ISPs should detect malware-infected machines of their subscribers and actually take the action to address the problem [172]. Complying with these initiatives, ISPs would make it more difficult for botnets to operate, thus helping to reduce the level of online crime on the Web. However, the problem is that ISPs have to find funding for those initiatives.

One possible approach is a government-sponsored program, such as the Australian Internet Security Initiative, in which a third-party helps identify malware-infected devices, notifies the appropriate ISPs that then notify and help the subscribers to remedy the problem. About 90%

of Australian ISP subscribers are covered by this initiative. A similar program is launched in 2010 in Germany, where ISPs are cooperating with the German Federal Office for Information Security [57]. In the case governments are unwilling to fund the initiative, ISPs need to find a way to make it, at the very least, cost neutral if not cost positive. In our model, we consider an ISP that is willing to comply to the initiative, if doing so is at least cost neutral.

4.4 Ad Fraud: Threats and Countermeasures

Due to the immense revenues generated by online advertising, the temptation to exploit the online advertising system is high. The loss of revenue for ad networks due to ad fraud is substantial. Based on the report from Click Forensics, the overall click-fraud rate was 14.1% in the third quarter of 2009 [22], which means that 14.1% of the clicks on ads were bogus. Thus, click fraud alone creates a significant loss of revenue for ad networks, advertisers and publishers. In addition, ad networks lose ad revenue due to new types of ad fraud, such as inflight modification of ad traffic [15, 188].

One possible approach for ad networks to protect their revenue is to improve the security of online advertising systems, thus making it more difficult for an adversary to successfully exploit those systems. In Chapter 3, we use game theory to model ad network's economic incentives and show that when facing ad fraud attacks securing ad systems can maximize the revenue of a rational ad network. For example, ad fraud can be reduced if content and ads are served over HTTPS instead of HTTP. As we have discussed, due to poor implementation of certificate-based authentication in practice, if an ad network wants the secure protocol to be deployed, it should take care of the deployment and cover the costs itself. As explained previously, websites are not of the same value to the ad network, because of the different ad revenue they generate, but the cost of securing the ad revenue from a website is the same for all websites. Therefore, the ad network might decide to selectively secure only the websites that generate sufficient ad revenue that would compensate the costs.

Another possible approach for ad networks to protect their revenue is to cooperate with ISPs and eliminate the major cause of the revenue loss – botnets. They can do so by funding the existing initiatives for IPSs to detect and remove botnets, since ISPs are in a privileged position to fight botnets. As removing botnets would benefit ad networks, they have economic incentives to subsidize ISPs to fight botnets.

Thus, we envision the following two scenarios of ad networks fighting ad fraud: (i) improving the security of the online advertising systems or (ii) funding ISPs to fight botnets involved in ad fraud.

4.5 Botnet Ad Fraud: A Case Study

Consider the system as described in Section 4.3, in which N_B devices (e.g., end-users's computers or routers) have been infected by a malware and participate in ad fraud. We consider exclusively the types of ad fraud: (i) that has been the most prominent lately [20, 27], in which malware causes infected devices to return altered Search Engine Result Pages (SERPs) or altered ads in Web pages, due to DNS poisoning and (ii) in which subverted users' routers modify ad traffic on-the-fly between a Web server and a user (e.g., perform inflight attacks presented in Section 1.3). In the example of Bahama botnet, malware uses DNS poisoning by modifying `HOSTS` files on infected machines to redirect traffic to rogue Google servers which

return altered results [21]. Thus, affected users see ads and links that are different from what they would otherwise be. When users click on the altered ads, the clicks generate revenue for the bot master instead of the ad network. Thus, the bots divert a part of the ad revenue from the ad network. For simplicity of treatment, we assume that each bot diverts an equal part of the revenue and in aggregate, all the bots together divert $\lambda \in (0, 1]$ fraction of the total ad network's revenue P. Thus, the ad network's revenue in the presence of ad fraud is $P(1 - \lambda)$.

The popularity of websites, and consequently the number of user-generated clicks on ads, follow a heavy-tail distribution [85]. We infer the generated volume of clicks on ads on the 1000 most popular websites, based on the data of page views on each website in 2009, obtained from `Compete.com`. The exposure of users to online ads has been evaluated extensively in [161], showing that 58% of the top 1000 websites host advertisements and there are 8 ads per Web page on average. The probability that a click occurs on an advertisement is 0.1% [18]. Consequently, to convert the number of page views into the number of clicks on ads on each website, we use the following formula:

$$Q(n) = (\text{Page views on the website } n) \cdot 0.58 \cdot 8 \cdot 0.001.$$

Figure 4.3 shows the annual number of clicks $Q(n)$ on ads , where $n \in \{1, 2, \cdots 1000\}$ is the popularity rank of a website.

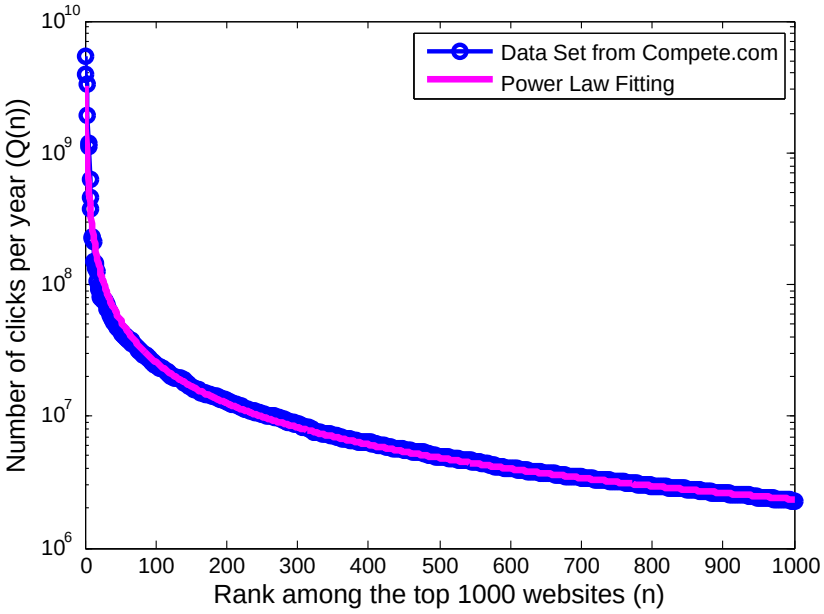


Figure 4.3: Annual number of clicks for the 1000 most popular websites and the power law fitting curve, $Q(n) = \alpha n^{-\beta} = 3.18 \cdot 10^9 n^{-1.044}$.

Applying curve fitting to the data set, we obtain that the distribution of clicks on ads across websites corresponds to the power law $Q(n) = \alpha \cdot n^{-\beta}$, where $Q(n)$ is the annual number of clicks on ads that occurred at the website with the n-th rank. The obtained parameters of the power law are $\alpha = 3.18 \cdot 10^9$ and $\beta = 1.044$ (Figure 4.3). In general, we assume that the number of clicks on ads follows the power law distribution $Q(n) = \alpha \cdot n^{-\beta}$, where $Q(n)$ is the annual number of clicks on ads that occurred at the website with the n-th rank and $\beta > 1$ [85]. Note that the values of parameters α and β are characteristics of a

given ad network and depend on the number and the type of associated websites. In order to extend our analysis and investigate what would be the effect on the entire Web (i.e., for all websites), we extrapolate the `Compete.com` data set with the obtained power law.

Given the power law distribution of the clicks, the ad revenue generated by the top x websites can be estimated by[1]

$$k \int_1^x \alpha n^{-\beta} dn = k \frac{\alpha}{(\beta - 1)} (1 - x^{1-\beta}),$$

where k is the amount of revenue that each click on ads generates for the ad network[2]. If P is the total annual revenue of the ad network, generated by all the websites (i.e., when $x \to \infty$), then per-click revenue can be calculated by $k = \frac{P(\beta-1)}{\alpha}$. According to the reports [32], the total ad revenue P in 2009 in the US is $22.4 billion.

In the following two subsections, we analyze the two proposed strategies (i.e., improving the security of the online advertising system and cooperation between the ad network and the ISP) to fight botnet ad fraud. Table 4.1 shows the used notation.

Table 4.1: Table of symbols.

Symbol	**Definition**
N_B	Number of bots
λ	Fraction of diverted ad revenue by the botnet
P	Total online advertising revenue of the ad network
k	Amount of generated revenue for each click
$Q(n)$	Number of clicks per year for the top 1000 websites
n	Popularity rank of the websites
α, β	Estimated parameters of power law distribution for $Q(n)$
c_S	Cost of securing a website
N_S	Optimal number of secured websites with S strategy
N_{SC}	Optimal number of secured websites with $S + C$ strategy
P_D	Fraction of bots detected by the ISP
c_D	Cost of the botnet detection system
c_R	Cost for the ISP per remediated infected device
R	Cost for the AN per remediated infected device
N_R	Optimal number of remediated infected devices
C	Cooperation strategy (employed by the ISP or the AN)
S	Secure websites strategy by the AN
$S + C$	Simultaneous Secure and Cooperation strategy by the AN
A	Abstain strategy (employed by the ISP or the AN)

[1] Due to the impossibility of obtaining closed-form expressions in the discrete domain, we perform computations in the continuous domain. The upper bound of the error is 8% [230].

[2] Modeling auctions and different per click revenue for ad networks is out of the scope of this work, thus we assume that all the clicks are of the same quality.

4.5.1 Securing Websites

As a countermeasure to the considered type of ad fraud (i.e., rogue servers delivering altered ads due to DNS poisoning attack on users' machines or inflight traffic modifications by compromised users' routers), the ad network can secure the communication between users and Web servers as well as between users and ad servers. For example, secure communication can be provided by the HTTPS protocol. Deploying HTTPS requires Web servers to obtain an authentication certificate from a trusted third party. In the case when websites and ad servers deploy HTTPS with valid authentication certificates, even if an adversary successfully mounts a DNS poisoning attack and redirects users' communication to rogue servers, the rogues servers cannot serve valid authentication certificates that correspond to the domain names users originally wanted to visit, thus browsers will detect security issues. HTTPS also prevents inflight modifications of the content. Consequently, users would receive unaltered links and ads and the clicks on unaltered ads would generate revenue for the intended ad network, not the adversary.

As discussed in Section 4.4 and in more detail in Chapter 2, websites usually do not implement certificate-based authentication properly. Thus, to secure the communication with HTTPS, and consequently the ad revenue, the ad network would have to take care of securing the websites and bear the costs. The cost of deploying HTTPS at ad servers can be considered negligible, given that ad networks already have valid certificates and the knowhow to properly deploy them. Moreover, there are typically only a few ad servers (compared to the number of Web servers).

Let c_S be the cost of securing a website, i.e., the cost of obtaining a certificate and deploying HTTPS at a Web server. Then the AN should pay $N_S \cdot c_S$ to secure N_S websites. N_S is the optimal number of websites that AN secures to maximize its payoff in the presence of N_B bots diverting fraction λ of the revenue. It can be calculated by the following lemma.

Lemma 1. *If the ad network fights botnet ad fraud by securing the websites, the optimal number of those secured websites is equal to* $N_S = \left(\frac{P}{c_S}\lambda(\beta-1)\right)^{\frac{1}{\beta}}$.

Proof. The total amount of revenue for the ad network (u_{AN}) when it secures x websites, due to the attack of N_B bots diverting fraction λ of the revenue, can be estimated by

$$u_{AN} = k\int_1^x \alpha n^{-\beta} dn + (1-\lambda)k\int_x^\infty \alpha n^{-\beta} dn - c_S x.$$

Recall that k is the revenue generated per each click and can be calculated as $\frac{P(\beta-1)}{\alpha}$. The first term in the revenue equation represents the revenue that the AN obtains from clicks generated on secured websites. The second term shows that the AN obtains only the remaining fraction $(1-\lambda)$ of the revenue from clicks generated on unsecured websites, as the bots divert the fraction λ of the revenue.

After simplifications we obtain: $u_{AN} = P(1-\lambda x^{1-\beta}) - c_S x$, which is a concave function of x. We obtain the optimal N_S by finding the root of the first derivation of u_{AN} with respect to x, that is $\left(\frac{P}{c_S}\lambda(\beta-1)\right)^{\frac{1}{\beta}}$. □

4.5.2 ISP and Ad Network Cooperation

In addition to the described countermeasure of securing websites, the AN can offer the ISP to cooperate to thwart botnets. The AN has an economic incentive to fund the ISP to perform detection of the botnets and remediation of the infected devices, as discussed in Section 4.3. To detect bots in the network, the ISP must deploy a detection system [151, 172]. We note the deployment cost of the detection system as c_D and we assume that such a system successfully detects a fraction P_D of the bots in the network. The proposed initiatives [151, 172] envision an online help desk where all the subscribers whose devices have been detected as bots can obtain instructions on how to remediate the problem and restore the functionality of their devices. Thus, the ISP has a cost per each remediated infected device, which we note as c_R.

For the ISP to cooperate with the AN, the AN has to provide a sufficient reward such that the detection and remediation is at least cost neutral for the ISP. Let R represent the reward the AN should pay to the ISP for the remediation of each infected device.[3]

If the AN and the ISP agree to cooperate, the outcome is that the ISP remediates N_R infected devices and the AN pays $N_R \cdot R$ to the ISP. The optimal N_R that maximizes both revenues, of the ISP and the AN, can be calculated by the following lemma.

Lemma 2. *The cooperative ISP and the cooperative AN can maximize their revenues by remediation of $N_R = P_D N_B$ infected devices.*

Proof. The total amount of revenue that the ISP obtains by cooperation and remediation of x infected devices is $x(R-c_R)-c_D$ which is a linear function of x. Therefore, the ISP maximizes its revenue by remediating all of the detected bots $P_D N_B$. Remediation of x infected devices reduces the aggregate power of the bots in the network, and together they can divert only a fraction $\lambda(1 - \frac{x}{N_B})$ of the revenue. The total amount of revenue that the AN obtains by cooperation is then $P(1 - \lambda(1 - \frac{x}{N_B})) - xR = \left(\frac{P\lambda}{N_B} - R\right) x + P(1 - \lambda)$, which is a linear function of x and is maximized at $x = N_R = P_D N_B$, i.e., for all of the detected bots. □

In summary, the ad network can use one of the above two actions to fight botnet ad fraud on the Internet. Each strategy has different benefits and costs for the ISP and the AN. In the next section, we use game theory to model this situation and consequently predict the behavior of the AN and the ISP under various conditions.

4.6 Game-theoretic Model

In this section, we model the interactions between the ISP and the AN as a static game **G** with perfect and complete information. Our model considers potential strategies of the ISP and the AN to protect against the above defined threats. We assume that the players have common knowledge about their strategies and payoffs and can observe the actions of each other. An ad network that implements click-fraud detection and mitigation has a competitive advantage on the advertising market as it has been shown that such practices have a central role in advertisers' perceived benefits of online advertising which influences their intentions to advertise online [112]. Therefore if the ad network implements such techniques it is in its

[3] Our model also applies to the case when ISPs and ANs jointly bear the costs (i.e., when it is cost negative for ISPs to thwart the botnets), as well as the case when partial funding is provided by a third-party (e.g., government) by adapting the values R or c_R.

best interest to make the information public. Similarly, if the ISP deploys countermeasures against botnets, it gives it a competitive advantage and it improves the ISP's reputation, thus it is in the ISP's best interest to make this information public. Considering the benefits and the costs of different strategies we also present the equilibria for the defined game. The key point of our game-theoretic analysis is that by using the computed equilibria it is possible to choose the optimal countermeasure protocol for different situations.

Table 4.2: Static game **G**: ISP chooses an action from $\{A, C\}$; AN from $\{A, C, S+C, S\}$. Strategy profiles (C, A) and $(S+C, A)$ are not applicable unless when ISP plays C.

		ISP	
		A	***C***
AN	***A***	$(0, P(1-\lambda))$	$(-c_D, P(1-\lambda))$
	C	N/A	$\Big(N_R(R-c_R)-c_D, P(1-\lambda(1-\frac{N_R}{N_B}))-N_R R\Big)$
	S+C	N/A	$\Big(N_R(R-c_R)-c_D, P(1-\lambda(1-\frac{N_R}{N_B})N_{SC}^{1-\beta})-N_{SC}c_S-N_R R\Big)$
	S	$\Big(0, P(1-\lambda N_S^{1-\beta})-N_S c_S\Big)$	$\Big(-c_D, P(1-\lambda N_S^{1-\beta})-N_S c_S\Big)$

4.6.1 Game Model: Strategies and Payoffs

Table 4.2 shows the normal form of the proposed static game **G**. In this game, the players play simultaneously. The ISP can choose between the following two actions: *Abstain* (A) and *Cooperate* (C). The *Abstain* action models the behavior of the ISP that is not willing to participate in the detection and rememediation of the bots . Hence the payoff of the ISP is 0, when it plays A. The cooperative ISP (that plays C) first detects the bots and then remediates the infected devices. In return, the ISP receives a reward $N_R R$ from the AN. Recall that the cost for the ISP to remediate all detected devices is $c_R N_R$. Consequently, when the ISP and the AN cooperate, the payoff of the ISP is $N_R(R-c_R)-c_D$.

In our model, the AN can choose one of the following four possible actions: *Abstain* (A), *Cooperate* (C), *Secure and Cooperate* ($S+C$), and *Secure* (S). With the *Abstain* action we model the behavior of the AN that is not willing to perform any countermeasures. In this case, the payoff of the AN will decrease to $P(1-\lambda)$. Recall that $\lambda \in [0,1]$ is the fraction of diverted ad revenue by the bots.

If the AN cooperates with the ISP, its utility will increase to $P\left(1-\lambda(1-\frac{N_R}{N_B})\right)$, where N_R is the optimal number of infected devices remediated by the ISP, which can be calculated by Lemma 2. However, the AN pays $N_R R$ to the ISP for N_R remediated devices. As a result, the total payoff of the AN when both players are cooperative is $P\left(1-\lambda(1-\frac{N_R}{N_B})\right)-N_R R$.

The AN can also secure the websites by choosing the action S, as discussed in Section 4.5.1. The AN pays $N_S c_S$ to secure N_S websites. The benefit of the AN then increases to $P(1-\lambda N_S^{1-\beta})$. Consequently, the total payoff of the AN when it plays S is $P(1-\lambda N_S^{1-\beta})-N_S c_S$, independently of whether the ISP plays C or A.

Finally, the AN can choose to simultaneously secure some of the websites and cooperate with the ISP to remediate some of the infected devices. This action is represented by $S+C$ and the total payoff of the AN in this case is $P(1-\lambda N_{SC}^{1-\beta}(1-\frac{N_R}{N_B}))-N_{SC}c_S-N_R R$, where

N_{SC} is the optimal number of secured websites when the AN plays $S+C$ and can be obtained by the following lemma.

Lemma 3. *If the AN fights botnet ad fraud with both countermeasures (action $S+C$), the optimal number of secured websites is equal to* $N_{SC} = \left(\frac{P}{c_S}\lambda(\beta-1)(1-\frac{N_R}{N_B})\right)^{\frac{1}{\beta}}$.

Proof. The proof is similar to Lemma 1. We can obtain the optimal N_{SC}, by maximizing the total payoff of the AN when it plays $S+C$. □

Lemma 3 shows that when the AN plays action $S+C$ a smaller number (N_{SC}) of websites is secured, compared to the number (N_S) of secured websites when the AN plays S (i.e., $N_{SC} = N_S(1-\frac{N_R}{N_B})^{\frac{1}{\beta}} < N_S$).

4.6.2 Game Results

In order to predict and choose the optimal action for the ISP and the AN we investigate all Nash equilibrium strategy profiles of the defined game **G**. Nash Equilibrium is a solution concept of a complete information game that finds optimal strategies that players will choose such that no player can benefit by changing only his own strategy unilaterally. In other words, we are interested in finding the strategy profiles, where neither the ISP nor the AN can increase their payoffs by unilaterally changing their strategies. We will check the existence of Nash equilibria by comparing the payoffs obtained in the game **G**.

The following theorem states conditions when the AN does not provide sufficient incentive to the ISP, such that the ISP will abstain at the Nash equilibrium.

Theorem 1. *In* **G**, *if* $R < \frac{c_D}{N_R} + c_R$, *the best response of the ISP is to play action A.*

Proof. By comparing the ISP's payoff when it plays C (i.e., whether $-c_D$ or $N_R(R-c_R)-c_D$) with that of A (i.e., 0) we obtain that the best response of the ISP is A if $N_R(R-c_R)-c_D < 0$ or $R < \frac{c_D}{N_R} + c_R$. □

This means that if the reward for remediation of the infected devices is small, the ISP will not be willing to cooperate with the AN to fight the bots.

The following theorem states when the revenue loss due to ad fraud is not significant enough to cause the AN and the ISP to perform any countermeasure against the bots.

Theorem 2. *In* **G**, *if* $R < \frac{c_D}{N_R} + c_R$ *and* $\lambda \leq \frac{N_S c_S}{P(1-N_S^{1-\beta})}$, *the action A by the ISP and the AN result in a Nash equilibrium.*

Proof. Considering Theorem 1, the ISP chooses A as its best response. The AN also plays A if its payoff when playing A (i.e., $P(1-\lambda)$) is bigger than its higher when playing S (i.e., $P(1-\lambda N_S^{1-\beta}) - N_S c_S$). Comparing these two payoffs results in the second condition of this theorem, i.e., $\lambda \leq \frac{N_S c_S}{P(1-N_S^{1-\beta})}$. □

In other words, if the reward provided by the AN does not generate sufficient incentives for the ISP to cooperate, and the amount of revenue diverted by the bots is smaller than a given threshold, both the ISP and the AN choose A to be at Nash equilibrium.
Theorem 3 shows when the AN fights the bots alone by securing some of the websites.

Theorem 3. *In* **G**, *if* $R < \frac{c_D}{N_R} + c_R$ *and* $\lambda > \frac{N_S c_S}{P(1-N_S^{1-\beta})}$, *action* A *by the ISP and action* S *by the AN result in a Nash equilibrium.*

Proof. The proof is similar to Theorem 2. □

This result shows that the amount of diverted ad revenue is significant such that a countermeasure should be deployed, but the ISP does not have enough incentive to cooperate and fight bots at this equilibrium. Consequently, the AN secures some of the websites.

Let us assume that λ is very small. Considering all the possible actions and the corresponding payoffs for the AN, the *Abstain* results in maximum payoff for the AN. In fact, action A avoids unnecessary costs for the AN, such as $N_S c_S$ or $N_R R = P_D N_B R$. These results are also in line with Theorem 2 meaning that playing A by both players results in a Nash equilibrium when λ is very small.

When λ increases (i.e., more ad revenue is diverted by the bots) the AN should deviate from A and select one of the three remaining actions as its best response. The following lemma states when the AN should begin securing N_S websites.

Conjecture 1. *In* **G**, *the AN should start securing the websites (Play* S*) when* $\lambda > \frac{N_S c_S}{P(1-N_S^{1-\beta})}$, *which corresponds to the equilibrium presented by Theorem 3.*

Proof. We compare the payoffs of the AN when it plays S or C, with the one obtained by playing the action A. The AN should then play C if $\lambda > \frac{RN_B}{P} = \lambda_1$ and should play S if $\lambda > \frac{N_S c_S}{P(1-N_S^{1-\beta})} = \lambda_2$. One can show that $\lambda_1 > \lambda_2$ when λ, and consequently N_S, is small enough. This means that the AN switches from A to S at equilibrium, when λ increases. □

Note that the AN does not switch from the A to the $S + C$, when λ increases, because the AN can protect the revenue first by playing S. In other words, the AN does not need to pay $N_R R$ to the ISP, since the cost would exceed the revenue loss. Consequently, the equilibrium of **G** corresponds to the one presented by Theorem 3. Finally, the following conjecture shows when the AN plays $S + C$ in the response to the cooperative ISP.

Conjecture 2. *In* **G**, *if the ISP is cooperative, the best response of the AN is action* $S + C$ *if* $\lambda > \frac{N_R R - N_S c_S G}{P N_S^{1-\beta} G}$, *where* $G = 1 - (1 - \frac{N_R}{N_B})^{\frac{1}{\beta}}$.

Proof. The above threshold can be obtained by comparing the payoffs of the AN when it plays $S + C$ and S. □

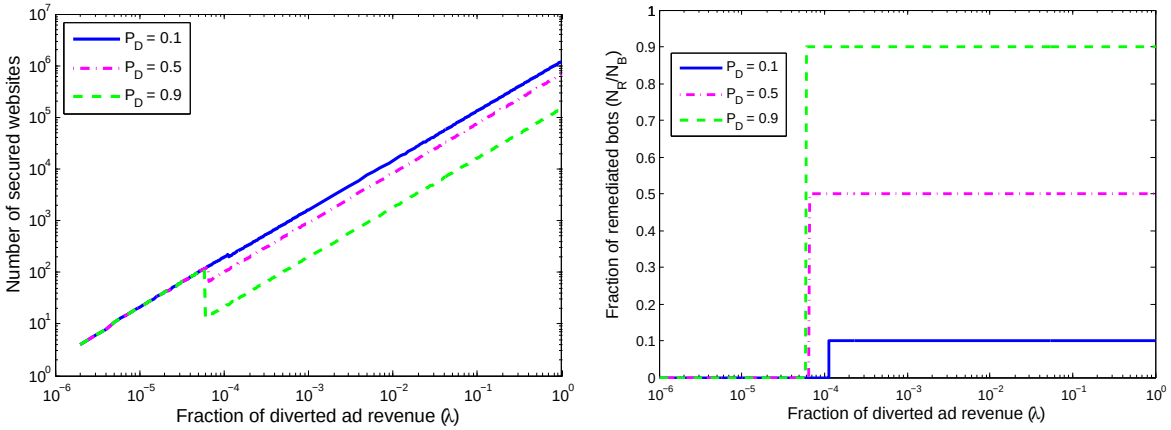
Conjecture 2 shows that if the bots divert even more revenue from the ad network, the AN will cooperate with the ISP and pay $N_R R$ to the ISP to remediate N_R bots. It will then secure a smaller number of websites compared to the case when it plays S.

4.7 Numerical Analysis

In order to understand implications of the analytical results (presented in Section 4.6) in practice, we simulate the game using real data. We compute numerically the payoffs of the static game (Table 4.2), identify the resulting equilibria and present conclusions. To investigate the effect on the entire Web (i.e., for all websites), we extrapolate the data set we have obtained from *Compete.com* with the obtained power law, as explained in Section 4.5.

We use the following assumptions for the costs in our evaluations: (i) the cost of deploying HTTPS at a Web server c_S = \$400 [83]; (ii) the cost of remediating an infected device c_R = \$100 (given that it is done via online support [151], it is the estimated cost of human labor for remediating one device per hour); (iii) the cost of the intrusion detection system c_D = \$100K [1].

We take into account different values of the fraction $\lambda \in (0, 1]$ of the ad revenue that the AN loses due to botnet ad fraud and the number of bots N_B. Given that the largest botnets detected so far [19] had several million bots each, we consider the total number of infected devices that participate in the ad fraud considered in our case study to be up to 100 million (regardless of whether they form a single or multiple botnets).



(a) (b)

Figure 4.4: Outcomes of the game applied to real data when $N_B = 10^4$: (a) Number of the most popular websites that should be secured; (b) Fraction of infected devices remediated by the ISP.

We represent the outcomes of the game for $N_B = 10^4$ in Figure 4.4. Figure 4.4a shows the number of secured websites depending on the level of threat λ. When the AN cooperates with the ISP, the fraction of remediated devices depending on the level of threat λ is shown in Figure 4.4b. We consider three scenarios (the three curves in Figure 4.4), for three different efficiencies of the detection system employed by the ISP (i.e., when the fraction of detected bots is $P_D = 0.1$, $P_D = 0.5$ and $P_D = 0.9$).

When the threat of the botnet ad fraud is very small, $\lambda < 2 \cdot 10^{-6}$, the AN does not perceive the need to perform any countermeasure against bots. Thus, there are no websites that are secured ($N_S = 0$ in Figure 4.4a)[4] and no devices are remediated ($N_R = 0$ in Figure 4.4b). This result corresponds to Theorem 2.

When the bots divert a higher fraction of ad revenue, $\lambda > 2 \cdot 10^{-6}$, the AN first secures a number of websites (Figure 4.4a). As there is no cooperation with the ISP ($N_R = 0$ in Figure 4.4b) the number of secured websites does not depend on P_D, thus it is the same in all three scenarios. The result corresponds to the finding of Theorem 3, i.e., the best choice for the AN is to play *Secure* and for the ISP to *Abstain*. The intuition behind this result is that the relatively small threat λ is distributed over N_B infected devices, thus each bot diverts a

[4] Absence of curves in Figure 4.4a signifies log(0), i.e., that zero websites are secured.

small amount of ad revenue. The cost of remediating an infected device is higher than the loss of ad revenue the bot causes, thus it does not pay off for the AN to cooperate with the ISP. However, the loss is significant enough that the AN has to deploy a countermeasure, hence it secures some of the websites. The number of secured websites corresponds to the Lemma 1.

We observe that the higher λ is, the higher is the number of websites to be secured (Figure 4.4a), until λ reaches a threshold value ($\lambda_1 = 1.12 \cdot 10^{-4}$, $\lambda_2 = 6.6 \cdot 10^{-5}$ and $\lambda_3 = 6 \cdot 10^{-5}$ for $P_D = 0.1$, $P_D = 0.5$ and $P_D = 0.9$, respectively). At the threshold values of λ the AN starts cooperating with the ISP (N_R becomes greater than zero, Figure 4.4b). Thus, the threshold value of λ represents the level of threat after which it is not sufficient to only secure the websites, but the AN will also cooperate with the ISP to fight bots (i.e., plays $S + C$). This result corresponds to Lemma 2.

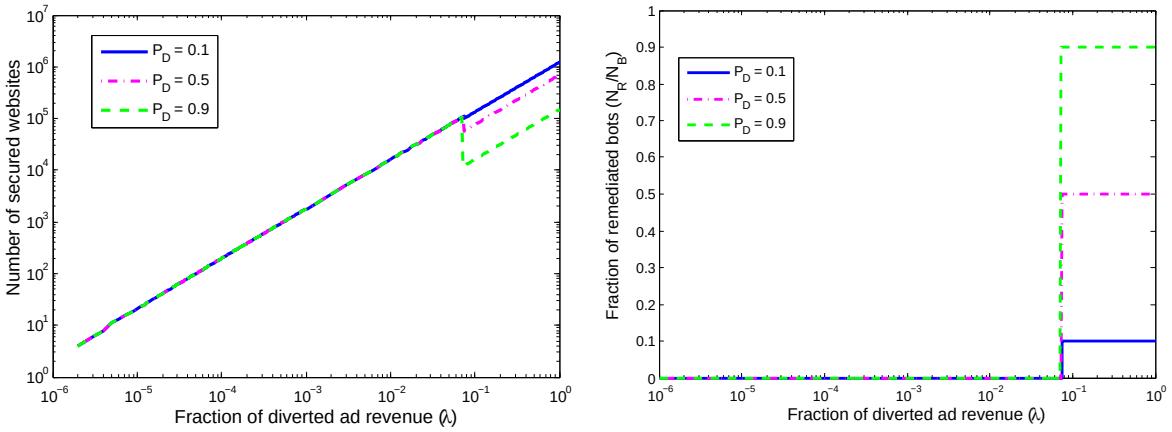
When the AN plays $S + C$, each countermeasure protects a given part of the revenue that is otherwise diverted by the bots. The total loss of revenue for the AN due to ad fraud committed by N_B bots is $P\lambda$. The remediation of N_R infected devices reduces the loss of revenue to $P\lambda(1 - \frac{N_R}{N_B})$. As the part of the revenue loss is now eliminated by the ISP, the remaining loss is smaller and consequently the AN secures a smaller number of websites. This explains the drop in the number of secured websites (Figure 4.4a), which happens at the threshold value of λ when the AN starts cooperating with the ISP. When λ increases (for values of λ greater than the thresholds), since N_R is constant for a given P_D (Figure 4.4b), in order to eliminate the increasing loss, the AN secures an increasing number of websites for the increasing λ (Figure 4.4a). In Figure 4.4b, we observe that the number of remediated devices is equal to $P_D N_B$, which confirms analytical results stated by Lemma 2. The higher the P_D is, the bigger the benefit of cooperation is, because a larger number of devices is remediated. Consequently, the AN secures a smaller number of websites for a higher P_D (Figure 4.4a).

In summary, the obtained results illustrate that: (i) For a very low level of threat λ, no countermeasures will taken against bots; (ii) When the fraction λ of the diverted revenue increases, the AN will secure a number of websites; (iii) Securing websites is not sufficient for an even higher level of threat, thus the AN will also cooperate with the ISP to remediate infected devices.

Next, we analyze the effect of the number of bots in the system (N_B) on the equilibrium outcomes of the game. Figure 4.5 represents the outcomes of the game, in the case of $N_B = 10^7$. Figure 4.5a shows the number of secured websites depending on the level of threat λ. The fraction of remediated devices depending on the level of threat λ is shown in Figure 4.5b. As before, the three curves in Figures 4.5a and 4.5b, correspond to the three scenarios ($P_D = 0.1$, $P_D = 0.5$ and $P_D = 0.9$).

We observe the same behavior as in the case of $N_B = 10^4$ bots in the system. The difference in the results for the case of $N_B = 10^7$ (Figure 4.5) compared to results for the case of $N_B = 10^4$ (Figure 4.4) is that the threshold values of λ, for which the AN begins to cooperate with the ISP, are higher. The explanation for this result is the following.

When cooperating, the ISP remediates $P_D N_B$ devices, and the AN pays $P_D N_B \cdot R$ to the ISP. Therefore, the cost of cooperation for the AN is higher when N_B is higher. Whereas, the benefit for the AN, due to remediation of $N_R = P_D N_B$ devices is $P\lambda\frac{N_R}{N_B} = P\lambda P_D$, which does not depend on N_B. For a given P_D, the cooperation benefit for the AN is higher only for the higher threat λ. Hence, when the number of bots N_B is high, the AN agrees to cooperate and pay the high cost $P_D N_B R$, only when the fraction λ of the revenue bots divert is high. Because only for the high λ the cooperation benefit $P\lambda P_D$ is high enough to justify the costs of cooperation.



(a) (b)

Figure 4.5: Outcomes of the game applied to real data when $N_B = 10^7$: (a) Number of the most popular websites to be secured; (b) Fraction of infected devices remediated by the ISP.

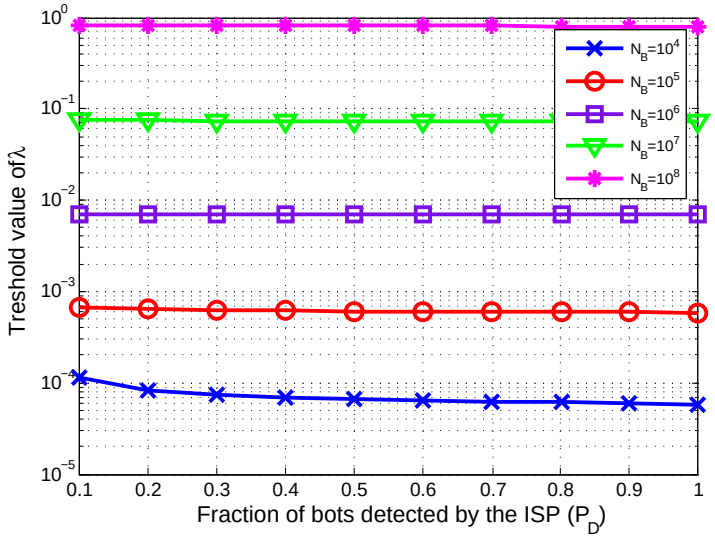


Figure 4.6: Threshold values of λ for which the AN begins cooperating with the ISP, in addition to securing the websites.

Figure 4.6 illustrates the threshold values of λ for different numbers of bots N_B in the system and for different efficiencies P_D of the detection system. For example, in the system with $P_D = 0.5$ and $N_B = 10^4$ the AN is cooperative when $\lambda > 6.6 \cdot 10^{-5}$. Whereas, if N_B is much higher, $N_B = 10^8$, the AN is cooperative only if the fraction of diverted revenue is much higher, $\lambda > 0.8$. The results confirm that for a system with a given P_D, when the number of bots is high, the AN is cooperative only when the revenue loss is very high. Based on the results in Figure 4.6, we also observe that the threshold value of λ does not vary much for different values of P_D. Hence, we can conclude that the value of N_B is the dominant factor in the decision of the AN whether to cooperate with the ISP or not. These results are also confirmed by Lemma 2.

4.8 Summary

In this chapter, we have investigated the novel situation of ISPs and ad networks behaving as strategic participants in the efforts to fight botnets. Due to the revenue loss caused by botnet ad fraud, ad networks have economic incentives to protect their revenue by either: (i) improving the security of the online advertising systems or (ii) fighting the major cause of the revenue loss – botnets. To fight botnets, ad networks might need help from ISPs, who are in a better position to deploy detection and remediation mechanisms. We have proposed a game-theoretic model to study the behavior and interactions of the ISPs and ad networks. We have applied our model to the real data to understand the meaning of the results in practice. Our analysis shows that cooperation between the AN and the ISP could emerge under certain conditions that mostly depend on: (i) the number of infected devices (ii) the aggregate power with which bots divert revenue from the ad network and (iii) the efficiency of the botnet detection system. The cooperation is a win-win situation where: (i) users benefit from the ISP's help in maintaining the security of users' devices; (ii) the AN protects its ad revenue as the botnet ad fraud is reduced; (iii) it is at least cost neutral, if not cost positive for the ISP to fight botnets. Cooperation between the AN and the ISP would help to reduce the level of online crime and improve the Web security in general.

Publication: [227]

Chapter 5

Ad-blocking Games: Monetizing Online Content Under the Threat of Ad Avoidance

Much of the Internet economy relies on online advertising for monetizing digital content: Users are expected to accept the presence of online advertisements in exchange for content being free. However, online advertisements have become a serious problem for many Internet users: while some are merely annoyed by the incessant display of distracting ads cluttering Web pages, others are highly concerned about the privacy implications – as ad providers typically track users' behavior for ad targeting purposes. Similarly, security problems related to technologies and practices employed for online advertisement have frustrated many users. Consequently, a number of software solutions have emerged that block online ads from being downloaded and displayed on users' screens as they browse the Web. In this chapter, we focus on the ad-avoidance technologies for online content and their economic ramifications for the monetization of websites. More specifically, our work addresses the interplay between users' attempts to avoid commercial messages and content providers' design of countermeasures. Our investigation is substantiated by the development of a game-theoretic model that serves as a framework usable by content providers to ponder their options to mitigate the consequences of ad-avoidance technologies. We complement our analytical approach with simulation results, addressing different assumptions about user heterogeneity. Our findings show that publishers who treat each user individually, and strategically deploy fee-financed or ad-financed monetization strategy, obtain higher revenues, compared to deploying one monetization strategy across all users. In addition, our analysis shows that understanding the distribution of users' aversion to ads and valuation of the content is essential for publishers to make a well-informed decision.

Chapter Outline We present the problem of ad avoidance and its economic ramifications for online content monetization in Section 5.1. We survey the related work in Section 5.2. In Section 5.3, we introduce the reader to background information relevant to the problem area of ad avoidance. After briefly laying out the roadmap for our analysis in Section 5.4, we delve into the details of our game-theoretic models in Section 5.5. We present simulation results in Section 5.6, before we end with concluding remarks in Section 5.7.

5.1 Introduction

It is difficult to produce a television documentary that is both incisive and probing when every twelve minutes one is interrupted by twelve dancing rabbits singing about toilet paper.
(Rod Serling, 1997)

Consumers and content providers have a love-hate relationship with advertisements. In the area of online news sites, 81% of a surveyed consumer sample report the acceptance of the presence of online advertisements in exchange for content being free. At the same time, 77% state that they would hardly ever click on these ads [187]. More significantly, across all media channels, 69% say they are "interested in products and services that would help them *skip* or *block* marketing messages [202]."

Each media genre is affected with its own specific advertisement circumvention challenges. During TV commercial breaks, viewers can leave the room to do small chores. Ads in video recordings can be manually skipped with fast-forwarding or are automatically marginalized with advanced functions of digital video recorders (e.g., TiVo) and VCRs [142]. This trend has accelerated with the availability of Home Theater PC systems such as Windows Media Center, SageTV Media Center and MythTV where available third-party add-ons allow consumers to conveniently skip ads (e.g., Comskip and ShowAnalyzer). In telemarketing, consumers are able to screen calls with CallerID or utilize software tools that act on their behalf (e.g., Telemarketing Blocker). Further, regulatory intervention can have a significant impact, for example, with the US Do-Not-Call list that upon registration allows consumers to opt out from unsolicited telephone marketing calls [220].

We focus on ad-avoidance technologies (AATs) for Web content and their economic ramifications. In the past few years, a number of effective software solutions have emerged of which the most prominent is perhaps the Adblock Plus third-party extension for the Firefox browser family [93, 201]. According to up-to-date statistics provided by Mozilla, Adblock Plus has been downloaded over 172 Million times since July 2006, and has an active daily user base of about 14 Million consumers. Further, it is also among the most popular add-ons for the Google Chrome browser with more than 100,000 weekly installs. Observers from the advertising business have predicted that the "importance of Adblock is its potential for extreme menace to the online-advertising business model [105]". However, many other technology options exist to block ads.

The emergence of behavioral ad-targeting and the associated increase in advertisers' incentives for user tracking and profiling, has led to what some observers call a "data collection arms race" (see, for example [60]). Most recently, Google's proposed changes to its privacy policy that would allow for more pervasive user data aggregation have refreshed privacy concerns in consumers' minds (see, for example [65]). And consumers object to such practices [173, 215]. However, in absence of truly effective *and* wide-spread technologies to opt-in/opt-out from tracking and the later usage of such information for ads, consumers only have the option to decide on their own personal mix of ad-avoidance technologies. For example, while consensus for a powerful and broadly applicable Do-Not-Track mechanism is still absent, some users might seek to disable scripting languages, Flash or cache cookies.[1] Others might use advanced privacy-enhancing technologies such as Tor just for the purpose of evading such commercially-motivated tracking. Finally, to be effective, avoidance of tracking does frequently necessitate

[1] It is unlikely that a meaningful compromise on Do-Not-Track will be reached quickly. See, for example, the counter-arguments on such technology brought forward by leading content providers [39].

also the blocking of the display of ads since ad campaigns almost always involve some form of campaign management tools. While this is trivially necessary to allow for ad-related payment flows, consumers cannot easily distinguish between different degrees of tracking severity.

So far, the impact of the circumvention of online tracking and ads has been moderated by the overall growth of the market for Internet commercials. The Interactive Advertising Bureau estimates that online advertising in the United States in 2011 totaled $31.7 billion and has grown by 22 percent compared to the previous year [148]. Nevertheless, many content sites suffer from the burden of ad-blocking tools, in particular, if they cater to a technology-savvy audience (see, for example, [121]). The search for an adequate response to this threat has so far proven inconclusive. In particular, monetization approaches do not only have to be economically sensible, but need to be accompanied by technically sound implementations. So far, ad-block deterrence solutions have been absent from the marketplace, even though the cost of development and deployment of simple approaches would be very manageable. In fact, as the majority of ad-blocking tools are based on filtering out elements whose URLs contain keywords like *ad* or *click*, omitting these keywords would make the existing tools ineffective. In addition, the existing tools cannot automatically detect URLs likely to be ads. Therefore, if publishers start using different keywords, ad-blocking tools would not work [201].

The stakes described in this chapter are very high and are relevant beyond the discussions about the effectiveness of marketing or commercial mechanisms. In fact, the popularity of Adblock-style add-ons represents only the tip of the iceberg, as many related challenges are consuming the attention of content producers. For example, applications such as Flipboard allow users to conveniently grab pictures and articles from many different content resources to display them in a variety of user-defined formats, and ads could be left behind (or replaced).

Our work studies in detail and in a quantitative manner the implications of a (likely to happen) growing usage of ad-blocking technologies and addresses the economic justification for effective countermeasures concerning ad avoidance. To achieve that goal, we develop a game-theoretic model that takes into account the most relevant parameters, identifies different canonical options (strategies) that the content providers and the users can choose from and forecasts the most likely outcome of such situations. The models we provide rely on Subgame Perfect Nash Equilibria (SPNE) and on Perfect Bayesian Nash Equilibria (PBNE). We complement our analytical approach with simulation results by addressing different assumptions about user heterogeneity. We make "common sense" assumptions in terms of cost and show that in general, content providers are better off when they make use of a "mixed approach", namely when they simultaneously rely on fee-funded and ad-funded monetization strategies.

5.2 Related Work

Closely related to our work is an economic model by Tåg [212]. Content providers decide whether to offer to users a subscription option that eliminates advertisements as an alternative to the content with advertisements. The content provider would introduce such an option only if the revenue gained from those customers who are willing to pay the subscription fee is greater than the revenue that the content provider would earn by only offering the basic advertisement model (i.e., mediating those customers to advertisers). According to the model, if the subscription option is introduced, it causes an increase in advertising quantity in the free version, thus increasing the annoyance due to ads and reducing the perceived quality of the free version. Moreover, consumers' aggregate utility decreases, while content providers' and

advertisers' profits increase. By increasing the amount of advertisements to non-subscribers, the content provider can further increase the differentiation between the two options. Prasad et al. [184] analyze the incentives to price discriminate when consumers are of two given types and a content provider offers two versions differing in advertising quantity and price. They show that offering two versions (price discrimination) tends to be optimal in most cases.

In another model, Shah accounts for ad-avoidance technologies [195]. Users can invest in AATs but will still see a certain fraction of the commercials. A content provider can make use of this fact by optimally differentiating the amount of ads catered to the two groups (i.e., users with and without AATs). In a two-sided market model for television advertising, Anderson and Gans similarly show that content providers could increase the number of ads to those users who do not invest in ad-avoidance technologies, as they are less averse to advertising [89]. They note that this effect is not solely due to the incentive of content providers to regain the revenue, but rather due to revealed preferences of those who do not invest in AATs. In practice, this might be one of the contributing reasons that larger number of ads per hour are observed in US television recently (the US does not impose a cap on the number of commercials, in contrast to the EU). As a result, overall welfare and program quality could decrease and programming would be tailored to appeal to a broader range of viewers.

In [234], Wilbur presents a two-sided, empirical model of television advertising and models the effects of an ad-avoidance technology on an advertisement-supported media industry. The model considers the following two possibilities. First, to overcome the loss caused by the ad-avoidance technology, networks could increase the quantity of ads, which makes AAT even more valuable to ad-adverse viewers. Therefore, this scenario leads to mutually reinforcing increases in AAT penetration and advertising time. Second, if advertisers value users with AATs less, as they fast-forward through ads, then non-AAT users become scarce and more valuable. Due to this self-selection, the remaining market is composed of viewers who accept ads which might lead to increased ad prices for advertising space. The competition for non-ad-avoiding viewers can lead to lower advertising levels, rendering ad-avoidance technologies less valuable and slowing down its rate of growth. The author uses a counterfactual experiment to gain insight into how AATs affects the industry. It is shown that when AAT penetration increases, then ad levels rise as well. Nevertheless, increased AAT levels lead to revenue loss, which implies that AATs might decrease a content provider's incentives to invest in program quality. Another model analyzes the impact of ad-avoidance behavior considering two alternative schemes by which media channels are financed: free-to-air and pay-TV [210]. The model also considers market competition in the two scenarios. The analysis shows that increased AAT levels lower profits and decrease entry in the free-to-air model. In contrast, in the pay-TV regime, lower income from ads is compensated by higher subscription fees, therefore the profits and the number of channels are unaffected.

In our model, we explicitly consider the limited information aspects related to ad-avoidance technology and its detection. As a result, content providers must invest in detection technologies to be able to distinguish between consumers that utilize AATs and those who do not engage in such activities. Such user differentiation enables content providers to deploy a personalized approach, treating each user individually and applying an appropriate monetization strategy per user. It also enables deployment of countermeasures that affect only the AAT users (e.g., preventing access to the content unless they turn off AATs or subscribe). A personalized approach is not possible in the traditional TV market, as providers do not have technological means to detect who is using AAT (e.g., fast-forwarding through ads). Therefore, the previous work has only considered aggregate strategies for a content provider,

that are applied across all the users, regardless of whether they use AATs or not. In such a scenario, instead of impacting only AAT users, the countermeasures taken to offset losses due to AATs either affect all, or even worse, only the non-AAT users. For example, an increased advertisement level only impacts non-AAT users (while AAT users can fast-forward through ads). Thus, there are no incentives for AAT users to change their behavior. On the contrary, such an approach increases incentives to adopt AATs. In contrast, in our model, the countermeasures directly affect the AAT users and therefore discourage their use of AATs. Moreover, our model leads to stronger differentiation since AAT users are not of any value to advertisers as online AATs block all available ads, whereas in the TV market, users who fast-forward through ads are still exposed to traces of marketing content.

Further academic works on advertisement circumvention have been undertaken in the context of "old media" from a legal or ethical perspective [142, 203, 218]. Additional recent work has been focused on improvements of the mechanisms for ad allocations and techniques to lower the impact of manipulation by malicious actors. See, for example, research papers on ad auctions (e.g., [116, 219]) and click fraud [163, 177].

5.3 Background

In this section, we discuss the drivers of consumer resistance to advertisements and their propensity for ad avoidance. We also review existing technologies for ad avoidance and approaches by website owners to detect ad-blocking tools.

5.3.1 Why Do Consumers Block Ads?

Previous research has studied a variety of ad-avoidance behaviors such as eliminating, ignoring or quickly flipping past commercial messages [207]. Graphical and auditory stimuli are frequently considered annoying or unconvincing, irrespective of the actual information content [207]. Online ads are more likely to be avoided if consumers hold expectations of a negative experience, are generally skeptical towards the advertisements or contest their relevance [157]. Further, if a user perceives an interruption in his primary interaction objective or considers ads to clutter his workspace, advertisements are more likely to be blocked or ignored [103].

Further, sophisticated online advertising approaches such as personalized, behavioral or targeted delivery mechanisms rely on the collection and use of data about users' Web interactions. Different studies have documented users' misgivings and privacy concerns about these practices. For example, in an interview study of 1000 adult consumers, 66% objected to tailored ads [215]. Due to the pervasiveness of these concerns, (self-)regulatory and technical proposals are under consideration, e.g., that would allow users to opt-out from such data collection practices by signing up for a Do-Not-Track list [106]. At the same time, users can attempt to block advertisements altogether when suspecting that they are triggered by the tracking of their online trails. In addition to privacy issues, online advertisements also present security threats. Infected online ads are often used to compromise ad viewers' machines and spread malware [206] or direct the machines to participate in ad-fraud scams. Users do not even have to click on ads to trigger malware and the consequences can be devastating. In a sophisticated ad-fraud scheme discovered in 2012, shutting down malicious servers that orchestrate the fraud and control victims' machines would lead to all the victims losing their Internet service [74]. Most of these users were even unaware that their machines have been infected and mitigation of the effects of the scam represented a big challenge.

A survey of 1543 AdBlock Plus users further evidenced that privacy and security concerns are major factors to select this application [180]. Avoiding distractions and improving website load time performance, however, are the dominating reasons. Interestingly, the lowest score of importance was given to ideological reasons. See Table 5.1 for the full results [180].

Table 5.1: Survey results: Why do consumers use Adblock Plus?

Reasons	**No Opinion**	**Not Important**	**Somewhat Important**	**Important**
Distracting animations and sound	4.3%	5.6%	15.6%	74.5%
Offensive/inappropriate ad content	8.0%	20.1%	23.3%	48.6%
Reduce page load time and bandwidth use	5.7%	10.1%	22.6%	61.6%
Missing separation between ads and content	13.2%	11.5%	27.5%	47.8%
Privacy concerns	8.3%	9.9%	27.5%	54.3%
Security concerns	8.0%	9.7%	26.1%	56.3%
Ideological reasons	20.2%	32.0%	24.2%	23.7%

5.3.2 What Technologies Are Involved?

Ad-blocking technologies prevent online ads from being downloaded and displayed on users' screens as they browse the Web. They can also be considered privacy-preserving tools as some forms of online tracking (e.g., via cookies) can be evaded. Typically, ad-blocking tools are available as free downloadable plug-ins and exist for several Web browsers. For example, AdblockPlus is open-source and maintained by an international community of voluntary helpers. Internet Explorer 9 includes a directly embedded functionality primarily used for *tracking protection*, but also allows to block some unwanted content.

Ad-blocking tools rely on two mechanisms to block ads: (i) Prevent loading of elements whose URLs match *filter rules* used to classify elements as ads, and (ii) Hide page elements that match a Cascading Style Sheets (CSS) selector. Users can subscribe to different community-generated filter lists or manually specify filtering rules themselves. They can also decide to allow loading of some elements of a page or to turn-off ad-blocking on specific pages or websites. However, this feature is not widely used among Adblock users [180].

Ad-blocking causes revenue loss for advertisers and ad networks but it has the most significant impact on websites whose business model is based on online advertising. The majority of websites today rely on ad revenue, whereas only a few websites have successfully implemented subscription and membership based systems for revenue. Therefore, it is understandable that site operators might want to discourage or thwart ad-blocking. In particular, a website can detect the use of ad-blocking browser add-ons with a JavaScript that executes after the page has been loaded and verifies that the ads are indeed displayed. Then, the website could take one of the following countermeasures: (i) inform users about adverse effects of ad-blocking on the website and ask them to turn it off; (ii) prevent users from accessing the content unless they disable ad-blocking; (iii) embed ads in a way that ad-blocking filters cannot easily differentiate ads from content; (iv) tie the functionality of websites to the download of ad elements; and (v) offer users to pay subscription fees for ad-free content.

Both the blocking of ads and measures against ad-blocking currently come at a very low cost. The former requires the user to install a browser plug-in and subscribe to filter lists. As for detecting ad-blocking tools, the required JavaScript code is easily available online.

5.4 Analysis Overview and Assumptions

We propose a game-theoretic model of the informational consequences of consumers' ad avoidance and content providers' detection of these practices. In our analysis, we model the strategic interactions between a generic website W and a user U and we iteratively consider the following three cases: (i) without the presence of ad-blocking and ad-blocking detection technologies; (ii) with ad-blocking but no detection, and (iii) where both technologies are available to consumers and website owners, respectively. Throughout the rest of the chapter, we use the terms "website" and "website owner" interchangeably.

A key assumption we make is that the website attempts to analyze users *individually*. A number of technologies exist to implement various forms of conditional content and ad delivery (see, for example, [141]) ranging from tailoring a website's appearance to the type of browser and operation system in use by the consumer. Note that the individualized analysis does not necessarily translate into unique monetization strategies for each user.

Website owners can utilize two canonical types of monetization strategies in response to a particular user: either employ ad-financed content delivery or propose a micropayment for access to content (as a representative subcase of a wider range of payment-based strategies, such as subscriptions). The consideration of micropayments for newspaper content is extremely timely. Not only has the debate about micropayment schemes for news and other digital content been fought very passionately over the last few years [150, 167, 200]; But from an actual deployment point of view, easy-to-manage systems are now available, for example, One Pass from Google [40] or PayPal for Digital Goods [75]. And consumers seem more willing than ever to accept small charges in response for immediate content or entertainment needs [200].

We further assume that the website is aware of the user's valuation of content, for example, because of the cooperation with ad networks, inference about the resources the user is trying to access or previous interactions. In practice, websites work on obtaining such information and use it to, for example, compute appropriate prices for their services or content (e.g., New York Times' subscription price is based on the estimates of readers' valuations of the content, which is set such that the current paywall system should be accepted by a certain fraction of their readership [44]). Our analysis can also be easily extended to introduce uncertainty about user's content preferences from the content providers' perspective.

Not all aspects about user behavior are immediately observable without sophisticated detection technologies. In particular, the website cannot easily deduce whether the consumer is taking advantage of ad-blocking technologies. This is especially the problem in the impression-based ad revenue model, in which the website obtains ad revenue for each ad displayed to its visitors. For example, if the feedback cycle between the ad network and the content provider is not real-time then payoff consequences of ad avoidance are only realized at a later time. In the click-based ad revenue model, a website gets paid for users' clicks that get reported to the ad network, thus perhaps enabling more direct and immediate control. The absence of signals could indicate to ad networks (and websites) a change in the user's behavior (e.g., ad avoidance). The website can mitigate this information disadvantage by investing in technologies to detect ad-blocking. In this work, we focus on impression-based ad revenue model and we note that a similar analysis can be provided for the click-based ad revenue model.

Based on these assumptions, we model each website visit as a sequential game between the two players, a website W and a user U, to highlight the informational and strategic aspects of the interactions. We represent the different cases as game trees (see Figures 5.1 and 5.2)

with the notation provided in Table 5.2. In each game, the players can choose from the corresponding strategy sets and the payoffs achieved at the end of the game are represented in the format (P_W, P_U), where P_W and P_U are total payoffs of W and U, respectively.

Table 5.2: Symbols for the game-theoretic models.

Symbol	Definition
b	User's "benefit" of viewing content
c	User's "cost" of viewing ads
s	Subscription fee
r_i	Ad Network's per-impression ad revenue
C_B	Cost of using AB software
C_D	Cost of detecting AB software
α	Belief about the reached information set
AF	Ad-financed content
FF	Fee-financed content (micropayments)
DI	Invest in detection of AB software
NI	No investment in detection of AB software
B	Block ads
A	Abstain from blocking ads
P	Pay subscription
N	Not pay subscription
$(x\|y)$	First (x) and second (y) action of a player
(x, y)	Strategy profile: (first mover, second mover)

5.5 Game-theoretic Models

In this section, we introduce game-theoretic models that capture strategic interactions between W and U. For each model, we present analysis methodology and the obtained results.

5.5.1 Model 1: No Ad-blocking and No Detection

We introduce the reader to our approach by first proposing a basic model of the interaction between websites and consumers in which no ad-blocking or detection technologies are used by users and websites, respectively. Afterwards, we increase the complexity of the model to account for ad avoidance and countermeasures.

Model Setup: An Extensive Form Game with Complete Information

The content provider selects between fee-financed (e.g., micropayments) and ad-financed monetizing scheme for his content (denoted by FF and AF, respectively). If presented with a website that solicits a fee to access its content, users can decide to transmit a payment, P, or to deny payment and forfeit access, N. The website will either earn positive revenues from the ad impression, r_i, or from the subscription fee, s. The consumer receives a benefit, b, from accessing the content and pays either the fee, s, or has a cost c due to accepting ads. The

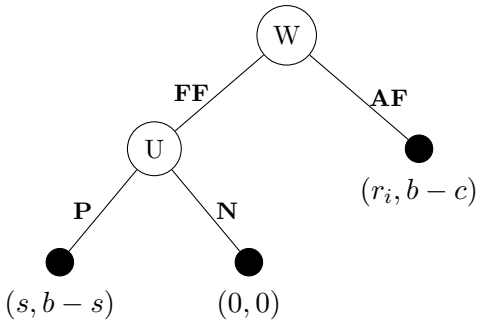


Figure 5.1: Model 1 – Game tree for the basic model with no blocking and no detection.

subscription fee s is determined by the content provider and it is the same for all the users, because it has been shown that price discrimination is not economically optimal for providers [84] and because of users' protest (e.g., case of Amazon[6]). Determining the optimal price is not the goal of this work, but is certainly worthy to explore. The cost c captures all the negative aspects of receiving ads from the users' point of view (summarized in Table 5.1). Figure 5.1 summarizes the characteristics of the basic model.

Analysis Methodology: Subgame Perfect Nash Equilibrium

The model in this section belongs to the class of *perfect and complete information extensive form* games. In these games, each player always knows the previous moves of all players when he has to make his move. In [125], it is proven that every finite extensive-form game of perfect information has a pure-strategy Nash equilibrium. We use a *Subgame Perfect Nash Equilibrium* (SPNE) solution concept that is a refinement of a Nash equilibrium used in dynamic games. In game theory, a strategy profile is a subgame perfect Nash equilibrium if it represents a Nash equilibrium of every subgame of the original game. A common method for determining SPNE is *backward induction* and we apply it in our analysis.

Backward induction can be applied to any finite game of perfect information. This technique eliminates incredible equilibria and assumes that: (i) the players can reliably forecast the behavior of other players and (ii) the players believe the other players can do the same. In the game defined by Figure 5.1, the user knows that he is the player that has the last move if the website plays FF. Hence, for each possible move of the website, the user selects his best response. For example, if the website plays FF, the user concludes that with move P he obtains the best payoff if and only if $b > s$.

Now we consider how the website chooses his best strategy using backward induction. Let us assume that $b > s$. The website then knows that if it plays FF, the user's best response is to play P, which results in the payoff of s for the website. However, if the website plays AF, its payoff would be r_i. Hence, the website's best response is FF, if $s > r_i$. In summary, if $b > s$ and $s > r_i$ strategy profile (FF, P) is the SPNE of the game in Figure 5.1.

SPNE strategy profile (x,y) represents the actions that (the first mover, the second mover) will take at the stages of the game when it is their turn to play, though it might happen that some stages are not reached during the unfolding of the game. For example, when the website plays AF, which corresponds to the strategy profiles (AF, P) and (AF, N), the user actually does not get to play his best response (i.e., P or N).

Results

Following this methodology, Table 5.3 summarizes all possible SPNE of the defined game, considering the different values of the game parameters.

Table 5.3: SPNE of Game Model 1.

$b > s$	$s > r_i$	(FF, P)
	$s < r_i$	(AF, P)
$b < s$		(AF, N)

It follows that a website would only implement fee-financed revenue scheme when users' value of the provided content is sufficiently high, $b > s$, and the expected ad-revenue does not exceed fee payments, $s > r_i$. The first condition is relatively difficult to assess for a large number of diverse users if the revenue policy cannot be set adaptively for each consumer. In contrast, the second condition allows for a more straightforward calculation – at least for an impression-based ad model. We address the impact of the heterogeneity of the user base in our simulations (Section 5.6).

5.5.2 Model 2: Ad-blocking, Detection vs. No Detection

In the following, we extend the analysis to include consumers having the opportunity to utilize ad-blocking tools and website owners to potentially respond by investing in detection technologies. The expanded game is represented in Figure 5.2.

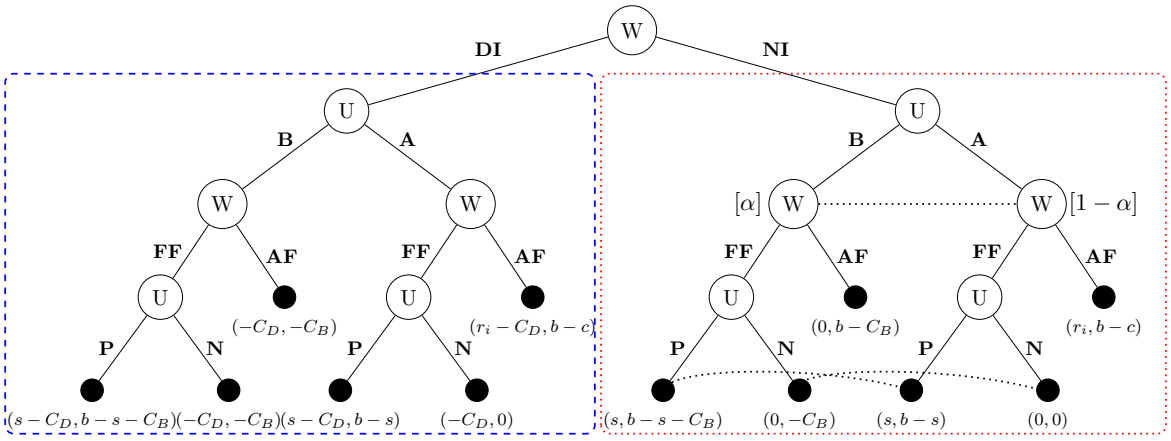


Figure 5.2: Model 2 – Game tree for ad-blocking with and without detection technologies.

Model Setup: An Extensive Form Game with Imperfect Information

Consumers now have the option to use an ad-blocking tool, B, at cost C_B, or to abstain from blocking ads, A, which does not incur any direct cost. We assume that websites are aware of the possibility of ad-blocking, but without an investment into detection technologies, NI, are not able to differentiate between users with and without ad-blocking tools and thus hold

only *imperfect* information about the user's action and its payoff consequences. In contrast, when the website is equipped with detection technologies, DI, at cost C_D, the information barrier is resolved. The informational consequences are easily discernible in Figure 5.2 by observing the dotted lines between information sets that indicate the website's uncertainty about the reached state in the game and the eventual outcomes. Websites have to formulate a probabilistic assessment α of the reached state of the game, following the user's decision to block ads or to abstain. We consider the information about the website's investment in detection technology to be common knowledge, as it is revealed after the first interactions with a couple of users. This information can then be shared among users through various (online) interactions or the users can learn directly through an earlier attempt to access the website. In addition, the website can make this information public on their websites, to inform the users about the consequences of using ad-blocking tools and to discourage such behavior.

We further break down the game into two subgames concerning the website's decision to invest or not in detection of ad-blocking tools, as highlighted by the left and right boxes in Figure 5.2, respectively. The analysis of the lefthand side subgame in Figure 5.2 (i.e., when W plays DI) is similar to the calculation of the SPNE presented in Section 5.5.1. Using the same methodology we obtain SPNE of this subgame and present the results later in Table 5.5.

Analysis Methodology: Perfect Bayesian Nash Equilibrium

The subgame in the righthand side of the game in Figure 5.2 belongs to the class of *complete imperfect sequential* games, because one player does not have information about the opponent's action played in the previous stage of the game. In other words, the website does not know whether the user has already installed an ad-blocking tool or not, when it should choose the monetization strategy for its content (i.e., ad-financed or fee-financed strategy).

Next, we discuss the game-theoretic concept of the *Perfect Bayesian Nash Equilibrium* (PBNE), used to get an insight into the strategic behavior of players in such games. PBNE is a refinement of the Bayesian Nash equilibrium concept that removes implausible equilibria in sequential games [127]. More specifically, the concept of PBNE is defined by four Bayes requirements that eliminate unwanted subgame-perfect equilibria [179]. We discuss these requirements considering the defined subgame represented in the righthand side box in Figure 5.2.

Requirement 1: The player with the move must have a belief about which node in the information set has been reached by the play of the game. For example, in Figure 5.2 the website believes that the user blocks ads with a probability of α.

Requirement 2: At the PBNE strategy profile, players must be sequentially rational given the players' beliefs. A strategy profile is sequentially rational if and only if the expected payoff of the player who has the move at that information set is maximal given the strategies played by all the other players. For example, in Figure 5.2 the website should calculate its expected payoff for playing AF and FF, given its belief α, and choose the strategy that maximizes its expected payoff. Given the website's belief and assuming that the user would accept to pay the subscription (i.e., $b > s$), the expected payoff from playing FF is $\alpha \times s + (1-\alpha) \times s = s$. The expected payoff from playing AF is $\alpha \times 0 + (1-\alpha) \times r_i = (1-\alpha)r_i$. Hence if $\alpha > \frac{r_i - s}{r_i}$ and $b > s$, the website plays FF to be sequentially rational.

Requirement 3: The player must update his belief at the PBNE to remove implausible equilibria of BNE on the equilibrium path. These beliefs are determined by Bayes' rule and the players' equilibrium strategies. In other words, players should first calculate the

equilibrium paths of the complete perfect information game. If the calculated strategy that satisfies sequential rationality is on the equilibrium path, there is no uncertainty for the player at the PBNE (i.e., α equals 0 or 1).

Requirement 4: Finally, the belief should be updated considering the sequential rationality and players' equilibrium strategies where it is possible.

A player's strategy profile $(x|y)$ corresponds to the actions that the player will take at the (first|second) stage of the game when it is his turn to play. In the righthand subgame presented in Figure 5.2, if $b > s$, $s < r_i$, and $C_B > c$, there exists an equilibrium path of $(A|P, AF)$. Although the user cannot play P when the website makes use of the AF strategy, we use the $A|P$ notation to represent the full strategy profile of the user on the equilibrium path. This means that if $\alpha < \frac{r_i - s}{r_i}$, the PBNE is $(A|P, AF; \alpha = 0)$ (i.e., Requirement 3). Requiring that each player has a belief and acts optimally given this belief suffices to eliminate the implausible equilibria for the belief of $0 < \alpha < \frac{r_i - s}{r_i}$. But, if $\alpha > \frac{r_i - s}{r_i}$, the PBNE is $(A|P, FF; \alpha)$, because we cannot eliminate any implausible equilibria for this strategy profile (i.e., Requirement 4). Similar calculations can be made for other cases.

Results

Applying this methodology, we derive results presented in tabular fashion for the righthand side (Table 5.4) and the lefthand side (Table 5.5) subgames in Figure 5.2.

If websites do not invest in detection, we observe that ad-blocking happens in two instances (see Table 5.4). First, when users do not value the content highly enough to pay a fee ($b < s$), and ad-blocking is cheap relative to the "cost" of viewing ads ($C_B < c$). Second, if websites believe it to be unlikely that users block ads ($\alpha < \frac{r_i - s}{r_i}$) and ad-blocking is cheap, then ad avoidance can persist even when users value the content sufficiently ($b > s$). In both cases, the user exploits his information advantage to avoid ad clutter, while the website gains nothing through the interaction (because it mistakenly relies on ad-financed monetization strategy).

Table 5.4: PBNE of submodel without detection.

		$C_B < c$	$C_B > c$
$b > s$	$s > r_i$	(A|P, FF; $\alpha = 0$)	(A|P, FF; $\alpha = 0$)
	$s < r_i$	(A|P, FF; α) $\alpha > \frac{r_i - s}{r_i}$ (B|P, AF; α) $\alpha < \frac{r_i - s}{r_i}$	(A|P, FF; α) $\alpha > \frac{r_i - s}{r_i}$ (A|P, AF; $\alpha = 0$)
$b < s$		(B|N, AF; $\alpha = 1$)	(A|N, AF; $\alpha = 0$)

In contrast, with an investment in detection technology, the website can partially crowd out the drawback of ad avoidance. He can successfully solicit a micropayment even when ad-blocking tools are cheap, as long as the user values the content sufficiently (see Table 5.5). However, the website will still not extract any benefits from a user who does not value the content highly and has access to cheap ad-blocking tools. Interestingly, the website is indifferent in the latter case about allowing the user to access the content freely (with blocked ads) or not. Importantly, the introduction of detection technology also lowers the threshold of what a user considers to be a cheap ad-blocking, i.e., the consumer now internalizes the cost of the expected price of the micropayment when making the assessment ($C_B < c - s$).

Table 5.5: SPNE of submodel with detection.

		$C_B < c - s$	$C_B > c - s$
$b > s$	$s > r_i$	(A|P, FF)	
	$s < r_i$	(B|P, FF)	(A|P, AF)
		$C_B < c - b$	$C_B > c - b$
$b < s$		(B|N, FF) (B|N, AF)	(A|N, AF)

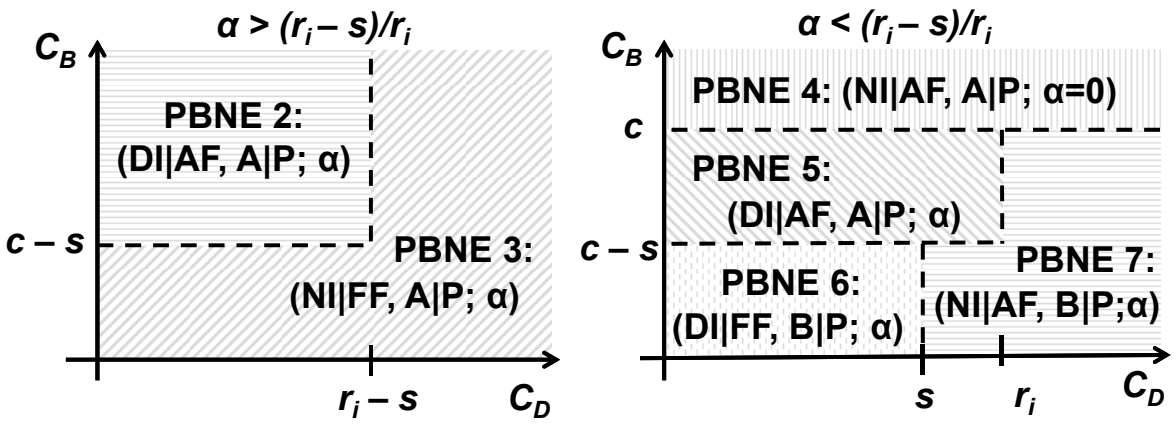


Figure 5.3: Game outcomes when users value the content sufficiently to pay subscription fees ($b > s$) and the website prefers ad-financed to fee-financed monetization strategy ($r_i > s$).

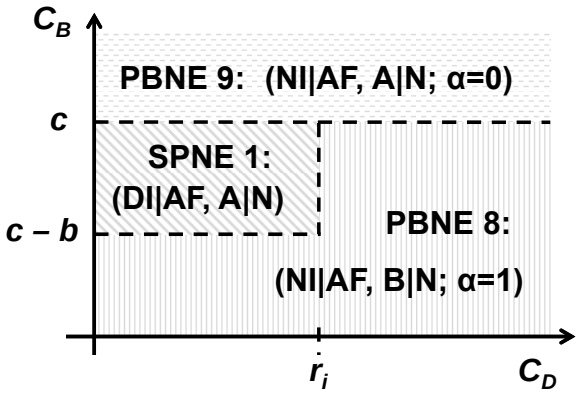


Figure 5.4: Game outcomes when users do not value the content sufficiently to pay subscription fees ($b < s$).

We now proceed to visualize the space of equilibria from a different perspective in Figures 5.3 and 5.4 by integrating the results of the subgames from the lefthand and righthand side

of Figure 5.2. The figures show how the equilibrium strategies of the players depend on the cost of detection, C_D, and ad-blocking, C_B, technologies, respectively. We break down the results based on the equilibrium beliefs of the website, i.e., Figure 5.3 is split according to the threshold belief, $\alpha^* = \frac{r_i - s}{r_i}$. Figure 5.4 shows the cases where the website is certain about the user's strategies. In addition (and not visualized), for the case of high content value, $b > s$, and low ad-revenue, $s > r_i$, we find that the website and the user select PBNE 1 : $(NI|FF, A|P; \alpha = 0)$, independently of C_B and C_D.

5.6 Simulation Approach and Results

Our analysis in Section 5.5 provides a framework that content providers can use to determine which countermeasures concerning ad avoidance they should use to maximize their revenue. Our results show that the best response depends on the type of users that a given website serves. In this section, we illustrate how our framework can be used to determine the best response while taking into account different assumptions about user heterogeneity with respect to user perception of content and ads.

5.6.1 Simulation Setup

We model the application of our framework to a popular website with specific and unique content that is of a high value to its visitors (e.g., Financial Times). Financial Times is a good example as it is a website that deploys both monetization strategies: fee-financed and ad-financed. Our game-theoretic analysis shows that the outcome of the game depends mostly on the parameters that characterize the visitors of a given website: users' benefit of viewing the content, users' cost of viewing ads with the content and ad revenue that the website earns for each pageview. As discussed in Section 5.4, the values of per-impression ad revenue and users' benefit of viewing the content are available to the stakeholders, namely websites and ad networks. It is more difficult to obtain exact values for users' cost of viewing ads and to do so, websites could perhaps position themselves with respect to the reasons users have named in the survey on why they block ads (Table 5.1). Depending on how much they match users' criteria, they can estimate their visitors' costs. In addition, as we will show, knowing the distribution of such a variable for which the relevant parameter is the fraction of users who use ad-blocking tools (e.g., available from Firefox statistics) is sufficient for the model.

We rely on Web analytics providers, Alexa and Google's DoubleClick Ad Planner, to obtain the data based on which we can estimate the parameter values. We make the following assumptions:

1. The website receives 1 million pageviews per day, as reported by Google's DoubleClick Ad Planner [2].

2. In the case of fee-financed content, we consider a micropayment of $s = \$0.32$ per pageview. We compute this value based on the Financial Times' subscription fee of \$4.99 per week [62] and an average of 2.22 pageviews per visitor per day, as reported by Alexa [55]. As explained in Section 5.5, the subscription fee is the same for all users.

3. We model the impression-based ad revenue per pageview with a beta distribution represented in Figure 5.5, based on an estimated cost-per-mille (CPM) between \$1 and several tens of dollars [36]. CPM is the cost that advertisers pay for thousand impressions and

thus we compute the per-pageview ad revenue as $CPM/1000$ for the considered values of CPM. We select a skewed distribution, as most advertisers pay CPM in the range of a couple of dollars and only very few major advertisers pay a high CPM (in the order of tens of dollars). The total ad revenue that the website can earn in our model is in the range of the reported ad revenues of the top blog websites [3] with a similar number of daily pageviews [2].

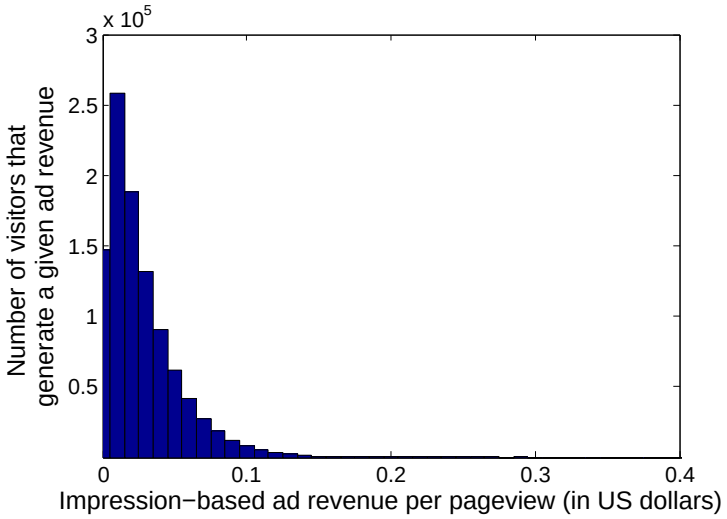


Figure 5.5: Distribution of user-generated impression-based ad revenue per pageview.

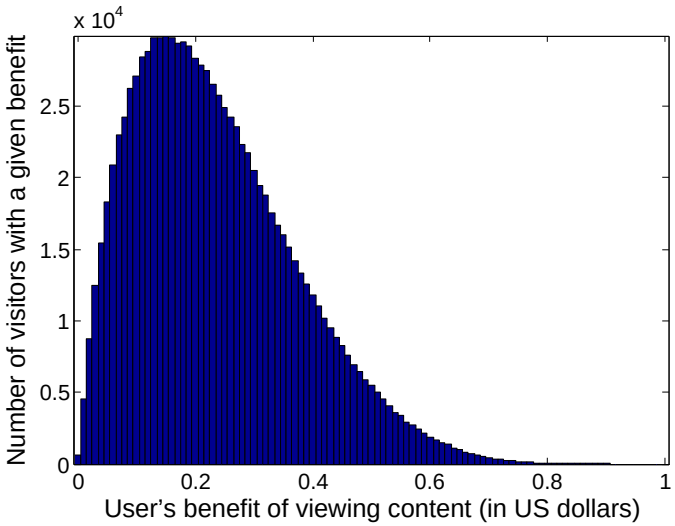


Figure 5.6: Distribution of users' benefits of viewing content per pageview.

4. Benefit b (expressed in US dollars) of users viewing the content (Figure 5.6) is drawn from a beta distribution (in the range of values comparable to the impression-based ad revenue per pageview), such that 25% of the visitors would opt for fee-financed content (i.e., has $b > s$). This number is in compliance with 25% of Financial Times's

visitors paying for digital subscriptions [61]. In addition, for most of the websites, users' benefits are high, due to users' self-selection bias. The knowledge of the exact values is not required, the important parameter is the fraction of users accepting to pay the subscription fees.

5. We consider a population of visitors that consists of: (i) a fraction $(1-\gamma)$ of users who are indifferent about ads and therefore do not use AB software and (ii) a fraction γ of users who are heterogenous in how much they like or dislike ads and therefore might use ad-blocking tools. Users who are indifferent about ads associate a small cost (expressed in US dollars) to viewing online ads. Other users, who are not indifferent about ads, have a higher cost of viewing ads, that can even surpass the benefit they associate to viewing the content. However, it does not necessarily mean that all of them block ads. Their decision on whether to use ad-blocking tools (Block) or not (Abstain) depends on the cost of viewing ads with respect to the values of other parameters (e.g., their valuation of the content or the cost of ad-blocking). Therefore, the parameter c that represents users' costs of viewing ads is drawn from a bimodal distribution (Figure 5.7), that assigns a small cost to the users indifferent about ads (the first mode of the distribution) and higher costs to other users (the second mode). The values of c are in the range comparable to the impression-based ad revenue per pageview and users' valuation of the content. Figure 5.7 depicts the distribution for $\gamma = 0.5$. We vary the value of γ in the simulations.

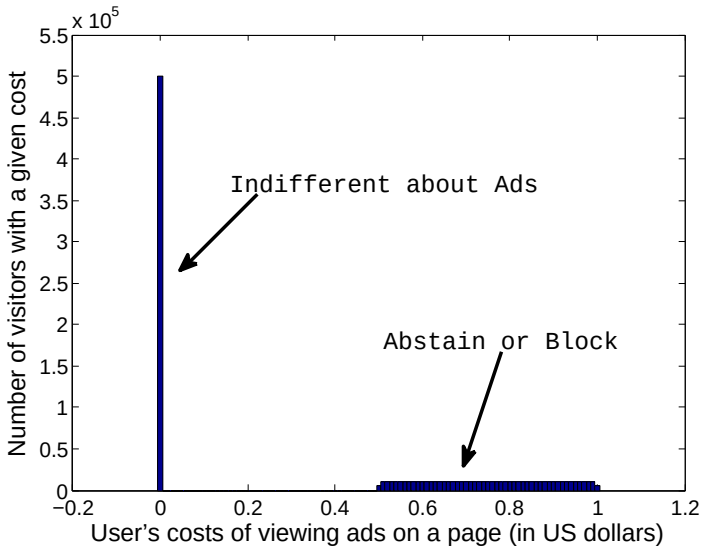


Figure 5.7: Distribution of users' costs of viewing ads per pageview: Fraction $(1-\gamma)$ of users indifferent to ads; Fraction γ of users who choose between Abstain (no ad-blocking tools) and Block (using ad-blocking tools).

6. In practice, the cost of blocking ads (C_B) corresponds to the cost of installing and maintaining a browser add-on and subscribing to filter lists that define blocking rules. At the moment, the cost (C_D) of detecting ad-blocking tools on users' machines corresponds to the cost of including a specific JavaScript into Web pages. Nowadays, both of these costs (expressed in US dollars) are very small and we use values of $C_B = \$0.01$ $C_D =$

$0.001 for our simulations. Note that these values represent costs per interaction and have such a low value as they are factored out on millions of users (for C_D) and the number of pageviews per day (for C_B). These costs could increase if an arms race develops between ad-blocking and detection technologies, as it was the case with pop-up ads and pop-up blockers [8]. We evaluate the effect of higher costs of ad-blocking and detection technologies later in the analysis.

5.6.2 Results

We simulate the interaction between the website and the population of users, based on our game-theoretic model and parameter values described above. The website treats each user individually and applies the framework to each of the visitors. We then aggregate the results of the interactions to represent the outcomes for the entire population of visitors. The fraction (γ) of users that might potentially install an ad-blocking tool is a variable in our simulations. For each value of $\gamma \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$, we generate a corresponding bimodal distribution (as in Figure 5.7) that assigns the values to users' costs of viewing ads (c). The values of all other parameters remain fixed.

First, we compare the revenues that the website obtains by deploying three different monetizing strategies: (i) serving ad-financed content (AF model) to all visitors, regardless of whether they block ads or not; (ii) serving fee-financed (FF) content, where users have to pay a subscription fee in order to access the content; (iii) game-theoretic approach (GT model) where a website chooses an appropriate strategy according to our analysis, and can either serve ad-financed or fee-financed content to different users.

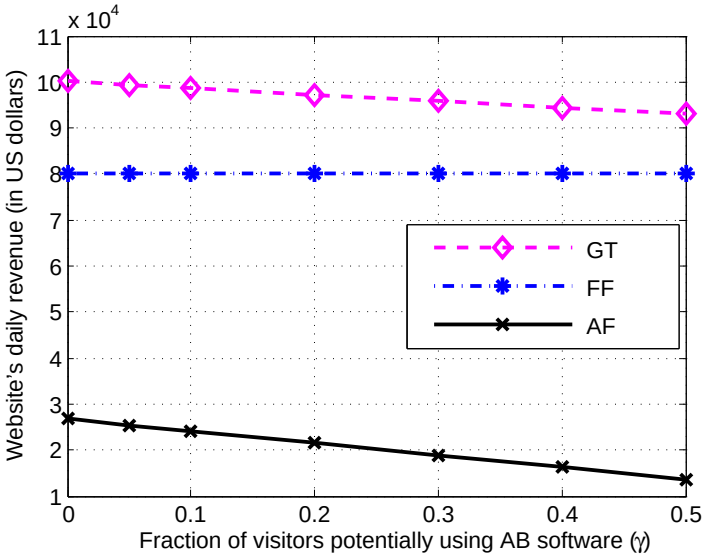


Figure 5.8: Website's daily revenue (in US dollars) with different monetizing models.

Figure 5.8 depicts the daily revenue of the website, for the three models, depending on the fraction of users that might block ads. We observe that the website's revenue obtained with the GT monetizing model is superior to using pure fee-financed (FF) or ad-financed models (AF). The rationale behind such a result is as follows. In the AF model, users with ad-blocking tools do not generate ad revenue for the website, as ad impressions are blocked

on their machines. The higher is the potential number of users with ad-blocking tools (γ), the higher is the revenue loss for the website. In the FF model, only users who value the content more than the subscription fee are willing to pay, thus the revenue is not influenced by the users who block ads, only by the number of subscriptions. FF revenue depends on the subscription fee that the website can charge, which mostly depends on the content it serves and how valuable it is to its visitors. The GT model represents a compromise between AF and FF models. For ad-adverse users who value the content enough to pay subscription fees, the website applies the FF strategy. With AF, the website cannot benefit from these users as they block ads. For users who do not dislike ads as much, the website might either use FF or AF strategy, whichever is more profitable. Thus, the GT model enables the website to take into account users' heterogeneity and maximize its profit.

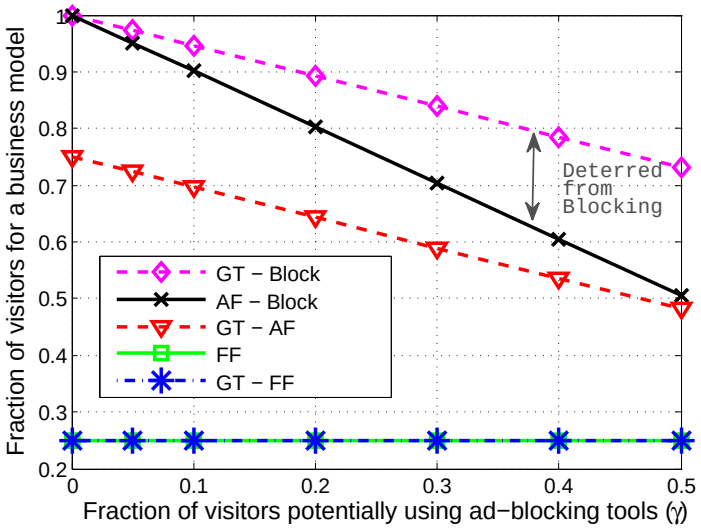


Figure 5.9: Fraction of visitors that generate revenue for each monetizing model.

In Figure 5.9 we show the fraction of users that generate profit for the website with the three monetizing models. The curve labeled *AF-Block* represents the fraction of users from which the website profits in the AF model. In this model, the ad revenue is generated only by the users without ad-blocking tools. Note that nevertheless all users obtain the content. The difference between the *AF-Block* curve and 1 corresponds to the fraction of users who block ads in the AF model. In the fee-financed (FF) model, only the users who opt to pay the subscription generate the revenue for the website and obtain the content (*FF* curve[2]). In the GT model, the website profits from serving ad-financed content to a fraction of users (*GT-AF* curve) and fee-financed content to another fraction of users (*GT-FF* curve). The sum of these two corresponds to the total fraction of users that the website can profit from, represented with *GT-Block* curve. The remaining fraction of users (i.e., the difference between the *GT-Block* curve and 1) corresponds to the users who block ads in the GT model. Users served with ad-financed content are those who: (i) accept to view ads in exchange for free content (corresponding to PBNE 9 : $(NI|AF, A|N; \alpha = 0)$), or (ii) value the content more than they dislike ads, but not enough to pay the subscription fee for ads-free content (corresponding to SPNE 1 : $(DI|AF, A|N)$). Users who are served fee-financed content are

[2]Note that *FF* curve overlaps with *GT-FF* curve.

those who: (i) dislike ads but value the content, or (ii) users who accept ads but also value the content, thus leaving the choice to the website that can decide to offer the subscription model to such users as it might be more profitable. For these users the outcome of the game is PBNE 1 : $(NI|FF, A|P; \alpha = 0)$. We observe that the total fraction of users that generate profit for the website in the GT model (*GT-Block*) is higher than in either AF or FF model.

Users who do not generate revenue and do not obtain the content in the GT model are ad-adverse users who do not value the content enough to pay subscription fees (outcome PBNE 8 : $(NI|AF, B|N; \alpha = 1)$). Note that the impact of ad-blocking tools is smaller in the GT model, and in the worst case about 27% of users block ads (and generate revenue loss for the website) compared to the 50% in the AF model. These results are in line with the results in Figure 5.8 and explain why the website earns more with the GT model. In the worst case, the GT revenue is around 16% higher than FF revenue and it might not seem justified to deploy the GT model for that increment in the revenue. However, the major advantage of the GT model is that it maximizes the number of users who obtain the content (73% in the GT model compared to 25% in the FF model, in the worst case). We conclude that the GT model allows the website to adapt its monetizing strategy such that it maximizes the number of visitors from whom it profits, as well as its visibility or impact factor.

As discussed previously, the website can make it more difficult for ad-blocking tools to filter out and block ads. In our GT model, this action can be represented with an increase in the users' cost of blocking ads and a higher investment in the detection. We simulate the effect of a higher ad-blocking and investment costs ($C_B \in 0.01, 0.1, 0.5, 0.7, 1$ and $C_D = \$0.1$)) and represent the results in Figure 5.10. Different curves correspond to the fraction of users that the website can profit from in the GT model, considering a different cost of ad-blocking. We observe that the fraction of users that will block ad-financed content decreases with the increase in the cost of blocking ads. As both the website and users are behaving strategically in the GT model, with the higher cost rational users deter from blocking ads and it shows that the website has a good return-on-investment with making ad-blocking more difficult.

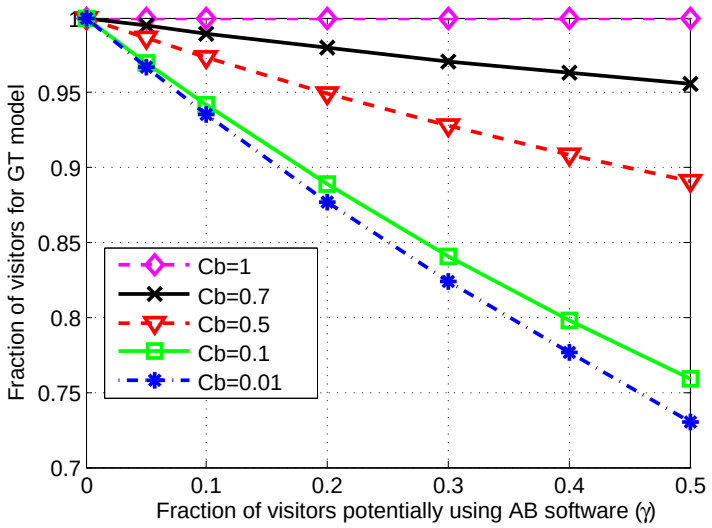


Figure 5.10: Fraction of visitors that generate revenue in the GT model, considering higher ad-blocking and detection costs.

In summary, we have illustrated how a website can use our framework in practice as a

decision help in addition to the content provider's overall business strategy and factors that are outside the scope of our model. We have demonstrated how a website maximizes its revenue with a strategic choice of its best response when facing users with different preferences with respect to ads and content. Such a strategic behavior enables the website to maximize the number of users from which it can benefit, as well as to apply the strategy that maximizes the profit. Users' strategic behavior enables them to maximize their utility as well, by having a choice of viewing ad-financed or fee-financed content.

5.7 Summary

In this chapter, we conducted a systematic study of the consequences of ad avoidance on the business model of content providers. We developed a framework usable by content providers to ponder their options to mitigate the consequences of ad-avoidance technologies. We carefully devised and analyzed a game-theoretic model of the impression-based ad revenue mechanism and illustrated with simulations the impact of different strategies under parameter assumptions motivated by real-world data. Our analysis shows that deploying a game-theoretic approach, i.e., strategically applying fee-financed or ad-financed monetization strategy, and treating each user individually yields higher revenues for publishers, compared to deploying one strategy across all users. Also, understanding the distribution of users' aversion to ads and valuation of the content is essential for publishers to make a well-informed decision. We expect that our modeling and simulation assumptions are a reasonable, but likely not a perfect fit for every situation involving content providers and ad avoiders.

Our contribution is only a first step to account for the complicated interactions between ad avoidance and online content monetization. For example, a promising area for additional work is to more carefully address the impact of the negative feedback spiral caused by the adoption of ad-avoidance under the presence of limited information. A loss of revenue through an increase of website visitors who use ad-blocking tools will frequently trigger a more aggressive pursuit of advertisement opportunities. Those might even include consumer-unfriendly affiliate marketing schemes. While this may create short-term benefits, additional consumers will depart or try to avoid these practices. In addition, we aim to consider measures of concentration and interdependency in the advertising industry. For example, a recent study shows that Google-controlled cookies were present on 97 of the top 100 websites [94]. The same study also documents the growing intricacy of tracking attempts that will make it very difficult for users to find adequate countermeasures in absence of market (self-)regulation.

In conclusion, we expect content providers that serve a technology-minded audience to suffer most from ad-avoidance technologies. And, in absence of a broad consensus between the advertising and content industry, on the one side, and consumers, on the other side, the trend towards blocking of advertisements is likely to grow. Resistance to user tracking and the desire for ad avoidance are tightly interwoven, even though we do not model the related long term trends in the moment, i.e., users rarely become technology-savvy ad avoiders over night. However, the potential for a significant shift in consumer behavior is large and should not be under-appreciated.[3]

Publication: [226]

[3] A 2010 study revealed that up to 40% consumers are willing to change their online behavior if advertisers were collecting data [173].

Part III

Privacy of Online Advertising

Chapter 6

Hyper Geolocalization for Location-targeted Advertising

Location privacy has been extensively studied over the last few years, especially in the context of location-based services where users purposely disclose their location in order to benefit from convenient context-aware services. To date, however, little attention has been devoted to the case of users' location privacy being *unintentionally* compromised *by others*. In this chapter, we study a concrete and widespread example of such situations, specifically the location-privacy threat created by access points (public hotspots, FON, home routers, *etc.*) using Network Address Translation (NAT). Indeed, because users connected to the same hotspot share a unique public IP address, a single user who reveals his current location to a service provider is enough to enable the provider to map the IP address of the hotspot to its geographic coordinates, thus compromising the location privacy of all the other connected users. When successful, the service provider can locate users within a few hundreds of meters, thus improving over existing IP-location databases. Even in the case where IP addresses change periodically (e.g., by using DHCP), the service provider is still able to update a previous (IP, Location) mapping by inferring IP changes from authenticated communications (e.g., cookies, usernames). Our contribution is three-fold: (i) We identify a novel threat to users' location privacy caused by the use of shared public IP addresses. Because the problem is inherent in the way networks (i.e., NAT) operate and its wide deployment, the potential impact of the threat is significant. (ii) We formalize and analyze theoretically the aforementioned problem. The resulting framework can be applied to any access-point setting to quantify the potential privacy threat. (iii) We experimentally assess the state in practice by using real traces of users accessing Google services, collected from deployed hotspots. Also, we discuss how existing countermeasures can thwart the threat.

Chapter Outline In Section 6.1 we explain the location-privacy threat that arises due to the use of shared public IP addresses and in Section 6.2 we provide the relevant background. We describe the system setting, the adversary and the threat model in Section 6.3. We formalize the problem in Section 6.4 and we analytically quantify the threat. We further evaluate the threat based on traces from deployed access points and present the results in Section 6.5. In Section 6.6, we consider possible countermeasures. Further discussion, including the implications of the proposed approach and the business opportunities created by the threat is presented in Section 6.7 and we conclude in Section 6.8.

6.1 Introduction

With the ubiquity of mobile devices with advanced capabilities, it is becoming the norm for users to be constantly connected to the Internet; users can benefit from many online services while on-the-go. Among others, location-based services (LBSs) are increasingly gaining popularity. With an LBS, users share their location information with a service provider in return for context-aware services, such as finding nearby restaurants. Users also enjoy sharing location information with their friends on social networks (e.g., Facebook and Twitter) [182]. For example, they can then find friends in the vicinity or recommend places they visit.

Although very convenient, the usage of LBSs raises serious privacy issues. Location privacy is a particularly acute problem as location information is valuable to many parties, because much of information can be inferred from users' locations (e.g., users' interests and activities). Location information is essential for many online service providers [135], especially for those whose business models revolve around personalized services. A prominent example is (mobile) online advertising, an ever-increasing business whose worldwide annual revenue is in billions of US Dollars (e.g., $31.7 billion in the US in 2011 [148]), as so-called location-specific ads based on the location information are significantly more appealing to users [35]. In 2012, Google reported on the efficiency of location-based information: 94% of smartphone users have looked for local information, 70% of them have contacted a business and 66% have visited one, and 36% have made a purchase [137]. Beyond pursuing new business opportunities that offer personalized services, parties can collect users' locations in order to track users' movements and associated activities (e.g., authoritarian regimes can check for participation in political gatherings).

Typically, users willingly disclose their location only to LBS service providers. Yet, non-LBS service providers can obtain users' locations through *IP-location*: determining the location of a device from its IP address. Existing IP-location services rely either on (i) active techniques, typically based on delay measurements [156, 228], or (ii) passive techniques, consisting of databases with records of IP-location mappings [72, 69]. Active techniques provide more accurate results than the passive ones, however they incur high measurement overhead and a high response time (in the range of several seconds to several minutes) to localize a single IP address. A passive approach is several orders of magnitude faster and thus preferred by service operators. A number of IP-location databases are available, either free (e.g., HostIP [67], IPInfoDB [70]) or commercial (e.g., MaxMind [72], IP2Location [69]). However, they provide a country-level, and at most a city-level, accuracy and the majority of the entries refer only to a few popular countries [183]. For instance, MaxMind reports to correctly geo-locate, within a radius of 40 km, 81% of IP addresses in the US and 60%-80% in Europe. This level of accuracy is only effective for regional advertising but is not sufficient for local businesses (e.g., coffee shops) which require neighborhood or street-level accuracy [35]. Thus, major Web companies, including Google, are actively working on improving IP-location[1].

Another way for the service providers to obtain a user's location is via transitivity, relying on other users to disclose their location and that of others in their vicinity: if a provider knows the location of user B and that user A is close to user B, the provider knows roughly the location of A. Examples of such situations are when users report neighboring users (e.g., Bluetooth), or *check-in* on online social networks and tag friends who are with them. Online

[1] Google reports an accuracy of 95% at the region-level and 75% at the city-level, with high variance across countries, and seeks to improve it to the street-level [136].

social networks can also infer a user's home address by correlating it with those of his friends [96]. In some cases, even if the proximity information is not directly revealed by users, the service provider is still able to infer it, as we will show.

In this chapter, we study a location-privacy threat users are exposed to on a daily basis. When a user connects to the Internet through the same access point (AP) as other users (e.g., a public hotspot, home router) who make LBS queries, the service provider learns the user's location. Indeed, because all of the devices connected to a public hotspot, implementing network address translation, share the AP's public IP address, when users generate LBS queries, the service provider learns the fine-grained geographic location of the AP and maps it to the AP's public IP. IP addresses remain the same for a certain amount of time, therefore for any connection for which the source IP is the same as the AP's IP, the service provider can conclude that the device is located nearby the location of the AP. The accuracy of the estimated location depends on the range of the AP (typically under one hundred meters) and on the accuracy of the locations reported by users in LBS queries (typically under ten meters with GPS-geolocation). Thus, it is significantly more accurate than the existing IP-location databases. Unfortunately, the user is usually not aware of this threat and, more importantly, protecting his location privacy is no longer in the user's control. The fact that the threat is based on observing the user's IP, which might be inferred, e.g., by using a Java applet [176, 181], even when the client tries to hide it, makes the threat even more difficult to evade.

The (IP, Location) mapping the service provider obtained for the AP stays valid until the IP changes. Dynamic IP addresses (provided that IPs are allocated to geo-diverse hosts), short DHCP leases, and systematic assignment of new IPs upon DHCP lease expirations therefore have a positive effect on location privacy. However, even when the IP is renewed and changes, service providers have means to learn about the IP change, for example, due to the widespread use of *authenticated* services (e.g., e-mails, online social networks). Consider a user connected to the AP who checks his e-mail shortly before and after an IP change. As a unique authentication cookie is appended to both requests, the service provider can conclude that the same user has connected with a new IP and can therefore update the (IP, Location) mapping with the new IP. More precisely, user requests do not need to be authenticated, it is sufficient that the service provider is able to link the requests to the same user, based on cookies, user agent strings, or any fingerprinting technique, e.g., [237].

Our contribution is three-fold: (i) We identify the location-privacy threat that arises from the use of shared public IPs. Because the problem is inherent in the way networks (i.e., NAT) operate and its wide deployment, the potential impact of the threat is significant. The expected accuracy of locating affected users is about few hundreds of meters. (ii) We formalize and analyze the problem theoretically and we provide a framework to estimate the location-privacy threat, namely the probability of a user being localized by a service provider. The framework is easily applicable to any access point setting: it employs our closed-form solution and takes as input an AP's parameters (i.e., a few aggregated parameters that can be easily extracted from logs, such as user connection and traffic rates) and it quantifies the potential threat. It is a light-weight alternative to extensive traffic analysis. The framework thus constitutes a valuable input to model sporadic location exposure. (iii) We evaluate experimentally the scale of the threat based on real traces of users accessing Google services, collected for a period of one month from deployed hotspots. Even at a moderately visited hotspot, we observe the large scale of the threat: the service provider, namely Google, learns the location of the AP only about an hour after users start connecting and within 24 hours it can locate up to 73% of the users. Finally, we discuss how existing countermeasures could

thwart the threat. To the best of our knowledge, this is the first work that addresses the problem of users' locations being exposed by others at NAT access points.

6.2 Background

In this section, we provide relevant background on the technical aspects underlying the considered problem.

IPv4 (public) Address Allocation To communicate on the Internet, hosts need public IP addresses. An IP can be either *static*, i.e., permanently fixed, or *dynamic*, i.e., periodically obtained from a pool of available addresses, typically through the Dynamic Host Configuration Protocol (DHCP). The host can use the IP for a limited amount of time specified by the *DHCP lease*. For convenience, upon DHCP lease expiration, hosts are often re-assigned the same IP. A large-scale study shows that over the period of one month, less than 1% of clients used more than one IP and less than 0.07% of clients used more than three IP addresses [100]. More than 62% of dynamic IPs on average remain the same over a period of at least 24 hours [236].

Network Address Translation (NAT) In order to cope with IP address depletion, Network Address Translation was introduced [208]. NAT hides an entire IP address space, usually consisting of private IP addresses, behind one or several public IPs. It is typically used in Local Area Networks (LANs), where each device has a private IP, including the gateway router that runs NAT. The router is also connected to the Internet with a public IP assigned by an ISP. As traffic is routed from the LAN to the Internet, the private source IP in each packet is translated on-the-fly to the public IP of the router: traffic from all of the hosts in the LAN appears with the same public IP – the public IP of the NAT router. A study shows that about 60% of users are behind NATs [100].

Geolocation Mobile devices determine their positions by using their embedded GPS or an online geolocation service. With a GPS, the computation takes place locally by using satellites positions and a time reference. Commercial GPS provides highly accurate location results (less than ten meters) [217], especially in "open sky" environments. With online geolocation services (e.g., Skyhook) a device shares some information about its surroundings, typically a list of nearby cell towers and Wi-Fi APs together with their signal strengths, based on which the geolocation server estimates the location of the device by using a reference database. This database is built typically by deploying GPS-equipped mobile units that scan for cell towers and Wi-Fi APs and plot their precise geographic locations. In addition, they take into account input reported by users with GPS-equipped devices who provide both their coordinates and the surrounding parameters. The accuracy of such systems is in the range of 10 meters [80].

Note that Skyhook cannot be used by a service provider to infer users' locations from their IP addresses. Indeed, Skyhook provides only APs' MAC address to location mappings and the service provider does not know the MAC addresses of a user's neighboring APs (not even the MAC address of the AP the user connects from) unless the user explicitly discloses them, allowing to be geo-located.

6.3 System Model

In this section, we elaborate on the considered setting, notably NAT access points, the location-privacy threat, and the adversary.

6.3.1 Setting

We consider a *NAT Access Point* setting, a prevalent network configuration, where users connect to the Internet through an access point (AP), such as a public hotspot, a home (wireless) router or an open-community Wi-Fi AP (e.g., FON [63]), as depicted in Figure 6.1. An AP, located at (x_1,y_1), is connected to the Internet by a given Internet Service Provider (ISP) and provides connectivity to the authorized users. The AP has a single *dynamic public* IP address that is allocated with DHCP by the ISP: The AP's public IP address is selected from a given DHCP pool of available IP addresses and is valid during the DHCP lease time. When connecting to the AP, each device is allocated a *private* IP address and the AP performs a Network Address Translation (NAT). Consequently, in the public network, all of the connections originating from the devices connecting through the AP will have the same source IP, which is the public IP address of the AP.

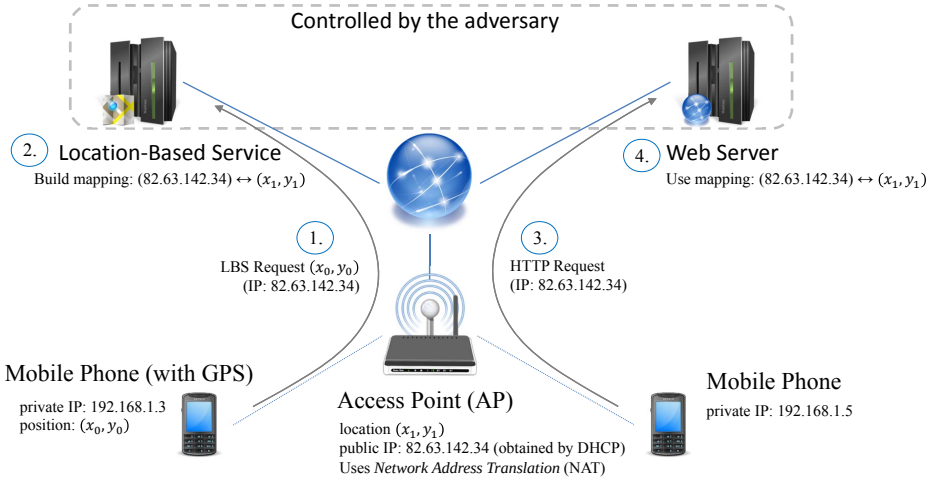


Figure 6.1: System and threat model. Devices connect through a NAT Access Point and share a single public IP address. A user making an LBS request reveals his location (close to the AP) to the adversary (step 1) who can then build the (IP,Location) mapping (step 2). When another user connects to a different server (controlled by the adversary) (step 3), the adversary can use the (IP,Location) mapping to locate the user because he connects with the same IP address (step 4).

While connected to the Internet through an AP, users make use of various online services including search engines, e-mail, social networks, location-based and online geolocation services. Services can be used either in an authenticated (e.g., e-mail) or unauthenticated way (e.g., search). We consider that the requests a server receives from the devices connected to the AP are of the following types:

1. Geolocation requests: `Geo-Req(MACs)`, where MACs refer to the MAC addresses of the APs and cell towers in the range of the device;

2. LBS requests: `LBS-Req(`(x_0, y_0)`)`, where (x_0, y_0) denotes the coordinates of the device[2] (assumed to be close to the AP's location (x_1, y_1)) provided by the user;

3. Authenticated standard (i.e., that are neither LBS nor Geolocation) requests: `Auth-Req(`*tok*`)`, where *tok* represents any information that allows for user authentication or linkability of user requests (e.g., a cookie or a username);

4. Unauthenticated standard requests: `Req()`.

With LBS requests, the service provider obtains the user's location under several forms and by different means. The user can specify his position in free-text (e.g., "bars close to Park and 57th, Manhattan") or by pin-pointing his position on a map. The location can also be determined by the user's device using one of the techniques described in Section 6.2 and communicated to the service provider by a mobile application (e.g., Maps) or by his browser through the HTML 5 `geolocation.getCurrentPosition` JavaScript function [146] used by websites. Note that non-LBS applications and websites might access and send the user's location to the service provider as well.

Both `Geo-Req` and `LBS-Req` contain an estimate of the AP's coordinates, thus they both enable the server to build the (IP, (x_1, y_1)) mapping. Consequently, there is no need to distinguish between these two types of requests, and we simply refer to both as LBS requests. For all four types of connections, the server knows the source IP addresses, specifically the AP's public IP address.

6.3.2 Adversary and Threat Models

We consider an adversary whose goal is to learn users' current locations, for instance, to make a profit by providing geo-targeted (mobile) ads and recommendations (e.g., a private company). The adversary has access to the information collected by a number of servers that provide online services described above. Companies, such as Google for instance, provide Web searches (Google), e-mail (GMail), social networking (Google+), and geolocation and location-based services (Google Maps). As such, it receives requests of the four types and consolidates all the information obtained [66]. The extent to which these services are used is exacerbated by their deep integration in the widely spread Android operating system. In addition, Google has an advertising network and thus has a strong incentive to obtain and monetize information about users' locations. As a matter of fact, Google is actively working on improving its IP-location based on users' traffic, in particular by mining location-related events (e.g., search queries associated with location such as "best burgers Manhattan") [136]. For instance, Google computes the distribution of the geographic origin of requests for "best burgers Manhattan" (e.g., 90% from New York City, 8% from New Jersey, etc.), based on IPs for which the location is known with high confidence as ground-truth. Such distributions are then used to geolocate a user based on his requests.

Microsoft (with Bing, Hotmail, Bing Maps, and Windows Phones) and Apple (with iCloud and iPhone) are other relevant potential candidates for the considered adversary. Besides these major companies, an alliance of service providers can be envisioned to jointly build an IP-location database: each provider contributes IP-location records of its visitors with

[2] We assume that all LBS requests concern users' actual locations, or that the server has means to distinguish between such LBS requests and other LBS requests. It is the case when the location is obtained directly using the methods described in Section 6.2, and communicated to the service provider.

known locations and benefits from the database for the IPs of users connecting from unknown locations. This joint effort can be coordinated by an ad network that is common to the participating service providers. This approach extends the potential of the threat as it increases the set of potential adversaries: it alleviates the need for each service provider to receive all three types of requests and a significant fraction of user traffic. Instead, they can to do so through aggregation.

In this chapter, we focus on the case where the adversary has access to all the four types of requests. The adversary is assumed to be *honest-but-curious*, meaning that he passively collects information but does not deviate from the specified protocol (e.g., purposely implementing active techniques to retrieve users' locations from their devices).

Given such an adversarial model, we consider the threat of the adversary who learns the location of a user without it being explicitly disclosed: The threat comes from the fact that the adversary can build mappings between the APs' IPs and their geographic coordinates based on LBS requests he receives from other users connected to the APs. Because all requests (from devices connected through the AP) share the same public IP, the adversary can subsequently infer the location of the other users. More specifically, considering the example depicted in Figure 6.1, when the LBS provider's server (assumed to be controlled by the adversary) receives an LBS request for position (x_0, y_0), which is the actual position of the user (located close to the AP) determined by his GPS-equipped mobile phone, the server can map the AP's public IP (i.e., `82.63.142.34`) to the approximated AP's location (i.e., $(x_1, y_1) \approx (x_0, y_0)$). Note that the accuracy of the AP's estimated location depends on the GPS accuracy of the user-reported location and the range of the AP. Later, when another user, connected through the AP, makes a request to a server (also controlled by the adversary), then the adversary can exploit the obtained mapping and infer from the source IP (i.e., the AP's public IP again) that the second user is at the same location (i.e., (x_1, y_1)). The adversary can subsequently provide geo-targeted ads. If the adversary is interested in tracking users, he can locate any user who makes an authenticated request before the IP changes.

We assume that the IP addresses in the DHCP pool can be assigned to clients at very distant locations [123]. For instance, some nation-wide ISPs (e.g., SFR in France) assign IPs among the whole set of their clients scattered all over the country. Consequently, the fact that the AP's public IP is dynamic limits in time the extent of the threat: If the AP is assigned a new IP by the ISP (with DHCP), the mapping built by the adversary becomes invalid, unless the adversary is able to infer the IP change. The inference can be based on authenticated requests as depicted in Figure 6.2: A request, authenticated by cookie `john@dom.com` and originating from IP `82.63.142.34`, is shortly followed by another request authenticated by the same cookie `john@dom.com` but originating from a different source IP (i.e., `82.63.140.25`). There are two options: either the AP's IP has changed or the user has moved and is now connected from a different AP. If the inference time interval (delimited with diamonds in Figure 6.2) around the IP renewal time is short enough, then the adversary can infer, with high confidence, that the IP has changed and its new value.

In summary, the problem we study is as follows. Considering a single AP, time is divided into intervals corresponding to DHCP leases, during which the AP's public IP address remains the same. At a certain point in time, the adversary knows the location of the AP associated to the IP because (i) a user made an LBS request earlier in the time interval or (ii) the adversary knew the location corresponding to the public IP address from the previous interval **and** a user made an authenticated request shortly before and after the public IP address was renewed. The location-privacy threat is to be evaluated with respect to the number of users whose

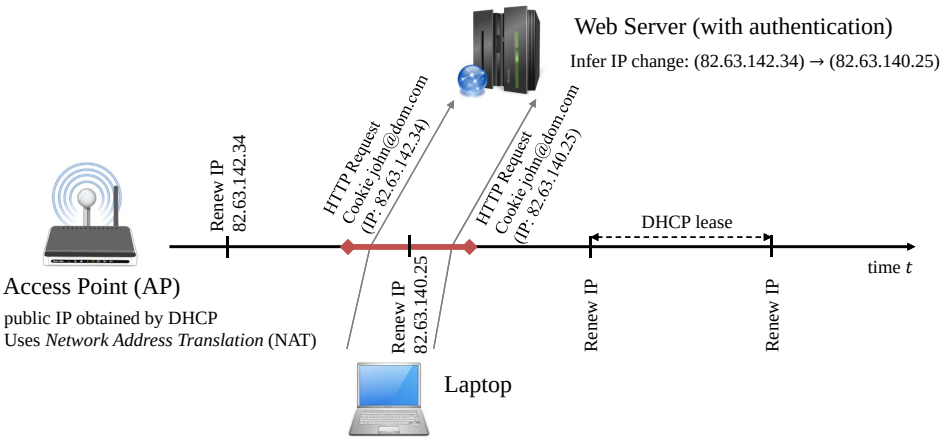


Figure 6.2: AP's IP address renewal and updating of the (IP, Location) mapping. A user generates an authenticated request (with a unique cookie) during a DHCP lease interval in which the adversary has obtained the (IP, Location) mapping, shortly before the DHCP lease expires and the AP is assigned a new IP. Shortly after the IP change, the same user generates another authenticated request (with the same cookie) from the new IP. As both requests occurred in a short time interval, the adversary can infer that the AP's IP changed from `82.63.142.34` to `82.63.140.25` and update the mapping.

locations are known by the adversary. In the case of geo-targeted mobile ads, the adversary needs to know the location of the user *when* the user makes a requests: the victims are therefore the users who make a standard request *after* the adversary learns the (IP, Location) mapping (during the same DHCP lease). If the adversary is interested in tracking users, he can maintain a log of the users who connected during a DHCP lease and sent requests, and locate them *a posteriori* if he learns the (IP, Location) mapping at some point during the same DHCP lease: the victims are the users who make an authenticated request *during* a DHCP lease in which the adversary learns the (IP, Location) mapping. In this chapter, we evaluate the threat with respect to an adversary who aims to exploit *current* location information through geo-targeted ads. However, it is possible to mount more powerful attacks on users' privacy (e.g., track users over time) based on the identified threat. We address these attacks and the resulting consequences in Section 6.7.

6.4 Formalization and Analysis

In this section, we model the aforementioned setting and we build a framework to quantify theoretically the location-privacy threat, which takes as input only a few key parameters. The used notations are summarized in Table 6.1.

6.4.1 Model

We consider an access point AP, a passive adversary $\mathcal{A}$, and a set of users who connect to AP and make requests to servers controlled by $\mathcal{A}$. We study the system over the continuous time interval $[0, +\infty)$. At each time instant t, AP has a single public IP. Every T time units, starting at time 0, the DHCP lease expires and AP is either re-assigned the same IP

Symbol	Definition
T	DHCP lease time
p_{New}	Probability of being assigned a new IP
I_k	k-th sub-interval
$\bar{t}$	Relative time within a sub-interval
λ_{Arr}	Rate of user arrivals at AP
$N_{\text{Arr}}(t)$	Number of arrivals in an interval of length t
T_{Dur}	Time users stay connected to the AP
$1/\lambda_{\text{Dur}}$	Average time users stay connected to the AP
$F_{\text{Dur}}, f_{\text{Dur}}$	Pdf/cdf of T_{Dur}
N_{Con}	Average number of users connected to the AP
$\lambda_{\text{Std}}, \lambda_{\text{Auth}}$	Rates of user standard/authenticated requests
$P_{\text{Std}}(t), P_{\text{Auth}}(t)$	Probability that a user makes at least one request during an interval t
α_{LBS}	Proportion of users who make LBS requests
λ_{LBS}	Rate of user LBS requests
Λ_{LBS}	Aggregated rate of users' LBS requests
T_{Comp}	First time an LBS request occurs in a sub-interval
$F_{\text{Comp}}, f_{\text{Comp}}$	Pdf/cdf of T_{Comp}
W	Length of the vulnerability window
ΔT	Time interval used to infer IP changes
$F_{\text{Link}}(t)$	Probability of inferring IP change before time t
$F^{(k)}_{\text{Map}}(t)$	Probability of having the mapping before time $t \in I_k$

Table 6.1: Table of notations.

or allocated a new one. We model this with independent random variables drawn from a Bernoulli distribution: with probability p_{New} AP is assigned a new IP, and with probability $1-p_{\text{New}}$ it is re-assigned the same IP. We divide time into successive sub-intervals I_k, $k \geq 0$, of duration T, corresponding to the DHCP leases: $I_k = [kT, (k+1)T]$. Without loss of generality, we assume the duration of IP leases to be constant. Each sub-interval is aligned with a DHCP lease. Therefore, within each sub-interval AP's public IP address remains unchanged. For any time instant t, we denote by $\bar{t}$, the relative time within the corresponding sub-interval, that is $\bar{t} = t \mod T$.

Users connect to AP, remain connected for a certain time and then disconnect. While connected, users make requests, each of which is of one of the following types: LBS, authenticated, or standard. All modeling choices in this section follow well-established conventions [192] – e.g., Poisson processes are known to fit well users arrival and access to services – and are backed up by several public Wi-Fi hotspot workload analysis (e.g., [131]). We model users who arrive and connect to AP with a homogeneous Poisson process with intensity λ_{Arr}, thus the number $N_{\text{Arr}}(t)$ of users who arrive and connect to AP during any time interval of length t follows a Poisson distribution with parameter $\lambda_{\text{Arr}}t$:

$$\mathbf{P}\left[N_{\text{Arr}}(t) = n\right] = \frac{(\lambda_{\text{Arr}}t)^n}{n!} e^{-\lambda_{\text{Arr}}t} \ , \ n \geq 0.$$

We denote the time users stay connected to AP by T_{Dur}, which follows an exponential distribution with average $\frac{1}{\lambda_{\text{Dur}}}$. This means that the associated cumulative distribution function (cdf) and probability density function (pdf) are

$$f_{\text{Dur}}(t) = \lambda_{\text{Dur}} e^{-\lambda_{\text{Dur}}t} \quad \text{and} \quad F_{\text{Dur}}(t) = \mathbf{P}\left[T_{\text{Dur}} < t\right] = 1 - e^{-\lambda_{\text{Dur}}t} \ .$$

A noteworthy property of exponential distributions is *memorylessness*: the probability distribution of the time spent by a given user at a certain AP *since* a given time instant t, provided that the user is still connected at time t, is the same for all t. In other words, $\mathbf{P}\left[T_{\text{Dur}} > \delta t\right] = \mathbf{P}\left[T_{\text{Dur}} > t + \delta t \mid T_{\text{Dur}} > t\right]$, for all t and δt.

We assume the system to be stationary with respect to user connections and disconnections. Based on Little's law [192], the average number of connected users at any time instant t is therefore constant and given by: $N_{\text{Con}} = \lambda_{\text{Arr}}/\lambda_{\text{Dur}}$.

Users generate requests independently of each other. For each user, the three types of requests he makes are also independent: Standard and authenticated requests are modeled by independent homogeneous Poisson processes with intensity λ_{Std} and λ_{Auth}, respectively. In particular, the probability that at least one request of a type is made during an interval of length t is

$$P_{\text{Std}}(t) = 1 - e^{\lambda_{\text{Std}} t} \quad \text{and} \quad P_{\text{Auth}}(t) = 1 - e^{\lambda_{\text{Auth}} t} \ .$$

Another noteworthy property of Poisson processes is that the numbers of requests in two disjoint intervals are independent. We assume that each user makes a request when he connects to *AP*. For instance, e-mail or RSS clients (e.g., GoogleReader) usually automatically connect to a server when an Internet connection is available. We assume that only a proportion α_{LBS} of the users make LBS requests, and we model such requests by independent homogeneous Poisson processes with intensity λ_{LBS} for each user.

Figure 6.3 depicts the user arrivals, departures, standard and LBS request processes and illustrates the key notations and concepts introduced in this section.

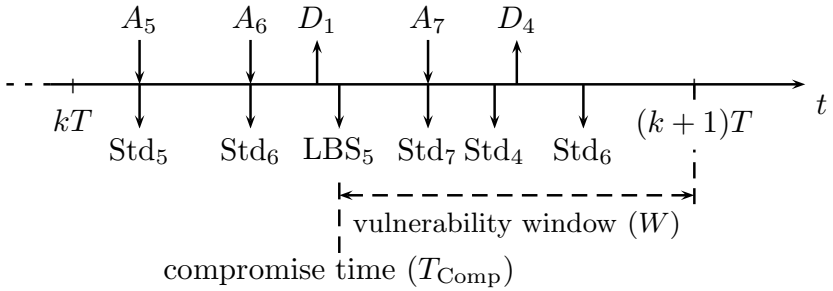


Figure 6.3: Threat due to a user making an LBS request. A_i and D_i represent User i's arrival and departure, respectively. Users 1 and 4 are already present at time kT. The time at which the first LBS request is made (LBS_5) is called the *compromise time* (T_{Comp}). From T_{Comp} on, any user who makes a standard request is a victim. Users already connected at T_{Comp} are victims if they make a standard request after T_{Comp}, e.g., User 4. Users who connect after T_{Comp} are, *de facto*, victims as users make a standard request when they connect, e.g., User 7.

6.4.2 Threat

We first focus on a single sub-interval and quantify the location-privacy threat, with respect to the number of users whose locations are disclosed to the adversary because of others. Specifically, we call a *victim* a user who makes a standard request at a time at which the adversary already knows the (IP, Location) mapping.

Quantifying the threat in a sub-interval If at least one user connected to AP uses an LBS at some time instant (thus revealing his current location), $\mathcal{A}$ obtains the (IP, Location) mapping based on which it can locate other users.

We define the *compromise time* T_{Comp} as the first time within the sub-interval, when a user connected to AP uses an LBS. If such an event does not occur, the compromise time is equal to T. At any time, there are on average N_{Con} users connected to AP, out of which $\alpha_{\text{LBS}} N_{\text{Con}}$ potentially make LBS queries. The aggregated process of LBS requests is a Poisson process with intensity $\Lambda_{\text{LBS}} = \alpha_{\text{LBS}} N_{\text{Con}} \lambda_{\text{LBS}}$. Therefore, the probability that at least one LBS request (from the aggregated process) is made before time $\bar{t}$ in the sub-interval is

$$F_{\text{Comp}}(\bar{t}) = \mathbf{P}\left[T_{\text{Comp}} < \bar{t}\right] = 1 - e^{-\Lambda_{\text{LBS}}\bar{t}} \ ,$$

and the expected compromise time is $\frac{1}{\Lambda_{\text{LBS}}}(1 - e^{-\Lambda_{\text{LBS}}T})$. We call f_{Comp} the corresponding probability density function. The time interval that spans from the compromise time to the end of the sub-interval is called the *vulnerability window* (see Figure 6.3) and the expected value W of its duration is

$$\mathbf{E}\left[W\right] = T - \frac{1 - e^{-\Lambda_{\text{LBS}}T}}{\Lambda_{\text{LBS}}}. \tag{6.1}$$

Figure 6.4 depicts the cumulative distribution function of the compromise time and its average value in an example setting. We observe that even with moderate AP popularity and LBS usage, the adversary obtains the mapping before the DHCP lease expires in 83% of the cases and he does so after 11 hours on average.

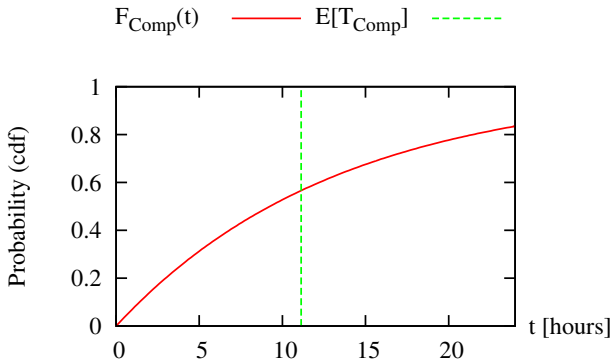


Figure 6.4: Cumulative distribution function of the compromise time T_{Comp} (expressed in hours). The parameters were set to $T = 24$ h, $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$ (i.e., average connection time of one hour and a half), $\lambda_{\text{LBS}} = 0.05$ req./h, and $\alpha_{\text{LBS}} = 0.2$.

In order to compute the number of victims, we distinguish between two groups of users: those who were already connected when the first LBS request was made, e.g., User 6 in Figure 6.3, and those who subsequently connected during the vulnerability window (and are, *de facto*, victims as they make a standard request when they connect), e.g., User 7. We call V_1 and V_2 the number of victims in each group.

There are N_{Con} users connected at the compromise time (recall that there are on average N_{Con} users in the system at any time). Whenever we compute an expected value involving the number of users connected, Wald's equation [192] allows us to consider that the system is composed of exactly N_{Con} users. Provided that an LBS request is made within the sub-interval, the number of victims at that time is the number V_1 of connected users who make

a subsequent standard request before leaving and before the end of the sub-interval. We compute the expected value of V_1 by applying the law of total probability, conditioning over both the compromise time (t) and the time spent in the system (u):

$$\begin{aligned}\mathbf{E}[V_1] &= N_{\text{Con}} \int_{t=0}^{T} \int_{u=0}^{\infty} f_{\text{Comp}}(t) f_{\text{Dur}}(u) P_{\text{Std}}(\min(u, T-t)) du dt \\ &= N_{\text{Con}} \frac{\Lambda_{\text{LBS}} \lambda_{\text{Std}}}{(\lambda_{\text{Std}} + \lambda_{\text{Dur}}) - \Lambda_{\text{LBS}}} \cdot \frac{1 - e^{-\Lambda_{\text{LBS}} T}}{\Lambda_{\text{LBS}}} - \frac{1 - e^{-(\lambda_{\text{Std}} + \lambda_{\text{Dur}})T}}{(\lambda_{\text{Std}} + \lambda_{\text{Dur}})}. \end{aligned} \tag{6.2}$$

We compute, on average, the number V_2 of users who connect to AP between the compromise time and the end of the sub-interval by applying the law of total probability, conditioning over the length of the vulnerability windows:

$$\mathbf{E}[V_2] = \mathbf{E}[N_{\text{Arr}}(W)] = \lambda_{\text{Arr}} \cdot \mathbf{E}[W] = \lambda_{\text{Arr}} \cdot (T - \frac{1 - e^{-\Lambda_{\text{LBS}} T}}{\Lambda_{\text{LBS}}}). \tag{6.3}$$

The average number of victims in a sub-interval is the expectation of the sum of the number of victims connected at the compromised time (V_1) and victims arriving within the vulnerability window (V_2), i.e., $\mathbf{E}[V_1] + \mathbf{E}[V_2]$. Naturally, this number has to be compared to the average number of users who have been connected at some point within the sub-interval: $V_{\text{total}} = N_{\text{Con}} + \lambda_{\text{Arr}} T$. It can be observed in Figure 6.5 that the proportion of victims $(\mathbf{E}[V_1] + \mathbf{E}[V_2])/V_{\text{total}}$ increases with T. This is because all users who connect during the vulnerability window are victims. As the probability of the adversary obtaining the mapping before time T increases with T, V_1/V_{total} first increases. However, because V_1 is upper-bounded by N_{Con} and V_{total} increases with T, V_1/V_{total} eventually tends to 0. In the end, when the DHCP lease expires, the location of more than half of the users is compromised.

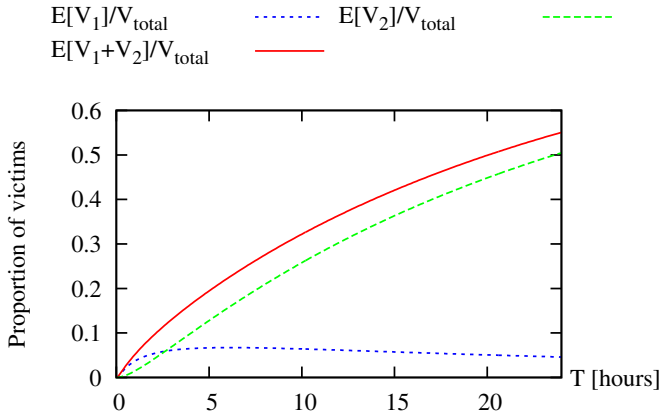


Figure 6.5: Proportion of victims within a sub-interval of length T, corresponding to a DHCP lease. The parameters were set to: $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1$ (i.e., average connection time of one hour), $\lambda_{\text{Std}} = 10$ req./h, $\lambda_{\text{LBS}} = 0.05$ req./h, and $\alpha_{\text{LBS}} = 0.2$. The dotted curve (resp. dashed) corresponds to the victims connected at (resp. arriving after) the compromise time. The solid curve represents the total proportion of victims.

Inferring IP change We consider two successive sub-intervals, without loss of generality I_0 and I_1, and we look at the linking probability F_{Link} that the adversary infers the IP change

from authenticated requests. This occurs if at least one user makes both an authenticated request at most ΔT time units ($\Delta T < T/2$) before the IP change and another authenticated request at most ΔT time units after the IP change.

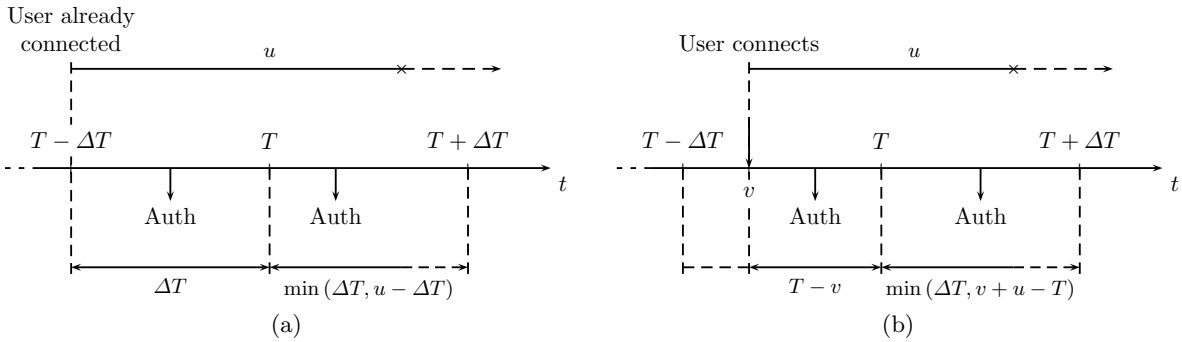


Figure 6.6: Timeline for the two groups of users from which the adversary can infer the IP change. A user remains connected for a random time u. (a) The user is already connected at time $T - \Delta T$. For the adversary to infer the IP change, u needs to be greater than ΔT and the user must make at least two authenticated requests: One during the time interval $[T - \Delta T, T]$ and another one during $[T, T + \min(\Delta T, u - \Delta T)]$. (b) The user connects at some time v during the time interval $[T - \Delta T, T]$. For the adversary to infer the IP change, u needs to be greater than $T - v$ (i.e., the user must still be connected at time T) and the user must make at least two authenticated requests: one during the time interval $[v, T]$ and another one during $[T, T + \min(\Delta T, v + u - T)]$.

Proceeding similarly as above, we compute the probability of inferring the IP change by distinguishing between two groups of users: those who were connected at time $T - \Delta T$ (see Figure 6.6a) and those who connected within $[T - \Delta T, T]$ (see Figure 6.6b). We denote by P_1 (resp. P_2) the probability that the adversary infers the IP change from the authenticated requests made by a user of the first group (resp. second group). First consider a user who was already connected at time $T - \Delta T$ (there are N_{Con} such users). In order to infer the IP change from the authenticated requests of such a user before time $t \in I_1$, the following conditions must be satisfied: (i) the user stays connected at least until time T, (ii) the user makes an authenticated request between the times $T - \Delta T$ and T, and (iii) the user makes an authenticated request, before time t, and before he leaves (if he leaves before time $T + \Delta T$ or until $T + \Delta T$ otherwise) (see Figure 6.6). We compute the probability that at least one user (among N_{Con}) satisfies the above conditions by applying the law of total probability, conditioning over the time spent in the system from time $T - \Delta T$:

$$P_1(t) = 1 - \left(1 - \int_{u=\Delta T}^{\infty} f_{\text{Dur}}(u) P_{\text{Auth}}(\Delta T) P_{\text{Auth}}(\min(\Delta T, u - \Delta T, t - T)) du\right)^{N_{\text{Con}}}. \quad (6.4)$$

Now consider the users who connect during the time interval $[T - \Delta T, T]$ (see Figure 6.6b). The number of such users follows the Poisson process $N_{\text{Arr}}(\Delta T)$. By applying the law of total probability, conditioning over the number of such users, their arrival times (independent of each other and uniformly distributed within $[T - \Delta T, T]$), and their departure times, we compute the probability that at least one of the newcomers satisfies the above conditions

$$P_2(t) = \sum_{n=1}^{\infty} \mathbf{P}\left[N_{\text{Arr}}(\Delta T) = n\right] \cdot (1 - (1 - P(t))^n) = 1 - e^{-\lambda_{\text{Arr}} \Delta T \cdot P(t)}, \quad (6.5)$$

where

$$P(t) \quad = \quad \int_{v=T-\Delta T}^{T} \Delta T^{-1} \int_{u=T-v}^{\infty} f_{\mathrm{Dur}}(u) P_{\mathrm{Auth}}(T-v) P_{\mathrm{Auth}}\left(\min\left(\Delta T, v+u-T, t-T\right)\right) du dv$$

Due to space constraints, we do not include the closed-form expressions of P_1 and P_2. These can be easily computed because all integrals are of the form $\int_{u=0}^{t} u e^{-a\,u} du$, which is equal to $(1-e^{-a\,t})/a$.

In conclusion, the probability that the adversary infers the IP change before time $t > T$, referred to as the *linking probability*, is given by:

$$F_{\mathrm{Link}}(t) = 1 - (1 - P_1(t))(1 - P_2(t)) \ .$$

Note that the above equations can easily be generalized to any sub-interval I_k, $k \geq 1$, by replacing $t-T$ (the relative time in I_1) by $\bar{t} = t \mod T$ (the relative time in any sub-interval).

The linking probability can be thought of as depending both on t and ΔT. Figure 6.7 depicts the linking probability at time $T + \Delta T$ as a function of ΔT. It can be observed that this probability rapidly converges to 1. The probability P_1 (resp. P_2) of inferring the IP change from the users already connected at time $T - \Delta T$ (resp. from the users who connect after time $T - \Delta T$ and before the end of the sub-interval) first increases with ΔT: the probability of generating authenticated requests increases with the length of the interval. For large values of ΔT however (typically higher than the average connection time), P_1 decreases. This is because users connected at time $T - \Delta T$ are not likely to still be connected at time T when ΔT is large (compared to the expected connection duration $1/\lambda_{\mathrm{Dur}}$). Note that the fact that linking probability increases with ΔT is balanced by the decreased confidence of the adversary. This is because the probability that a user makes two authenticated requests from two distinct access points in the time interval $[T-\Delta T, T+\Delta T]$ (moving from one to the other) increases with ΔT.

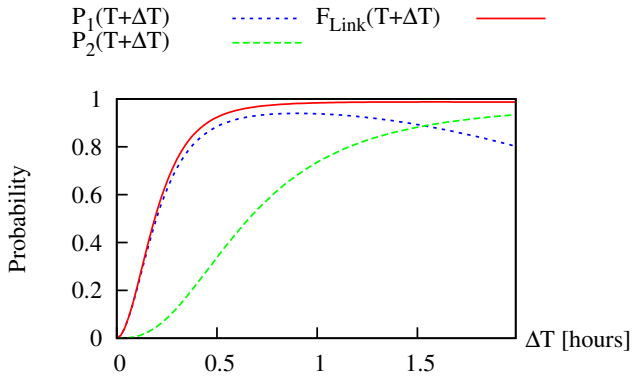


Figure 6.7: Linking probability at time $T+\Delta T$ as a function of ΔT. The parameters were set to $\lambda_{\mathrm{Arr}} = 5$ users/h, $\lambda_{\mathrm{Dur}} = 1/1.5$, $\lambda_{\mathrm{Std}} = 10$ req./h, $\lambda_{\mathrm{LBS}} = 0.05$ req./h, $\lambda_{\mathrm{Auth}} = 2$ req./h, and $\alpha_{\mathrm{LBS}} = 0.2$. The probability of the adversary inferring the IP change based on users connected at time $T - \Delta T$ i.e., P_1, and based on users connecting in the time interval $[T - \Delta T, T]$, i.e., P_2, are represented by the dotted and dashed curves, respectively. The solid curve represents the total probability of inferring the IP change i.e., F_{Link}.

Figure 6.8 depicts the linking probability as a function of t. It remains constant for $t \geq T + \Delta T$ because only authenticated requests made in the time interval $[T - \Delta T, T + \Delta T]$

are taken into account to infer the IP change. Note that with a value of ΔT as small as 5 minutes, which provides high confidence, the adversary can still infer the IP change with a probability of 43%.

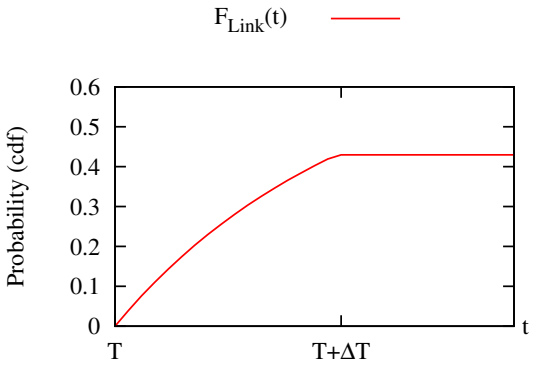


Figure 6.8: Probability of inferring the IP change before time $t > T$, i.e., $F_{\text{Link}}(t)$. The parameters were set to: $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$, $\lambda_{\text{Std}} = 10$ req./h, $\lambda_{\text{LBS}} = 0.05$ req./h, $\lambda_{\text{Auth}} = 2$ req./h, $\Delta T = 5$ minutes, and $\alpha_{\text{LBS}} = 0.2$. The dotted curve represents the probability that at least one user connected at time $T - \Delta T$ allows the adversary to infer the IP change before time t, i.e., $P_1(t)$. The dashed curve represents the same probability for users connecting in the interval $[T - \Delta T, T]$, i.e., $P_2(t)$.

Quantifying the threat over multiple sub-intervals When the adversary infers the IP changes, the probability $F^{(k)}_{\text{Map}}(t)$ that the adversary knows the (IP, Location) mapping at time $t \in I_k$, $k \geq 1$ is

$$F^{(k)}_{\text{Map}}(\bar{t}) = F_{\text{Comp}}(\bar{t}) + (1 - F_{\text{Comp}}(\bar{t})) \cdot F^{(k-1)}_{\text{Map}}(T) \cdot ((1 - p_{\text{New}}) + p_{\text{New}} F_{\text{Link}}(\bar{t})) \tag{6.6}$$

with initial condition $F^{(0)}_{\text{Map}}(\bar{t}) = F_{\text{Comp}}(\bar{t})$. Note that the assumption $\Delta T < T/2$ is required here. Indeed, this technical restriction ensures that the time interval $[kT - \Delta T, kT + \Delta T]$ (used by the adversary for the linking), does not overlap with the time interval $[(k-1)T - \Delta T, (k-1)T + \Delta T]$. Essentially, this makes the two intervals disjoint and therefore also independent with respect to authenticated requests, which allows us to multiply the corresponding probabilities. From Equation (6.6), it can be seen that $F^{(k)}_{\text{Map}}(T)$ obeys the following recursive equation:

$$F^{(k)}_{\text{Map}}(T) = a + bF^{(k-1)}_{\text{Map}}(T)$$

where $a = F_{\text{Comp}}(T)$ and $b = (1 - F_{\text{Comp}}(T)) \cdot ((1 - p_{\text{New}}) + p_{\text{New}} F_{\text{Link}}(T))$. This recursive equation can easily be seen to have as a solution $a(1 - b^{k+1})/(1 - b)$. As $b < 1$, $F^{(k)}_{\text{Map}}(T)$ converges to a finite value, i.e., $a/(1 - b)$.

The number of victims in the sub-interval I_k can be computed by replacing the density f_{Comp} in Equations (6.2) and (6.3) with the density of $F^{(k)}_{\text{Map}}$. The probability that the adversary has the mapping (IP, Location) at time t in sub-interval I_k, i.e., $F^{(k)}_{\text{Map}}$ is illustrated in Figure 6.9. It can be observed that the mapping probability increases over time and, after the convergence, the adversary successfully obtains the mapping before the DHCP lease expires in 79% of the cases and before the half-lease in 60% of the cases.

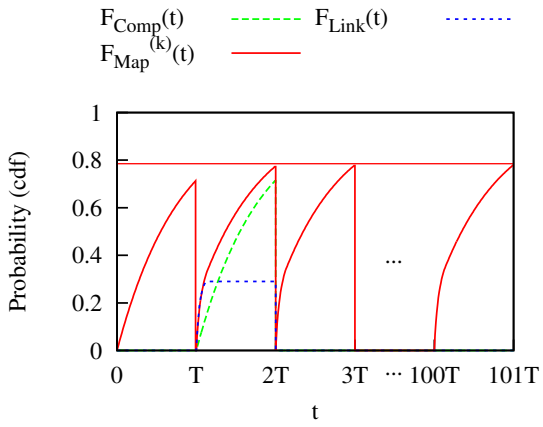


Figure 6.9: Probability of obtaining the (IP, Location) mapping over several sub-intervals. The solid curve represents the probability of obtaining the mapping before time t. The dashed curve represents the probability of obtaining the mapping from an LBS request. The dotted curve represents the probability of inferring the IP change. The used parameters are $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$, $\lambda_{\text{Std}} = 10$ req./h, $\lambda_{\text{LBS}} = 0.035$ req./h, $\lambda_{\text{Auth}} = 0.2$ req./h, $T = 24$ h, $\Delta T = 3$ h, $\alpha_{\text{LBS}} = 0.1$, and $p_{\text{New}} = 1$. To highlight the respective contributions of the linking and compromise probabilities, some values differ from our previous setting (e.g., ΔT). In the first sub-interval, the linking probability is zero and the probability of having the mapping is the compromise probability. In subsequent sub-intervals, this probability $F^{(k)}_{\text{Map}}(t)$ increases due to the potential inference of IP changes: it becomes a combination of $F_{\text{Link}}(\bar{t})$ and $F_{\text{Comp}}(\bar{t})$ (and the probability of having the mapping by the end of the preceding sub-interval).

6.5 Experimental Results

In this section, we complement our theoretical analysis with experimental results based on traces from a network of deployed Wi-Fi access points.

6.5.1 Dataset

Our dataset consists of daily user Wi-Fi *session traces*, *traffic traces* and *DNS traces* for a period of 23 days in June 2012. We aggregate the data of two APs located very close to each other ($\sim$ 15 meters), to emulate the scenario of a single popular hotspot and to avoid side effects of micro-mobility, i.e., devices frequently changing the AP they are connected to.

Session traces contain information related to users connecting and disconnecting from the APs, obtained from the RADIUS logs that the network uses for authentication, authorization, and accounting management [191]. There are three types of RADIUS events: (i) `start` – a user is successfully authenticated and the device is assigned an IP denoting the beginning of a session; (ii) `update` – a user connected to the AP periodically issues a status message; and (iii) `stop` – a user disconnects denoting the end of the session. Each entry in the log contains a timestamp, the device's anonymized MAC address, the assigned IP address, the ID of the AP the device is connected to, and an event type.

Traffic traces are obtained from the logs at a border router connecting the network to the Internet. Each entry in the log contains a timestamp, the source IP, and the destination (including the IP address and port). The mapping between a user's assigned IP address and

her MAC address allows us to correlate traffic with user session traces.

DNS traces are obtained from the local DNS servers and each entry in the log contains a timestamp, the source IP and the requested complete host name. Based on the source IP addresses, timestamps and requested resources, we correlate the DNS with the traffic traces.

The average number of users connected to the AP over a day (averaged over 23 days) is shown in Figure 6.10. We observe that users typically begin arriving around 7:AM. The number of connected users peaks around 6:PM (136 on average). In total, 4,302 users have connected during 23 days.

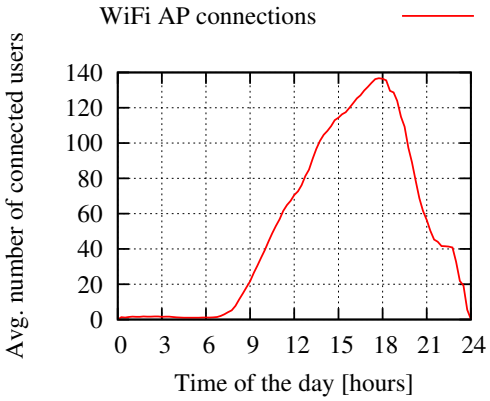


Figure 6.10: Average number of users connected to the AP over a day (averaged over 23 days).

We filtered traffic to a number of Google services (including e-mail, search, LBS, analytics, advertising) and classified each request (i.e., standard, LBS, or authenticated) based on the destination IP, port and DNS requests. We sanitized the traffic data beforehand by appropriately grouping traffic traces into user-service sessions. To do so, we correlated traffic and DNS requests. This was made possible by the fact that DNS replies for Google services are cached for a relatively short time (i.e., TTL of 300 seconds), and therefore a traffic request is very often preceded by a DNS request. Consequently, a request accounts for a user-service interaction, regardless of how much traffic the interaction generates. The monitored services and their classification is presented in Table 6.2. Entries of the type *service.** refer to all the top-level-domains observed in the traces (e.g., *.com*, *.fr*, *.ca*). Entry **.gmail.com* includes *imap.*, *smtp.*, *pop.*, *www.* and *m.* and *doubleclick.** includes *.de* and *.com*. The *m.* prefix stands for mobile services.

Request Type	**Services**
Standard	www.google.*, www.googlescholar.com, www.youtube.com, www.google-analytics.com, doubleclick.*, m.doubleclick.*, pagead2.googlesyndication.com
LBS	maps.google.*, earth.google.com
Authenticated	calendar.google.com, *.gmail.com, plus.google.com

Table 6.2: Monitored services.

Traffic to the monitored services (in terms of the number of user-service sessions) constitutes about 17% of the total traffic generated at the AP and 81.3% of users who connected have accessed at least one of the services, which shows the tremendous popularity of Google services. The average numbers of standard, authenticated and LBS requests (i.e., user-service interactions) during a day to the monitored services are depicted in Figure 6.11. It can be observed that standard requests are prevalent, followed by authenticated requests. The moderate usage of LBS services can be explained with the location of the APs: most of the users visit this area almost on a daily-basis, therefore the need for location-based information is expected to be low. In our dataset, 9.5% of users generate LBS requests.

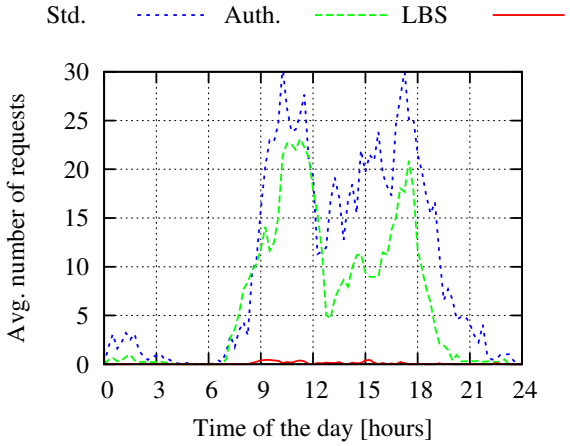


Figure 6.11: Average number of standard, authenticated, and LBS requests to the monitored services over a day (averaged over 23 days).

6.5.2 Results

First, we measure the compromise time and the proportion of victims based on the traces from our dataset. We compare the averaged experimental results with those from our theoretical analysis and show them in Figure 6.12. For the theoretical analysis, we use our framework with the parameters extracted from the real traces: $\lambda_{\text{Arr}} = 14.54$ users/h and an average connection time of 2.17 hours ($\lambda_{\text{Dur}} = 1/2.17$), obtained from the session traces; and traffic rates of $\lambda_{\text{Std}} = 28.3$ req./h, $\lambda_{\text{Auth}} = 14.6$ req./h and $\lambda_{\text{LBS}} = 0.16$ req./h (with $\alpha_{\text{LBS}} = 0.095$), obtained from the traffic traces. Because the theoretical model assumes a homogeneous user arrival rate, we compute the expected proportion of victims and compromise time as if the arrival process spanned from 7:30:AM – the time at which a significant number of users start connecting to the AP in our traces – to 7:PM. It can be observed that although the model does not capture the time-of-the-day effects of the user arrival and traffic processes, the theoretical and experimental expected proportions of victims match when considering the entire day.

We observe that around 8:AM (7:42:AM estimated with our theoretical analysis and 8:25:AM with our experimental results), only 1 hour after users typically start connecting to the AP, users' location privacy is compromised. By the end of the day, about 73% of the users who connected through the AP were compromised, out of which 90.5% did not make any LBS request ($\alpha_{\text{LBS}} = 0.095$). Note however, that with respect to the number of users who use Google services the proportion of victims actually corresponds to 90%. Thus, the result

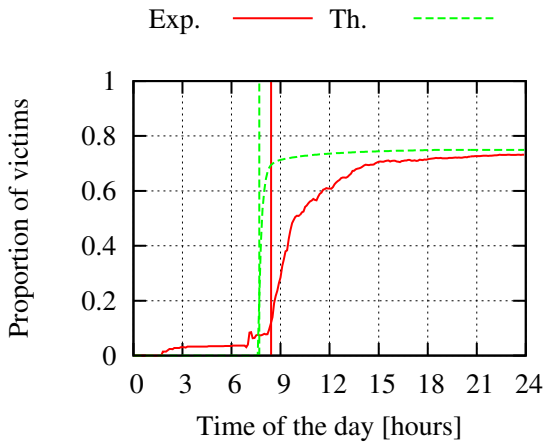


Figure 6.12: Expected proportion of victims. Vertical lines represent average compromise times: theoretical T_{Comp} = 7:42:AM and experimental T_{Comp} = 8:25:AM.

shows that Google is able to learn the location of 90% of its users who connect from the AP. This example shows the large extent of the threat users are exposed to, as it is expected to be even worse for more popular hotspots, e.g., airports. Note however, that Google constitutes a rather powerful potential adversary because it receives very much of user traffic.

Once the adversary obtains the (IP, Location) mapping, it can maintain it over time by relying on authenticated requests to infer the IP changes upon DHCP lease expirations, as discussed in Section 6.4. Using traces from our dataset, we compute the probability of the adversary inferring the IP change for different renewal times during a day, considering the authenticated requests made at most ΔT minutes before and after the IP is changed. We consider three different values, $\Delta T = 1$, $\Delta T = 5$ and $\Delta T = 10$ minutes, and show the results in Figure 6.13. We assume that each time the DHCP lease expires the AP is assigned a new IP address. Even with the smallest inference time window of 1 minute, the adversary can infer the IP change with the probability 1.0 between 2:PM and 5:PM. With higher values of ΔT the time during which the adversary can infer with probability 1.0 is even longer, i.e., from 11:AM to 7:PM with $\Delta T = 10$. However, the adversary's confidence decreases with larger ΔT. During the periods where there is less traffic (e.g., from 11:PM to 6:AM), the probability of the adversary inferring the mapping is smaller (less than 0.2) in all the cases. Between 5:AM and 6:AM, the adversary cannot infer the IP change, as there is no traffic during this time.

Consequently, the IP renewal time affects the adversary's success at maintaining the (IP, Location) mapping over time. To confirm this conjecture, we plot the probability of the adversary having the mapping over a period of three days, considering different IP renewal times (Figure 6.14). We plot two representative scenarios: (i) IP renewal time at 5:AM – when the probability of inferring the IP change is equal to zero (Figure 6.14a) and (ii) IP renewal time at 4:PM – when the probability of inferring the IP change is equal to 1.0 (Figure 6.14b). In both cases we set $\Delta T = 5$ minutes. As discussed in Section 6.4 and represented in Figure 6.9, the probability of adversary having the mapping (F_{Map}) is a combination of the probabilities that the compromise happens due to LBS usage (F_{Comp}) and the probability of having the mapping and inferring the IP change upon DHCP lease expiration (F_{Link}). Therefore, in both cases, we observe that the probability of obtaining the mapping for the first time corresponds to the probability of users generating LBS requests and revealing the mapping, i.e., F_{Comp}.

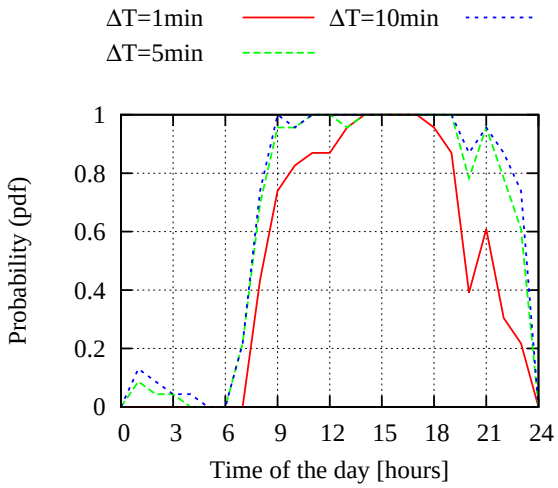


Figure 6.13: Linking probability (i.e., probability of inferring the IP change) as a function of the renewal time for different inference time window lengths (ΔT).

Once the adversary obtains the mapping, we notice the contrast in how successfully it can maintain it over time, due to different inference probabilities. Results in Figure 6.14a show that when the adversary cannot infer the IP change (i.e., $F_{\text{Link}} = 0$), its success over time depends solely on users' LBS requests, i.e., the curves F_{Map} and F_{Comp} overlap. Thus, there are periods of time during which the adversary does not have the mapping. On the contrary, in Figure 6.14b, when the IP renewal happens at 4:PM and the adversary can always infer the IP change (i.e., $F_{\text{Link}} = 1$), we observe that once the adversary learns the mapping it can successfully maintain it over time with probability 1.0.

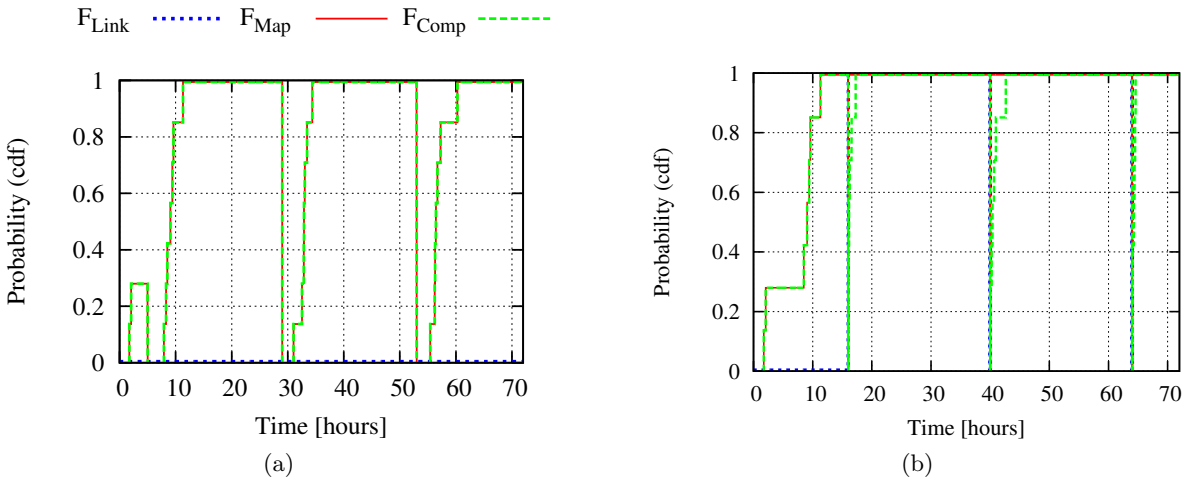


Figure 6.14: Probability of adversary having the (IP,Location) mapping (F_{Map}), depending on the IP renewal time: (a) at 5:AM and (b) at 4:PM. F_{Map} is a combination of the probabilities that the compromise happens due to LBS usage (F_{Comp}) and the probability of having the mapping and inferring the IP change upon DHCP lease expiration (F_{Link}).

To further confirm the importance of the IP renewal time and its affect on the adversary's success, we plot the cumulative number of victims compromised at the AP during three

weeks, depending on the IP renewal time (Figure 6.15). We set $\Delta T = 5$ minutes and based on the previous findings, we consider the renewal times at 5:AM, 4:PM and 8:PM, when the adversary is expected to be least successful, most successful and moderately successful, respectively. Indeed, from the results in Figure 6.15, we confirm that the highest number of users (3545 out of 4302 total number of users, which corresponds to virtually all users who access Google services) is compromised when the IP renewal happens at 4:PM, followed by 8:PM (3149 victims). The adversary is least successful when the IP renewal is at 5:AM (compromising 2879 users in total). These results confirm previous findings.

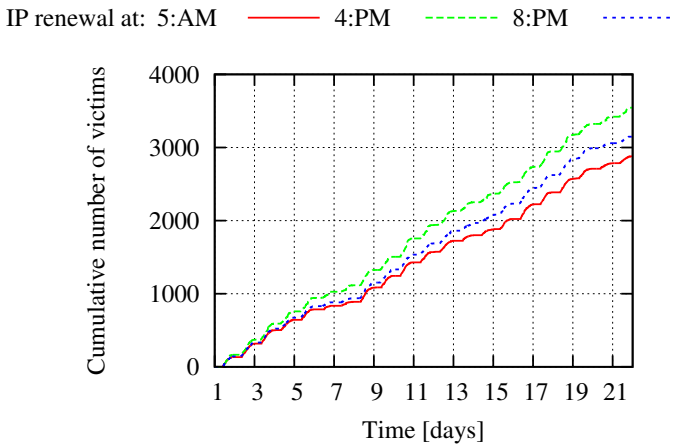


Figure 6.15: Cumulative number of victims at the AP during the whole experiment, for three different IP renewal times.

6.6 Countermeasures

Cryptographic primitives are efficient at protecting users' privacy, but because of the way networking protocols operate, they might not be sufficient, in particular, when the private information is the source IP address.

Hiding users' actual source IPs from the destination (i.e., the adversary) naturally comes to mind as a straightforward countermeasure against the considered threat. This can be done in several ways. In relay-based anonymous communications, a user's traffic is forwarded from the source to the destination by several relay nodes, in such a way that the destination cannot know the user's source IP. Examples of such networks include Tor [113], mix networks [101, 109], or simple HTTP proxies [196]. With Virtual Private Networks (VPNs) [132], the user is assigned an IP address that belongs to a remote network (e.g., a corporate network [58] or commercial/public VPN [81, 79]). To the adversary, the user's requests appear to originate from within the remote network, whose location is different from that of the user. Unfortunately, such techniques are not widely adopted, especially in the case of mobile communications [4]. In addition, several techniques exist to identify the source IP address of a client, even behind a NAT or a proxy, e.g., by using a Java applet [176, 181]. Finally some techniques allow to consistently track hosts behind NATs or proxies [235].

Alternatively, these countermeasures can be implemented by ISPs, for instance, by deploying a country-wide NAT that aggregates traffic from all hosts connected to the ISP at several gateways (e.g., Telefonica [82], Swisscom Hotspots) or by IP Mixing [186]. This also

applies to operators of AP networks (e.g., Starbucks, AT&T Wi-Fi). However, they may not have incentives to implement such solutions.

Another approach to thwart the threat consists in degrading the knowledge of the adversary, by reducing the accuracy of the reported location and by increasing the uncertainty about the AP's location. Examples of location privacy enhancing technologies (PETs) reducing the adversary's accuracy include spatial cloaking [91, 140] and adding noise to reported locations [86]. To increase the adversary's uncertainty, [158] proposes to inject "dummy" requests, i.e., not related to the user's location. It is not easy for users to implement these PETs, because some geolocation requests are implemented in the operating system, that can be controlled by the adversary (e.g., Google Android). Moreover, when these techniques are implemented in a non-coordinated fashion, the adversary might still be able to infer the actual location by filtering out requests that stand out from the bulk (increasing its certainty) and averaging the remaining requests (increasing its accuracy). Better results might be achieved by having the AP operators implement the location-privacy preserving mechanisms, but they might lack incentives to do so.

Finally, as highlighted by our analysis, various other countermeasures can be implemented by the ISP or the AP's owner: reduce the DHCP lease, always allocate a new IP, trigger the IP change when the traffic is low (e.g., at 5:AM as suggested by our experimental results) or purposely impose silent periods around the renewal time (reducing the chances that the adversary infers the IP change from authenticated requests). Unfortunately, all these techniques have a negative effect on the quality of service and impose a significant overhead in network management. Thus, they are unlikely to be deployed in practice. Besides technical countermeasures, we envision a "Do-Not-Geolocalize" initiative, similar to "Do-Not-Track" [120], letting users to opt-out of being localized.

6.7 Discussion

Scale and implications of the threat By maintaining (IP, Location) mappings in the manner we have described, an adversary can build an IP-location system with which he can obtain (at least) sporadic user locations. For an online service provider whose goal is to profit from delivering location-targeted information, it might be sufficient to learn only current user locations at the time users access services.

However, we can envision a different type of adversary, whose goal is to mount more powerful attacks on user privacy. In fact, once the adversary has access to sporadic user-location information, he is able to reconstruct entire trajectories, produce patterns of user movement habits, or infer other information about the user, e.g., users' real identities, interests and activities. For example, in [199] it is shown how an adversary that observes each user's sporadic locations (that could be noisy and anonymized) can de-anonymize the users, compute the probability that a given user is at a given location at a given time, and can construct a full trajectory of each user. Golle and Partridge [133], Beresford and Stajano [97], Hoh et al. [144], Krumm [162], and Freudiger et al. [124] use different techniques to show that users can be identified by inferring where they spend most of their time (notably their home and workplace). In these cases, the location-privacy threat we identified serves as a building block that enables other, more powerful attacks.

In this chapter, we focus on how an adversary can obtain the sporadic user-location information that is needed for commercial needs of service providers. Other attacks that are

enabled by this location-privacy threat are beyond the scope of this work and are largely addressed by the research community, as previously discussed. However, our work provides a framework for quantifying sporadic location exposure upon which the community can build.

Evolution of the threat with IPv6 The adoption of IPv6 is increasing. With IPv6, each host has a public IP, composed of a *prefix* (leftmost 64 bits), shared with other hosts in the same network, and a unique *host part* (rightmost 64 bits). Sharing a prefix is analogous to sharing a public IPv4 address behind a NAT: (IPv4, Location) mappings correspond to (Prefix, Location) mappings. Because IPv6 prefixes are intended to be less dynamic than IPv4 addresses, the threat is expected to be amplified.

Business opportunities Beyond threatening the location-privacy of users, the (IP, Location) mapping technique presented in this paper can be used as a novel IP-location solution potentially improving on existing solutions [183, 228]. Online service providers, such as Google and Microsoft, are in a position to build and monetize this service by simply utilizing user traffic they receive. Additional advantages of this approach are that it does not require a dedicated infrastructure or network measurements. Such a system can be used on its own, or as a complementary approach to one of the existing ones. Because ISPs control the IP address assignments and can prevent service providers from building the mapping (using the aforementioned countermeasure) they can make a profit by selling IP locations to service providers (e.g., Verizon in the US [51]) – some ISPs sell geographic information on the topology of their networks [176] – or by selling privacy-protection services to users.

Legal and policy aspects Because the threat presented in this papers is based only on a passive analysis of the received traffic, it does not raise addition legal or policy issues compared to what Web services already do, i.e., inferring information from IPs and mining user traffic to improve the quality and relevance of the offered services.

6.8 Summary

In this chapter we have presented a practical threat, effectively demonstrating that the location privacy of users connecting to access points can be (unintentionally) compromised by others. The scale of the threat is significant because it simply leverages on the way most networks are designed (i.e., using NAT). When successful, the service provider can locate users within a few hundreds of meters, i.e., more accurately than existing IP-location databases. Because such neighborhood or street-level accuracy of IP-location services is required for commercial needs (e.g., for advertising nearby local businesses) service providers have tremendous incentives to improve existing IP-location services and they could rely on the described threat to do so. This approach would be particularly successful for major service providers that receive much of users' traffic (e.g., Google, Microsoft, Apple). Our theoretical analysis provides a framework that enables us to quantify the threat for any access-point setting and to identify the key parameters and their impact on the service providers' success. The framework serves as a light-weight alternative to an extensive traffic analysis to estimate the threat. We experimentally investigate the state in practice, by analyzing real traces of users accessing Google services, collected from deployed Wi-Fi access points. We observe the large scale of the threat even with a modest use of LBS services. We survey possible countermeasures and

find that adequate ones can be used to protect individual user's location privacy. However, to completely thwart the threat, the countermeasures need to be widely deployed.

Publication: [222]

Conclusion

In this thesis, we have focused on security, privacy and the economic issues that stem from the fundamental element of the Web – online advertising. Online ad revenue, generated using the online advertising business model, is the root cause of the issues we study. Fraudsters have incentives to engage in ad fraud schemes in order to divert part of the ad revenue for themselves. Given that most of online services and applications are fueled by the online ad revenue, meddling with the online advertising business model can have serious consequences. Therefore, the stakeholders (e.g., ad networks) involved have incentives to protect the ad revenue and to invest in securing online advertising. To maximize the ad revenue, stakeholders have incentives to deploy techniques to track and profile users' online behavior, in order to customize ads to individual users' interests. These practices are often at odds with users' online privacy. Consequently, some users adopt ad-avoidance tools that block the download and display of ads and partially thwart online tracking. By doing so, however, users also unwittingly deprive service providers from ad revenue and undermine the online advertising business model. This pushes online service providers to look for alternative ways to monetize online content. As a result, online content and services might not be available free of charge for much longer.

In Part I, we have provided a better understanding of the vulnerabilities of online advertising systems, the attacks and possible countermeasures. In Chapter 1, we identify a novel type of ad fraud, based on inflight modification of ad traffic. We identify the attacks and the underlying techniques that allow for this type of ad fraud and explain how fraudsters can generate money from it. We provide a proof-of-concept implementation on Wi-Fi routers to demonstrate that the attacks can run successfully and transparently even on such resource-constrained devices. We propose a collaborative approach for securing online advertising against inflight ad traffic modification, ensuring the authenticity and integrity of Web content and advertisements. This countermeasure relies on valid certificates of ad networks, because websites typically do not implement certificate-based authentication properly, as we show in Chapter 2. We come to this conclusion by studying the deployment of certificate-based authentication on the top one million most popular websites. In most cases, authentication failures are due to domain mismatches between certificates and websites. We study the economic, legal and social aspects of this problem, and we show how the current economic model leads to distribution of cheap certificates for cheap security. We suggest a multidisciplinary approach for improving certificate-based authentication on the Web.

In Part II, we have studied the economic implications of threats to the online advertising business model and certain possible countermeasures. We use game theory to model strategic behavior of the involved entities and to analyze their mutually dependent actions. In Chapter 3, we study the consequences of ISPs becoming strategic participants in online advertising: either cooperating with ad networks by providing users' private information to achieve bet-

ter ad targeting in exchange for a share of the ad revenue, or by diverting a part of the ad revenue with inflight modification of ad traffic. Our work determines the conditions under which different behavior will occur: The outcome depends mostly on the value of the users' private information ISPs can provide to ad networks and the remuneration they require. If the information improves ad targeting significantly and ISPs do not require a high share of revenue in return, ISPs and ad networks will collaborate; otherwise, ISPs will divert a small part of the ad revenue for themselves or they will prompt deployment of countermeasures by ad networks. One positive side-effect of the countermeasures is improved Web security. In Chapter 4, we study strategic behavior of ad networks and ISPs when facing botnet ad fraud. We identify conditions under which ad networks are likely to solve the problem of botnet ad fraud by themselves and those under which they will subsidize the ISPs to achieve this goal. Our analysis shows that the optimal strategy is influenced mostly by the number of infected devices, the efficiency of the botnet detection and the ad revenue loss botnets cause. Cooperation between ad networks and ISPs is a desirable outcome that would benefit users (i.e., ISPs help maintain the security of users' devices), ad networks (i.e., protected ad revenue) and ISPs (i.e., bots removed from the ISPs' networks). In Chapter 5, we study the economic ramifications of ad-avoidance tools on the monetization of online content. We develop a game-theoretic framework for content providers to weigh their options to mitigate the consequences of ad-avoidance. We propose that websites should treat users individually and strategically apply ad-financed or fee-financed monetization strategy. We show that such a strategic approach yields higher revenue and respects users' preferences better than deploying one strategy across all users. We observe that understanding users' aversion to ads and preference for content is of crucial importance for publishers in order to make a well-informed decision. We expect that publishers will adopt alternative monetization strategies to online advertising, as the trend towards blocking ads is likely to grow given users' increasing resistance to online tracking.

In Part III, we have focused on privacy issues stemming from online advertising. To match ads to users' interests, stakeholders implement a number of techniques aimed at learning users' private information. Users' location information is of a particular interest because much additional information can be inferred from it (e.g., users' interests and activities) and because location-targeted ads are very effective. In Chapter 6, we identify a novel threat to users' location-privacy that enables service providers to geolocate users who connect through a shared access point (e.g., a hotspot or a home router), and they can do so with high accuracy (within a few hundreds of meters). The peculiarity of the threat is that users' location privacy is unintentionally compromised by other users who connect through the same access point and whose location is known to the service providers. The underlying problem is inherent in the way networks operate, notably due to Network Address Translation (NAT), thus the threat is prevalent. We propose an analytical framework that quantifies the potential privacy threat, and we experimentally assess the state in practice based on users' traffic to Google services, collected from deployed hotspots. We observed the large scale of the threat: Even at a moderately visited hotspot, Google could geolocate almost all of the users who access its services. Given the lack of efficient large-scale countermeasures, this threat to users' location privacy is very concerning.

Future Work

The results of this thesis shed more light on the security, privacy and economic issues arising from online advertising. Our work indicates that the many entities involved, their (often at odds) incentives and the intertwined effects of their interactions, result in a complex ecosystem with challenges to be further addressed. We suggest several research directions that can be pursued.

Presently, we are witnessing a proliferation of online tracking and profiling techniques as stakeholders continuously increase their efforts in obtaining users' private information. The later use of such private information to match ads to users' interest is expected to maximize ad revenue. However, users are mostly not aware of when and which information about them is collected, who has access to this data and how it is used. Until better regulation or industry practices are put into place, there is a need to design and implement tools that would empower users to be in better control of their private information. Also, efforts are needed to make it easier and more transparent for users to understand and control which information is shared for advertising purposes and which is for obtaining a relevant service. The first step towards this goal is to survey the existing tracking techniques, in order to observe the ways in which private information is collected, and to learn who has access to this data and how it is used. This can be done, for example, by conducting a large-scale investigation across the Web on how prevalent is behavioral advertising, on which private information is used for ad targeting and on what is the potential privacy leakage due to behavioral ads being observed by a third party. The ultimate goal would be to design a privacy-preserving ad system that satisfies both users' privacy preferences and stakeholders' expectancies in terms of ad revenue.

Due to the lack of other means to prevent online tracking (e.g., Do-Not-Track [106]), users deploy ad-avoidance tools that, in addition to (partially) thwarting tracking, also prevent the download and display of ads, which hurts the revenue of websites and deprives users of potentially relevant (non-intrusive) ads. Therefore, the existing technology does not allow for proper user differentiation: both privacy-aware and ad-adverse users deploy ad-avoidance tools. The adoption of a mechanism, e.g., Do-Not-Track, that enables users to signal their privacy preferences, would enable a proper classification that allows websites to take appropriate actions, e.g., to display non-behavioral, less obtrusive ads to privacy-aware users, which might lead to users being more acceptable to viewing ads. Our game-theoretic framework, presented in Chapter 5, can be extended to account for the presence of mechanisms such as Do-Not-Track and used to study their economic implications for the content monetization. This could bring a much needed understanding about the effects of the mechanisms such as Do-Not-Track on users' acceptance of ads and perhaps weaken the industry's resistance to empowering users to opt-out from being tracked online.

In Part II, we applied our game-theoretic frameworks to the estimated values of the key parameters (e.g., ad revenue generated at different websites) as we did not have access to the real data that is (currently) available only to the stakeholders. Obtaining and including the real data would provide considerable insight on our theoretical results. It would foster further understanding and modeling of economic implications of the strategic behavior in online advertising. Consequently, this would lead to a better prediction of the likely outcomes of such interactions and their effects on the Web.

Experts predict that (mobile) advertising for local businesses is a (still untapped) big source of revenue, especially due to the pervasiveness of modern mobile devices and users being online while on-the-go. To achieve such a level of targeting, ad networks need to know users'

locations with high accuracy (i.e., neighborhood or street level), thus they have increasing incentives to deploy techniques to obtain this knowledge. We identify one such possible technique in Chapter 6: it enables service providers to locate a user based on his IP address. For this purpose, the service provider builds and maintains (IP, Location) mappings of access points users connect through. Though learning users' locations with high accuracy might enable relevant (location-based) services, it also presents a threat to users' location-privacy. Promising ways to further study this problem are to focus on the following aspects: (i) the accuracy of this novel IP-location technique; (ii) the refinement of the analytical model we have provided for quantifying the location-privacy threat, for instance by modeling users' arrivals with an inhomogeneous Poisson process to capture time-of-the-day effects; (iii) the adversary's ability to maintain (IP, Location) records over time, i.e., inference about IP changes, influence of auxiliary information (e.g., users' persistent connections, fingerprinting users' connections) and the trade-off between the probability of inferring the IP change and the adversary's confidence; (iv) the adversary's ability to track users as they move and connect to different access points over time; and (v) the design of an efficient countermeasure against this threat.

In general, together, the wide adoption of modern mobile devices that feature localization and wireless connectivity, and advertisers' desire to reach users quickly based on users' surrounding context at a given time, introduces many new privacy threats, in particular those related to location-privacy. This creates a great research area with many challenges that could be further explored.

Bibliography

[1] Cisco Intrusion Detection Systems. http://www.google.com/products?q=cisco+intrusion+detection+system&aq=3&oq=cisco+in.

[2] DoubleClick Ad Planner by Google. https://www.google.com/adplanner/.

[3] Top Earning Blogs. http://onlineincometeacher.com/money/top-earning-blogs/.

[4] Tor Metrics Portal. https://metrics.torproject.org.

[5] The SSL Protocol, Version 3.0. http://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00, 1996.

[6] Bezos Calls Amazon Experiment "a Mistake". http://www.bizjournals.com/seattle/stories/2000/09/25/daily21.html, 2000.

[7] HTTP Over TLS. http://tools.ietf.org/html/rfc2818, 2000.

[8] Can't Stop the Pop–ups. http://news.cnet.com/2100-1024_3-5226273.html, 2004.

[9] Cardholders Targetted by Phishing Attack Using Visa-secure.com. http://news.netcraft.com/archives/2004/10/08/cardholders_targetted_by_phishing_attack_using_visasecurecom.html, 2004.

[10] Directive 2006/24/EC of the European Parlament and of the Council. *Official Journal of the European Union*, 2006.

[11] Click Through Rate of Google Search Results. http://www.redcardinal.ie/search-engine-optimisation/12-08-2006/clickthrough-analysis-of-aol-datatgz/, 2007.

[12] Growing Number Of ISPs Injecting Own Content Into Websites. http://www.techdirt.com/articles/20080417/041032874.shtml, 2008.

[13] Has Firefox 3 Certificate Handling Become Too Scary? http://www.betanews.com/article/Has-Firefox-3-certificate-handling-become-too-scary/1219180509, 2008.

[14] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. http://tools.ietf.org/html/rfc5280, 2008.

[15] ISPs Meddled With Their Customers' Web Traffic, Study Finds. http://www.pcworld.com/businesscenter/article/144682/isps_meddled_with_their_customers_web_traffic_study_finds.html, 2008.

[16] The TLS Protocol, Version 1.2. http://tools.ietf.org/html/rfc5246, 2008.

[17] Tim Callan's SSL Blog, MD5 attack resolved. https://blogs.verisign.com/ssl-blog/2008/12/on_md5_vulnerabilities_and_mit.php, 2008.

[18] 2008 Year-in-Review Benchmarks. *DoubleClick Research Report*, 2009.

[19] Biggest, Baddest Botnets: Wanted Dead or Alive. *PC World*, 2009.

[20] Botnet Caught Red Handed Stealing from Google. http://www.theregister.co.uk/2009/10/09/bahama_botnet_steals_from_google, 2009.

[21] Click Forensics Discovers Click Fraud Surge from New Sophisticated Bahama Botnet. http://www.clickforensics.com/newsroom/press-releases/144-bahama-botnet.html, 2009.

[22] Click Fraud Index. *ClickForennsics Inc.*, 2009.

[23] How Much Ads Cost. www.emarketer.com/Article.aspx?R=1007053, 2009.

[24] Network Bluepill: Stealth Router-based Botnet. http://dronebl.org/blog, 2009.

[25] Telecoms: Commission Launches Case Against UK Over Privacy and Personal Data Protection. http://europa.eu/rapid/pressReleasesAction.do?reference=IP/09/570, 2009.

[26] U.S. Average CPC by Category, June 2009. http://www.clickz.com/3634374, 2009.

[27] Viral Web Infection Siphons Ad Dollars from Google. http://www.theregister.co.uk/2009/05/14/viral_web_infection/, 2009.

[28] Australian Internet Security Initiative (AISI), Australian Communications and Media Authority. http://www.acma.gov.au/WEB/STANDARD/1001/pc=PC_310317, 2010.

[29] Comcast Reports Fourth Quarter and Year End 2009 Results. *Comcast Corporation*, 2010.

[30] EV and SSL Certificate Trends for the Top 100 Retailers. http://www.lexiconn.com/blog/2010/09/ev-ssl-top-100-retailers/, 2010.

[31] Guidelines For The Issuance and Management of Extended Validation Certificates. http://www.cabforum.org/Guidelines_v1_3.pdf, 2010.

[32] IAB Internet Advertising Revenue Report, 2009 Full Year Results. *Interactive Advertising Bureau*, 2010.

[33] Malvertising Attacks on Facebook Farm Town Players. http://www.spamfighter.com/News-14247-Malvertising-Attacks-on-Facebook-Farm-Town-Players.htm, 2010.

[34] Sponsored Malvertisement for Adobe Flash Player. http://stopmalvertising.com/malvertisements/sponsored-malvertisement-for-adobe-flash-player.html, 2010.

[35] Targeting Local Markets: An IAB Interactive Advertising Guide. Interactive Advertising Bureau, 2010.

[36] The Average CPM Rates Across Different Verticals. http://www.labnol.org/internet/average-cpm-rates/11315/, 2010.

[37] VeriSign Seal License Agreement. http://www.verisign.com.au/repository/seal/, 2010.

[38] Adometry Click Fraud Index. http://www.adometry.com/media/press/release.php?id=1, 2011.

[39] Arstechnica Opposition Letter. http://static.arstechnica.com/oppositionletter.pdf, 2011.

[40] Google One Pass. http://www.google.com/landing/onepass/, 2011.

[41] Home of the Mozilla Project. http://www.mozilla.org/, 2011.

[42] Improving SSL Certificate Security. http://googleonlinesecurity.blogspot.com/2011/04/improving-ssl-certificate-security.html, 2011.

[43] Mediacom Injecting Their Ads Into Other Websites. http://www.dslreports.com/shownews/Mediacom-Injecting-Their-Ads-Into-Other-Websites-112918, 2011.

[44] NYTimes' "Fair" Prices. http://www.mondaynote.com/2011/03/21/nytimes-%E2%80%9Cfair%E2%80%9D-prices/, 2011.

[45] OpenSSL: The Open Source Toolkit for SSL/TLS. http://www.openssl.org/, 2011.

[46] SQLite Home Page. http://www.sqlite.org/, 2011.

[47] SSL Certificate for Mozilla.com Issued Without Validation. http://www.sslshopper.com/article-ssl-certificate-for-mozilla.com-issued-without-validation.html, 2011.

[48] The EFF SSL Observatory, Electronic Frontier Foundation. http://www.eff.org/observatory, 2011.

[49] Trusted Certificates vs. Browser Recognized Certificates. http://www.instantssl.com/ssl-certificate-support/guides/ssl-certificate-validation.html, 2011.

[50] What Are the Types of SSL Certificates? http://www.globalsign.com/ssl-information-center/what-are-the-types-of-ssl-certificate.html, 2011.

[51] Your Phone Company is Selling Your Personal Data. http://money.cnn.com/2011/11/01/technology/verizon_att_sprint_tmobile_privacy, 2011.

[52] ab - Apache HTTP Server Benchmarking Tool. http://httpd.apache.org/docs/2.0/programs/ab.html, 2012.

[53] Adblock Plus – For Annoyance-Free Web Surfing. http://adblockplus.org, 2012.

[54] AdBrite Referral Program. http://www.adbrite.com/mb/affiliate_info.php, 2012.

[55] Alexa Analytics for Financial Times. http://www.alexa.com/siteinfo/ft.com, 2012.

[56] Alexa the Web Information Company. http://www.alexa.com, 2012.

[57] Anti-Botnet-Advisory Center – Association of the German Internet Industry with support from the Federal Office for Information Security. http://www.botfrei.de, 2012.

[58] Cisco VPN Client. http://www.cisco.com/en/US/products/sw/secursw/ps2308/index.html, 2012.

[59] Crypto++ 5.6.0 Benchmarks. www.cryptopp.com/benchmarks.html, 2012.

[60] Data Collection Arms Race Feeds Privacy Fears. http://www.reuters.com/article/2012/02/19/us-data-collection-idUSTRE81I0AP20120219, 2012.

[61] Financial Times, Digital Subscribers. http://aboutus.ft.com/corporate-information/ft-company/, 2012.

[62] Financial Times, Subscription Fees. https://registration.ft.com/signup/standard?execution=e1s1, 2012.

[63] FON: A Global, Community Wi-Fi Network. http://corp.fon.com, 2012.

[64] Free Internet Access Providers. http://www.thefreesite.com/Free_Internet_Access, 2012.

[65] Google Privacy Changes Must Be Stopped, Group's Lawsuit Says. http://www.businessweek.com/news/2012-02-13/google-privacy-changes-must-be-stopped-group-s-lawsuit-says.html, 2012.

[66] Google Privacy Policy. http://www.google.com/intl/en/policies/privacy/preview/, 2012.

[67] HostIP: My IP Address Lookup and GeoTargeting Community Geotarget IP Project. http://www.hostip.info/, 2012.

[68] Hotel's Free Wi-Fi Comes With Hidden Extras. http://bits.blogs.nytimes.com/2012/04/06/courtyard-marriott-wifi/, 2012.

[69] IP2Location: Bringing Location to the Internet. http://www.ip2location.com/, 2012.

[70] IPInfoDB: Free IP Adress Geolocation Tools. http://ipinfodb.com/, 2012.

[71] Making Ads More Interesting. http://googleblog.blogspot.com/2009/03/making-ads-more-interesting.html, 2012.

[72] MaxMind Geolocation and Online Fraud Prevention. http://www.maxmind.com/, 2012.

[73] OpenWrt – Wireless Freedom. http://openwrt.org, 2012.

[74] Operation Ghost Click. http://www.fbi.gov/news/stories/2011/november/malware_110911, 2012.

[75] PayPal Merchant Services. https://merchant.paypal.com/cgi-bin/marketingweb?cmd=_render-content&content_ID=merchant/digital_goods, 2012.

[76] Phorm. http://www.phorm.com/, 2012.

[77] Retargeting Ads Follow Surfers to Other Sites. http://www.nytimes.com/2010/08/30/technology/30adstalk.html, 2012.

[78] Samba's Advert-Supported 3G Data Service Launches in UK. http://www.bbc.co.uk/news/technology-18693200, 2012.

[79] Security Kiss. http://www.securitykiss.com/index.php?lang=en, 2012.

[80] Skyhook Location Performance. http://www.skyhookwireless.com/location-technology/performance.php, 2012.

[81] Strong VPN. http://www.strongvpn.com/, 2012.

[82] Telefonica Implements NAT for ADSL Users. http://bandaancha.eu/articulos/usuarios-adsl-movistar-compartiran-misma-7844, 2012.

[83] VeriSign Inc. http://www.verisign.com/ssl/buy-ssl-certificates/secure-site-services/index.html, 2012.

[84] A. Acquisti and H. R. Varian. Conditioning Prices on Purchase History. *Marketing Science*, 24, 2005.

[85] L. A. Adamic and B. A. Huberman. The Web's Hidden Order. *Communications of the ACM*, 44, 2001.

[86] R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2000.

[87] D. Ahmad. Two Years of Broken Crypto: Debian's Dress Rehearsal for a Global PKI Compromise. *IEEE Security and Privacy*, 6, 2008.

[88] R. Anderson and T. Moore. Information Security Economics – and Beyond. In *Proceedings of the 27th International Cryptology Conference on Advances in Cryptology (CRYPTO)*, 2007.

[89] S. Anderson and J. Gans. Platform Siphoning: Ad-Avoidance and Media Content. *American Economic Journal: Microeconomics*, 3(4), 2011.

[90] B. April, F. Hacquebord, and R. Link. A Cybercrime Hub. *A Trend Micro White Paper*, 2009.

[91] C. A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. An Obfuscation-Based Approach for Protecting Location Privacy. *IEEE Transactions on Dependable and Secure Computing*, 8(1), 2011.

[92] K. Asdemir, O. Yurtseven, and M. Yahya. An Economic Model of Click Fraud in Publisher Networks. *International Journal of Electronic Commerce*, 13(2), 2008.

[93] J. Aycock. *Spyware and Adware (Advances in Information Security)*. 2010.

[94] M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle. Flash Cookies and Privacy II: Now with HTML5 and ETag Respawning. *World Wide Web Internet And Web Information Systems*, 2011.

[95] S. W. B. Mungamuru and H. Garcia-Molina. Should Ad Networks Bother Fighting Click Fraud? (Yes, They Should.). Technical report, Stanford InfoLab, 2008.

[96] L. Backstrom, E. Sun, and C. Marlow. Find Me If You Can: Improving Geographical Prediction With Social and Spatial Proximity. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, 2010.

[97] A. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1), 2003.

[98] R. Biddle, P. C. van Oorschot, A. S. Patrick, J. Sobey, and T. Whalen. Browser Interfaces and Extended Validation SSL Certificates: An Empirical Study. In *Proceedings of the ACM workshop on Cloud computing security (CCSW)*, 2009.

[99] R. Böhme. Cyber-Insurance Revisited. In *Proceedings of the 4th Workshop on the Economics of Information Security (WEIS)*, 2005.

[100] M. Casado and M. J. Freedman. Peering Through the Shroud: The Effect of Edge Opacity on IP-Based Client Identification. In *Proceedings of the 4th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, 2007.

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

[102] E. Chien. Techniques of Adware and Spyware. In *Proceedings of the 15th Virus Bulletin Conference (VB)*, volume 47, 2005.

[103] C. Cho and H. Cheon. Why Do People Avoid Advertising on the Internet? *Journal of Advertising*, 33(4), 2004.

[104] C. Coarfa, P. Druschel, and D. S. Wallach. Performance Analysis of TLS Web Servers. *ACM Transactions on Computer Systems*, 24(1), 2006.

[105] N. Cohen. Whiting Out the Ads, but at What Cost? *The New York Times*, 2007.

[106] F. T. Commission. Protecting Consumer Privacy in an Era of Rapid Change: A Proposed Framework for Businesses and Policymakers. Preliminary FTC Staff Report, 2010.

[107] D. Coppersmith and M. Jakobsson. Almost Optimal Hash Sequence Traversal. In *Proceedings of the 6th International Conference on Financial Cryptography and Data Security (FC)*, 2003.

[108] J. Crowcroft. Net Neutrality: The Technical Side of the Debate: A White Paper. *SIGCOMM Computer Communication Review*, 37(1), 2007.

[109] G. Danezis, R. Dingledine, D. Hopwood, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 24th IEEE Symposium on Security and Privacy (S&P)*, 2003.

[110] N. Daswani and M. Stoppelman. The Anatomy of Clickbot.A. In *Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots)*, 2007.

[111] R. Dhamija, J. D. Tygar, and M. Hearst. Why Phishing Works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2006.

[112] T. Dinev, Q. Hu, and A. Yayla. Is There an On-line Advertisers' Dilemma? A Study of Click Fraud in the Pay-Per-Click Model. *International Journal of Electronic Commerce*, 13(2), 2008.

[113] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th Conference on USENIX Security Symposium (USENIX Security)*, 2004.

[114] N. Doraswamy and D. Harkins. *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks*. Prentice Hall PTR, USA, 1999.

[115] J. S. Downs, M. B. Holbrook, and L. F. Cranor. Decision Strategies and Susceptibility to Phishing. In *Proceedings of the 2nd Symposium on Usable Privacy and Security (SOUPS)*, 2006.

[116] B. Edelman, M. Ostrovosky, and M. Schwarz. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review*, 97(1), 2007.

[117] B. G. Edelman. Securing Online Advertising: Rustlers and Sheriffs in the New Wild West. *SSRN eLibrary*, 2008.

[118] B. G. Edelman. Deterring Online Advertising Fraud Through Optimal Payment in Arrears. *SSRN eLibrary*, 2009.

[119] S. M. Edwards, H. Li, J. hyun Lee, and S. M. Edwards. Forced Exposure and Psychological Reactance: Antecedents and Consequences of the Perceived Intrusiveness of Pop-up Ads. *Journal of Advertising*, 31, 2002.

[120] Federal Trade Commission. Protecting Consumer Privacy in an Era of Rapid Change: A proposed framework for businesses and policymakers. Preliminary FTC Staff Report, 2010.

[121] K. Fisher. Why Ad Blocking is Devastating to the Sites You Love. *Ars Technica*, 2010.

[122] R. Ford and S. Gordon. Cent, Five Cent, Ten Cent, Dollar: Hitting Botnets Where it Really Hurts. In *Proceedings of the Workshop on New Security Paradigms (NSPW)*, 2006.

[123] M. J. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan. Geographic Locality of IP Prefixes. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2005.

[124] J. Freudiger, R. Shokri, and J.-P. Hubaux. Evaluating the Privacy Risk of Location-Based Services. In *Proceedings of the 15th International Conference on Financial Cryptography and Data Security (FC)*, 2011.

[125] D. Fudenberg and D. Levine. Subgame–Perfect Equilibria of Finite– and Infinite–Horizon Games. *Journal of Economic Theory*, 31(2), 1983.

[126] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

[127] D. Fudenberg and J. Tirole. Perfect Bayesian Equilibrium and Sequential Equilibrium. *Journal of Economic Theory*, 53(2), 1991.

[128] M. Gandhi, M. Jakobsson, and J. Ratkiewicz. Badvertisements: Stealthy Click-Fraud with Unwitting Accessories. *Online Fraud, Part I Journal of Digital Forensic Practice*, 1(2), 2006.

[129] C. Gaspard, S. Goldberg, W. Itani, E. Bertino, and C. Nita-Rotaru. SINE: Cache-Friendly Integrity for the Web. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 2009.

[130] A. Ghose and U. Rajan. The Economic Impact of Regulatory Information Disclosure on Information Security Investments, Competition, and Social Welfare. In *Proceedings of the 5th Workshop on the Economics of Information Security (WEIS)*, 2006.

[131] A. Ghosh, R. Jana, V. Ramaswami, J. Rowland, and N. Shankaranarayanan. Modeling and Characterization of Large-Scale Wi-Fi Traffic in Public Hotspots. In *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM)*, 2011.

[132] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. A Framework for IP-Based Virtual Private Networks. RFC 2764, 2000.

[133] P. Golle and K. Partridge. On the Anonymity of Home/Work Location Pairs. In *Proceedings of the 7th International Conference on Pervasive Computing (Pervasive)*, 2009.

[134] N. Good, R. Dhamija, J. Grossklags, D. Thaw, S. Aronowitz, D. Mulligan, and J. Konstan. Stopping Spyware at the Gate: A User Study of Privacy, Notice and Spyware. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, 2005.

[135] G. Goodell and P. Syverson. The Right Place at the Right Time. *Communications of the ACM*, 50(5), 2007.

[136] Google Engineering Center Zurich. Technology and Innovation for Web Search. Private communication, October 2012.

[137] Google/Ipsos OTX MediaCT. Our Mobile Planet: United States. http://www.thinkwithgoogle.com/insights/library/studies/our-mobile-planet-us/, 2012.

[138] L. A. Gordon, M. P. Loeb, and T. Sohail. A Framework for Using Insurance for Cyber-Risk Management. *Communications of the ACM*, 46(3), 2003.

[139] J. Grossklags, N. Christin, and J. Chuang. Secure or Insure?: A Game-Theoretic Analysis of Information Security Games. In *Proceedings of the 17th International Conference on World Wide Web (WWW)*, 2008.

[140] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the 1st International Conference on Mobile systems, Applications and Services (MobiSys)*, 2003.

[141] S. Ha. An Intelligent System for Personalized Advertising on the Internet. In *Proceedings of the 5th International Conference on E-commerce and Web Technologies (EC-Web)*, 2004.

[142] J. Haskins. Commercial Skipping Technology and the New Market Dynamic: The Relevance of Antitrust Law to an Emerging Technology. *Duke Law & Technology Review*, (6), 2009.

[143] A. Herzberg and A. Jbara. Security and Identification Indicators for Browsers Against Spoofing and Phishing Attacks. *ACM Transactions on Internet Technology*, 8(4), 2008.

[144] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing Security and Privacy in Traffic-Monitoring Systems. *IEEE Pervasive Computing*, 5(4), 2006.

[145] Home Office. Retention of Communications Data under Part 11: Anti-Terrorism, Crime and Security Act 2001.

[146] HTML5 Geolocation. http://www.w3schools.com/html/html5_geolocation.asp.

[147] G. V. Hulme. Malvertising Continues to Pound Legitimate Web Sites. http://www.csoonline.com/article/675064/malvertising-continues-to-pound-legitimate-web-sites, 2011.

[148] Internet Advertising Bureau. IAB Internet Advertising Revenue Report, 2011 Full Year Results. http://www.iab.net/media/file/IAB_Internet_Advertising_Revenue_Report_FY_2011.pdf, 2011.

[149] P. Ipeirotis. Uncovering an Advertising Fraud Scheme. Or "The Internet is for porn". http://behind-the-enemy-lines.blogspot.com/2011/03/uncovering-advertising-fraud-scheme.html, 2011.

[150] W. Isaacson. How to Save Your Newspaper. *TIME Magazine*, 2009.

[151] M. O. J. Livingood, N. Mody and C. Communications. Recommendations for the Remediation of Bots in ISP Networks. Internet-Draft Version 3, IETF, 2009.

[152] C. Jackson and A. Barth. ForceHTTPS: Protecting High-Security Web Sites from Network Attacks. In *Proceedings of the 17th International Conference on World Wide Web (WWW)*, 2008.

[153] M. Jakobsson and S. Myers. *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley-Interscience, 2006.

[154] M. Jakobsson and Z. Ramzan. *Crimeware*. Addison-Wesley, Reading, MA, 2008.

[155] B. Johnson. Internet Companies Face Up to Malvertising Threat. http://www.guardian.co.uk/technology/2009/sep/25/malvertising, 2009.

[156] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. Towards IP Geolocation Using Delay and Topology Measurements. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2006.

[157] L. Kelly, G. Kerr, and J. Drennan. Avoidance of Advertising in Social Networking Sites: The Teenage Perspective. *Journal of Interactive Advertising*, 10(2), 2010.

[158] H. Kido, Y. Yanagisawa, and T. Satoh. An Anonymous Communication Technique using Dummies for Location-Based Services. In *Proceedings of the IEEE International Conference on Pervasive Services (ICPS)*, 2005.

[159] S. Kiernan, P. Krysiuk, A. Lelli, and G. O'Gorman. W32.Xpaj.B: Making Easy Money from Complex Code. *Symantec White Paper*, 2011.

[160] B. Krishnamurthy, D. Malandrino, and C. E. Wills. Measuring Privacy Loss and the Impact of Privacy Protection in Web Browsing. In *Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS)*, 2007.

[161] B. Krishnamurthy and C. E. Wills. Cat and Mouse: Content Delivery Tradeoffs in Web Access. In *Proceedings of the 15th International Conference on World Wide Web (WWW)*, 2006.

[162] J. Krumm. Inference Attacks on Location Tracks. In *Proceedings of the 5th International Conference on Pervasive Computing (Pervasive)*, 2007.

[163] N. Kshetri. The Economics of Click Fraud. *IEEE Security & Privacy*, 8(3), 2010.

[164] C. E. Landwehr. Improving Information Flow In the Information Security Market. *Economics of Information Security*, 12, 2004.

[165] A. Langley. Opportunistic Encryption Everywhere. In *Proceedings of the IEEE Oakland Web 2.0 Security and Privacy (W2SP)*, 2009.

[166] C. Larsen. Exploiting Trust in Advertising Networks. http://rocket.bluecoat.com/blog/exploiting-trust-advertising-networks, 2010.

[167] D. Lee. Micropayments: Would You Pay 20p to Read an Article? *BBC News*, 2012.

[168] M. Lelarge and J. Bolot. Economic Incentives to Increase Security in the Internet: The Case for Insurance. In *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, 2009.

[169] A. K. Lenstra. Key Length. *Contribution to The Handbook of Information Security*, 2004.

[170] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter. Ron was wrong, Whit is right. *IACR Cryptology ePrint Archive*, 2012.

[171] Z. Li, Q. Liao, and A. Striegel. Botnet Economics: Uncertainty Matters. In *Managing Information Risk and the Economics of Security*, 2009.

[172] J. Livingood, N. Mody, M. O'Reirdan, and Comcast Communications. ISP Voluntary Code of Practice for Industry Self-regulation in the Area of e-Security. Internet industry code of practice, Internet Industry Association, 2009.

[173] A. M. McDonald and L. F. Cranor. Americans' Attitudes About Internet Behavioral Advertising Practices. In *Proceedings of the 9th ACM Workshop on Privacy in the Electronic Society (WPES)*, 2010.

[174] K. Mochalski and H. Schulze. Deep Packet Inspection - Technology, Applications and Net Neutrality. *ipoque White Paper*, 2009.

[175] T. Moore and B. Edelman. *Measuring the Perpetrators and Funders of Typosquatting.* 2010.

[176] J. A. Muir and P. C. V. Oorschot. Internet Geolocation: Evasion and Counterevasion. *ACM Computing Survey*, 42, 2009.

[177] B. Mungamuru and S. Weis. Competition and Fraud in Online Advertising Markets. In *Proceedings of the 12th International Conference on Financial Cryptography and Data Security (FC)*, 2008.

[178] Y. Namestnikov. The Economics of Botnets. *Kaspersky Lab White Paper*, 2009.

[179] A. Okada. Perfect Bayesian Equilibrium and Sequential Equilibrium. In *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[180] W. Palant. Adblock Plus User Survey. http://adblockplus.org/blog/adblock-plus-user-survey-results-part-2, 2011.

[181] S. Parekh, R. Friedman, N. Tibrewala, and B. Lutch. Systems and Methods for Determining Collecting and Using Geographic Locations of Internet Users., 2004.

[182] S. Patil, G. Norcie, A. Kapadia, and A. Lee. "Check Out Where I Am!": Location-Sharing Motivations, Preferences, and Practices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems, Extended Abstracts (CHI)*, 2012.

[183] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP Geolocation Databases: Unreliable? *ACM SIGCOMM Computer Communication Review*, 41(2), 2011.

[184] A. Prasad, V. Mahajan, and B. Bronnenberg. Advertising Versus Pay-Per-View in Electronic Media. *International Journal of Research in Marketing*, 20(1), 2003.

[185] N. Provos, P. Mavrommatis, M. Rajab, and F. Monrose. All Your iFrames Point to Us. In *Proceedings of the 17th Conference on Security Symposium (USENIX Security)*, 2008.

[186] B. Raghavan, T. Kohno, A. C. Snoeren, and D. Wetherall. Enlisting ISPs to Improve Online Privacy: IP Address Mixing by Default. In *Proceedings of the 9th International Symposium on Privacy Enhancing Technologies (PETS)*, 2009.

[187] L. Rainie and K. Purcell. State of the News Media 2010: Online Economics and Consumer Attitudes. Report produced by the Pew Internet Project and the Pew Research Center's Project for Excellence in Journalism, 2010.

[188] C. Reis, S. D. Gribble, T. Kohno, and N. C. Weaver. Detecting In-Flight Page Changes with Web Tripwires. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.

[189] E. Rescorla. *SSL and TLS: Designing and Building Secure Systems*. Addison-Wesley, 2000.

[190] C. Riederer, V. Erramilli, A. Chaintreau, B. Krishnamurthy, and P. Rodriguez. For Sale : Your Data By : You. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets)*, 2011.

[191] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial In User Service (RADIUS). RFC 2865 (Proposed Standard), 2000.

[192] S. M. Ross. *Stochastic Processes*. Wiley, 1995.

[193] G. Rydstedt, E. Bursztein, D. Boneh, and C. Jackson. Busting Frame Busting: A Study of Clickjacking Vulnerabilities at Popular Sites. In *Proceedings of the IEEE Oakland Web 2.0 Security and Privacy (W2SP)*, 2010.

[194] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The Emperor's New Security Indicators. In *Proceedings of the 28th IEEE Symposium on Security and Privacy (S&P)*, 2007.

[195] S. Shah. Ad-Skipping and Time-Shifting: A Theoretical Examination of the Digital Video Recorder, 2011. Working paper. University of Virginia.

[196] M. Shapiro. Structure and Encapsulation in Distributed Systems: The Proxy Principle. In *Proceedings of the 6th International Conference on Distributed Computing Systems (ICDCS)*, 1986.

[197] G. Shaw. Spyware & Adware: The Risks Facing Businesses. *Network Security*, 2003(9), 2003.

[198] N. Shetty, G. Schwartz, M. Felegyhazi, and J. Walrand. Competitive Cyber-Insurance and Internet Security. In *Proceedings of the 8th Workshop on the Economics of Information Security (WEIS)*, 2009.

[199] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying Location Privacy. In *Proceedings of the 32nd IEEE Symposium on Security and Privacy (S&P)*, 2011.

[200] A. Sindik and G. Graybeal. Newspaper Micropayments and Millennial Generation Acceptance: A Brand Loyalty Perspective. *Journal of Media Business Studies*, 8(1), 2011.

[201] A. Singh and V. Potdar. Blocking Online Advertising – A State of the Art. In *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, 2009.

[202] W. Smith. Consumer Resistance to Marketing Reaches All-Time High, Marketing Productivity Plummets, According to Yankelovich Study (Yankelovich President J. Walker Smith Addressed Topic at AAAA Conference Today), 2004.

[203] N. Snow. The TiVo Question: Does Skipping Commercials Violate Copyright Law? *Syracuse Law Review*, 56(1), 2005.

[204] C. Soghoian and S. Stamm. Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL. *Available at SSRN*, 2010.

[205] C. Soghoian and S. Stamm. Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL. In *Proceedings of the 15th International Conference on Financial Cryptography and Data Security (FC)*, 2011.

[206] A. K. Sood and R. J. Enbody. Malvertising — Exploiting Web Advertising. *Computer Fraud & Security*, 2011(4), 2011.

[207] P. Speck and M. Elliott. Predictors of Advertising Avoidance in Print and Broadcast Media. *Journal of Advertising*, 26(3), 1997.

[208] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022, 2001.

[209] M. Stevens, A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. A. Osvik, and B. Weger. Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate. In *Proceedings of the 29th International Cryptology Conference on Advances in Cryptology (CRYPTO)*, 2009.

[210] T. Stühmeier and T. Wenzel. Getting Beer During Commercials: Adverse Effects of Ad-Avoidance. *Information Economics and Policy*, 23(1), 2011.

[211] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In *Proceedings of the 18th Conference on USENIX Security Symposium (USENIX Security)*, 2009.

[212] J. Tåg. Paying to Remove Advertisements. *Information Economics and Policy*, 22(4), 2009.

[213] Trend Micro Threat Research. A Cybercrime Hub. http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/a_cybercrime_hub.pdf, 2009.

[214] A. Tsow, M. Jakobsson, L. Yang, and S. Wetzel. Warkitting: The Drive-by Subversion of Wireless Home Routers. *Journal of Digital Forensic Practice*, 1(3), 2006.

[215] J. Turow, J. King, C. Hoofnagle, A. Bleakley, and M. Hennessy. Americans Reject Tailored Advertising and Three Activities that Enable It. Available at University of Pennsylvania Scholarly Commons, 2009.

[216] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang. Smart, Useful, Scary, Creepy: Perceptions of Online Behavioral Advertising. In *Proceedings of the 8th Symposium on Usable Privacy and Security (SOUPS)*, 2012.

[217] USA Department of Defenses. Global Positioning System: Standard Positioning Service Performance Standard. 2008.

[218] J. Vallade. AdBlock Plus and the Legal Implications of Online Commercial-Skipping. *Rutgers Law Review*, 61(3), 2009.

[219] H. Varian. Position Auctions. *International Journal of Industrial Organization*, 25(6), 2007.

[220] H. Varian, F. Wallenberg, and G. Woroch. The Demographics of the Do-Not-Call List. *IEEE Security & Privacy*, 3(1), 2005.

[221] N. Vratonjic, M. Raya, J.-P. Hubaux, and D. C. Parkes. Security Games in Online Advertising: Can Ads Help Secure the Web? In *Proceedings of the 9th Workshop on the Economics of Information Security (WEIS)*, 2010.

[222] N. Vratonjic, V. Bindschaedler, K. Huguenin, and J.-P. Hubaux. Location Privacy Threats at Public Hotspots. In *Proceedings of the 5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2012.

[223] N. Vratonjic, J. Freudiger, V. Bindschaedler, and J.-P. Hubaux. The Inconvenient Truth about Web Certificates. In *Proceedings of the 10th Workshop on the Economics of Information Security (WEIS)*, 2011.

[224] N. Vratonjic, J. Freudiger, and J.-P. Hubaux. Integrity of the Web Content: The Case of Online Advertising. In *Proceedings of the Workshop on Collaborative Methods for Security and Privacy (USENIX CollSec)*, 2010.

[225] N. Vratonjic, M. Hossein Manshaei, J.-P. Hubaux, R. Zhu, M. Jakobsson, and W. Leddy. *How Criminals Profit. In The Death of the Internet.* John Wiley & Sons, Inc., 2012.

[226] N. Vratonjic, M. H. Manshaei, J. Grossklags, and J.-P. Hubaux. Ad-blocking Games: Monetizing Online Content Under the Threat of Ad Avoidance. In *Proceedings of the 11th Workshop on the Economics of Information Security (WEIS)*, 2012.

[227] N. Vratonjic, M. H. Manshaei, M. Raya, and J.-P. Hubaux. ISPs and Ad Networks Against Botnet Ad Fraud. In *Proceedings of the 1st International Conference on Decision and Game Theory for Security (GameSec)*, 2010.

[228] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang. Towards Street-Level Client-Independent IP Geolocation. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, 2011.

[229] L. Weinstein. Google Hijacked – Major ISP to Intercept and Modify Web Pages. http://lauren.vortex.com, 2007.

[230] E. Weisstein. Euler-Maclaurin Integration Formulas. *MathWorld*, 2010.

[231] D. Wendlandt, D. G. Andersen, and A. Perrig. Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2008.

[232] D. Wendlandt, D. G. Andersen, and A. Perrig. Perspectives: Improving SSH-style Host Authentication with Multi-path Probing. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2008.

[233] T. Whalen and K. M. Inkpen. Gathering Evidence: Use of Visual Security Cues in Web Browsers. In *Proceedings of Graphics Interface Annual Conference (GI)*, 2005.

[234] K. Wilbur. A Two-Sided, Empirical Model of Television Advertising and Viewing Markets. *Marketing Science*, 27(3), 2008.

[235] Y. Xie, F. Yu, and M. Abadi. De-Anonymizing the Internet using Unreliable IDs. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2009.

[236] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How Dynamic are IP Addresses? In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2007.

[237] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi. Host Fingerprinting and Tracking on the Web: Privacy and Security Implications. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS)*, 2012.

[238] C. Zhang, C. Huang, K. Ross, D. Maltz, and J. Li. Inflight Modifications of Content: Who are the Culprits? In *Proceedings of the Workshop of Large-Scale Exploits and Emerging Threats (LEET)*, 2011.

[239] X. Zhao, F. Fang, and A. B. Whinston. An Economic Mechanism for Better Internet Security. *Decision Support Systems*, 45(4), 2008.

Index

Nevena Vratonjic

EPFL-IC-LCA
Station 14
1015 Lausanne, Switzerland

nevena.vratonjic@epfl.ch
http://people.epfl.ch/nevena.vratonjic
+41 21 693 3697

Personal

Born on October 19th, 1982 in Pozarevac, Serbia. Citizen of Serbia.
Languages: Serbian (native), English (fluent), French (basic), Russian (basic)

Education

Ph.D. in Communication Systems, EPFL, *Sep. 2007 – present*
Thesis: "Security, Privacy and Economics of Online Advertising"
Thesis director: Prof. Jean-Pierre Hubaux

M.Sc. in Computer Science, EPFL, *Oct. 2006 – Sep. 2007*

M.Sc. in Communication Systems, Belgrade University, *Oct. 2001 – Jun. 2006*
Thesis: "Spectrum: Overlay Network Bandwidth Provisioning"
Advisors: Prof. Aleksandra Smiljanic and Prof. Dejan Kostic

Professional Experience

Research and Teaching Assistant *Sep. 2007 – present*
Laboratory of Computer Communications and Applications, EPFL

Research Assistant *Jun. 2006 – Apr. 2007*
Networked Systems Lab, EPFL

Research Assistant *Oct. 2005 – Mar. 2006*
University of Belgrade, Serbia

Professional Activities

Reviewer for scientific journals and conferences
IEEE TMC, IEEE WCM-WISEC, ACM CCS, PETS, WEIS, ACM WiSec, IEEE GameSec

Teaching

Teaching Assistant
Computer Networks, Mobile Networks, Security and Cooperation in Wireless Networks

Supervised Research Student Projects
Vincent Bindschaedler, *Survey on Deployment of SSL Certificates*
Vincent Bindschaedler, *Modeling Location-Privacy Threats due to Shared IP Addresses*

Supervised Student Projects
Alevtina Dubovitskaya, *Location-Privacy Threats at HotSpots*
Vincent Brillault, *Measuring Web-browsing Footprints on Google Ad Network*
Romain Poiffaut, *Measuring Web-browsing Footprints on Google Ad Network*
Alexandra Oltaneu, *Tell Me Your Ads and I'll Tell You Who You Are*
Vincent Bindschaedler, *Security Issues in Mobile Advertising*
Asli Bay, *Effects of Ad-avoidance on the Online Advertising Revenue Model*
Maximilien Cuony, *Survey on SSL Certificates Deployment*
Selma-Yasmine Chouaki, *Privacy Issues in Mobile Advertising*
Robin Francois, *Location Privacy Threats in Mobile Advertising*
Tomasz Trzcinski, *Analyse, Target & Advertise: Privacy in Mobile Advertising*
Acacio Martins and Emanuel Cino, *FriendFinder Application for Android Smartphones*
Maxime Augier, *Hidden Channels for 802.11 Access Point Botnets*
Anthony Durussel, *Integrity of the Web Content: the Case of Online Advertising*
Anjan Som, *Injecting Sponsored Points-of-Interest on Google Maps*
Paul Landry, *Security Vulnerabilities of YouTube Ads*
Raoul Neu, *Electronic Landmines: Boobytrapping the Firmware of Wireless Access Points*
Yi Liu, *Modeling Changes on the Internet Caused by Web Insecurity*
David Klopfenstein, *BeSafe: Browser Extension for Safe Browsing*

Supervised Master Theses
Michael Jubin, *Hide & Seek: Security of Location Based Services*

Publications

N. Vratonjic, V. Bindschaedler, K. Huguenin and J.-P. Hubaux. **Location Privacy Threats at Public Hotspots**, *The Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2012.

N. Vratonjic, M. H. Manshaei, and J.-P. Hubaux. **How Criminals Profit**, in *The Death of the Internet*, John Wiley & Sons, Inc., 2012.

N. Vratonjic, M. H. Manshaei, J. Grossklags, and J.-P. Hubaux. **Ad-blocking Games: Monetizing Online Content Under the Threat of Ad Avoidance**, *Proceedings of the 11th Workshop on Economics of Information Security (WEIS)*, 2012.

N. Vratonjic, J. Freudiger, V. Bindschaedler, and J.-P. Hubaux. **The Inconvenient Truth about Web Certificates**, *Proceedings of the 10th Workshop on Economics of Information Security (WEIS)*, 2011.

N. Vratonjic, M. H. Manshaei and J.-P. Hubaux. **ISPs and Ad Networks Against Botnet Ad Fraud**, *IEEE E-Letter of Multimedia Communications Technical Committee (MMTC) on GameSec and Multimedia Security, vol. 6, num. 4, p. 11-14*, 2011.

N. Vratonjic, M. Manshaei, M. Raya, and J.-P. Hubaux. **ISPs and Ad Networks Against Botnet Ad Fraud**, *Proceedings of the First International Conference on Decision and Game Theory for Security (GameSec)*, 2010.

N. Vratonjic, J. Freudiger, and J.-P. Hubaux. **Integrity of the Web Content: The Case of Online Advertising**, *Usenix Collaborative Security (CollSec)*, 2010.

N. Vratonjic, M. Raya, J.-P. Hubaux and D. C. Parkes. **Security Games in Online Advertising: Can Ads Help Secure the Web?**, *Proceedings of the 9th Workshop on the Economics of Information Security (WEIS)*, 2010.

J. Freudiger, N. Vratonjic, and J.-P. Hubaux. **Towards Privacy-Friendly Online Advertising**, *IEEE Web 2.0 Security and Privacy (W2SP)*, 2009.

N. Vratonjic, P. Gupta, N. Knezevic, D. Kostic, A. Rowstron. **Enabling DVD-like Features in P2P Video-on-demand Systems**, *ACM SIGCOMM Workshop on Peer-to-Peer Streaming and IP-TV systems (P2P-TV)*, 2007.