Online Social Networks : Is it the end of Privacy ?

Abdelberi Chaabane

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Abdelberi Chaabane

Thèse dirigée par **Dr. Claude Castelluccia et Dr. Mohamed Ali Kaafar**

préparée au sein d' **INRIA Rhônes-Alpes, équipes Privatics**
et de **l'École Doctorale : Mathématiques, Sciences et Technologies de l'Information, Informatique**

Online Social Networks: Is it the end of Privacy?

Thèse soutenue publiquement le ,
devant le jury composé de :

M. Kave Salamatian
Professeur, Polytech Annecy-Chambéry, Président
Mme. Anne-Marie Kermarrec
Directrice de recherche, INRIA Rennes, Rapporteur
M. Refik Molva
Professeur, Eurecom Sophia Antipolis, Rapporteur
M. Gene Tsudik
Professeur, University of California, Irvine, Examinateur
M. Emiliano De Cristofaro
Senior Lecturer, University College London, Examinateur
M. Claude Castelluccia
Directeur de recherche, Inria, Directeur de thèse
M. Mohamed Ali Kaafar
Chargé de recherche, Inria, Directeur de thèse

To Ommi, Zouhour and Ines

Abstract

Sharing information between users constitutes the cornerstone of the Web 2.0. Online Social Networks (OSN), with their billions of users, are a core component of this new generation of the web. In fact, OSNs offer innovative services allowing users to share their self-generated content (e.g., status, photos etc.) for free. However, this free access is usually synonymous with a subtle counterpart: the collection and usage of users' personal information in targeted advertisement. To achieve this goal, OSN providers are collecting a tremendous amount of personal, and usually sensitive, information about their users. This raises concerns as this data can be exploited by several entities to breach user privacy. The primary research goals of this thesis are directed toward understanding the privacy impact of OSNs.

Our first contribution consists in demonstrating the privacy threats behind releasing personal information publicly. Two attacks are constructed to show that a malicious attacker (i.e., any external attacker with access to the public profile) can breach user privacy and even threaten his online security.

Our first attack shows how seemingly harmless interests (e.g., music interests) can leak privacy-sensitive information about users. In particular, we infer their undisclosed (private) attributes using the public attributes of other users sharing similar interests. Leveraging semantic knowledge from Wikipedia and a statistical learning method, we demonstrated through experiments —based on more than 104K Facebook profiles— that our inference technique efficiently predicts attributes that are very often hidden by users.

Our second attack is at the intersection of computer security and privacy. In fact, we show the disastrous consequence of privacy breach on security by exploiting user personal information —gathered from his public profile— to improve the password cracking process. First, we propose a Markov chain password cracker and show through extensive experiments that it outperforms all probabilistic password crackers we compared against. In a second step, we systematically analyze the idea that additional personal information about a user helps in speeding up password guessing. We propose a methodology that exploits this information in the cracking process and demonstrate that the gain can go up to 30%.

These studies clearly indicate that publicly disclosing personal information harms privacy, which calls for a method to estimate this loss. Our second contribution tries to answer this question by providing a quantitative measure of privacy. We propose a practical, yet formally proved, method to estimate the uniqueness of each profile by studying the amount of information carried by public profile attributes. To achieve our goal, we leverage Ads Audience Estimation platform and an unbiased sample of more than 400K Facebook

public profiles. Our measurement results show that the combination of gender, current city and age can identify close to 55% of users to within a group of 20 and uniquely identify around 18% of them.

In the second part of this thesis, we investigate the privacy threats resulting from the interactions between the OSN platform and external entities. First, we explore the tracking capabilities of the three major OSNs (i.e., Facebook, Google+ and Twitter) and show that "share-buttons" enable them to persistently and accurately track users' web activity. Our findings indicate that OSN tracking is diffused among almost all website categories which allows OSNs to reconstruct a significant portion of users' web profile and browsing history.

Finally, we develop a measurement platform to study the interaction between OSN applications — of Facebook and RenRen — and fourth parties. We show that several third party applications are leaking user information to "fourth" party entities such as trackers and advertisers. This behavior affects both Facebook and RenRen with varying severity.

Résumé

Les réseaux sociaux en ligne (OSNs) collectent une masse de données à caractère privé. Le recueil de ces données ainsi que leur utilisation relèvent de nouveaux enjeux économiques et suscite, à juste titre, de nombreuses interrogations notamment celles relatives à la protection de la vie privée. Nous nous sommes proposés dans cette thèse de répondre à certaines de ces interrogations.

Dans le premier chapitre nous analysons l'impact du partage des données personnelles de l'utilisateur sur sa vie privée. Tout d'abord, nous montrons comment les intérêts d'un utilisateur – à titre d'exemple ses préférences musicales – peuvent être à l'origine de fuite d'informations sensibles. Pour ce faire, nous inférons les attributs non divulgués du profil de l'utilisateur en exploitant d'autres profils partageant les mêmes "goûts musicaux". Nos expérimentations réalisées sur plus de 104,000 profils publics collectés sur Facebook montrent que notre technique d'inférence prédit efficacement les attributs qui sont très souvent cachés par les utilisateurs.

Dans un deuxième temps, nous exposons les conséquences désastreuses du partage des données privées sur la sécurité. Nous nous focalisons sur les informations recueillies à partir de profils publics et montrons comment celles-ci peuvent être exploitées pour accélérer le craquage des mots de passe. Premièrement, nous proposons un nouveau « craqueur » de mots de passe basé sur les chaînes de Markov permettant le cassage de plus de 80% des mots de passe, dépassant ainsi toutes les autres méthodes de l'état de l'art. Deuxièmement, et afin de mesurer l'impact sur la vie privée, nous proposons une méthodologie qui intègre les informations personnelles d'un utilisateur afin d'accélérer le cassage de ses mots de passe.

Nos résultats mettent en évidence la nécessité de créer de nouvelles méthodes d'estimation des fuites d'informations personnelles. Nous proposons pour cela une méthode formelle pour estimer l'unicité de chaque profil en quantifiant l'information portée par chaque attribut public.

Notre travail se base sur la plate-forme publicitaire d'estimation des utilisateurs de Facebook pour calculer l'entropie de chaque attribut public. Ce calcul permet d'évaluer l'impact du partage d'informations. Nos résultats, basés sur un échantillon de plus de 400,000 profils publics Facebook, montrent que la combinaison des attributs: sexe, ville de résidence et âge permet d'identifier d'une manière unique environ 18% des utilisateurs.

Dans la deuxième section de notre thèse nous analysons les interactions entre la plate-forme du réseau social et des tiers et son impact sur la vie privée des utilisateurs.

Dans une première étude, nous explorons les capacités de « tracking » des réseaux sociaux Facebook, Google+ et Twitter. Nous étudions les mécanismes qui permettent à ces

services de suivre d'une façon persistante l'activité web des utilisateurs ainsi que d'évaluer sa couverture. Nos résultats indiquent que le « tracking » utilisé par les OSNs couvre la quasi-totalité des catégories Web, indépendamment du contenu et de l'auditoire.

Finalement, nous développons une plate-forme de mesure pour étudier l'interaction entre les plates-formes OSNs, les applications sociales et les « tierces parties » (e.g., fournisseurs de publicité). Nous démontrons que plusieurs applications tierces laissent filtrer des informations relatives aux utilisateurs à des tiers non autorisés. Ce comportement affecte à la fois Facebook et RenRen avec une sévérité variable: 22% des applications Facebook testées transmettent au moins un attribut à une entité externe. Quant à, RenRen, nous démontrons qu'il souffre d'une faille majeure causée par la fuite du jeton d'accès dans 69% des cas.

Acknowledgment

I would like to express my sincere appreciation to all the people who helped me during my doctoral work.

My advisor, Dali, has been a continual source of inspiration and support throughout the years that I have had the pleasure to work with him. Dali achieved the magical balance between guidance and freedom of exploring new directions. For these reasons and more, I am extremely grateful.

I am thankful to Claude whose teaching and feedback have tremendously contributed to my research education.

To my friend Gergely, a special debt of gratitude is owed. Since coming to Privatics as a postdoc, he has been an unflagging fount of knowledge and advice, setting an example as a researcher that I have strived, in my own limited way, to emulate.

I am indebted to a number of exceptional people and researchers: Roksana Boreli, Emiliano De Cristofaro, Ersin Uzun, Markus Dürmuth, Engin Kirda and Keith Ross – just to mention a few. It is not easy to explain how much I learned from them.

I have also been very fortunate to work with other great folks: Stevens Le Blond, Terence Chen, Pierre Ugo Tournoux, Mathieu Cunche, Arik Friedman, Tobias Lauinger, Kaan Onarlioglu and many others.

My gratitude goes also to all the friends who made my stay at INRIA and Grenoble an enjoyable journey. Many thanks to Pere with whom I shared hard and pleasant moments (I hope that you found, in the ocean the dream, that you are ever running after), to Hedi Harzallah for his wise advice and support, to Amine the machine learning instructor :), to my roommates Christophe, Hedi, Morgane and Martin.

Thanks to my beloved parents Fatym and Rachid, my sister Zouhour (the rock), my brother Fadhel and his wife Juanita for their support.

Finally, my special thanks to Ines for her unconditional love, support, and patience over the years.

Contents

List of Figures

List of Tables

List of publications

Abdelberi Chaabane, Pere Manils, and Mohamed Ali Kaafar. Digging Into Anonymous Traffic: A Deep Analysis of the TOR Anonymizing Network. In *4th International Conference on Network and System Security (NSS)*, 2010.

Stevens Le Blond, Pere Manils, Abdelberi Chaabane, Mohamed Ali Kaafar, Claude Castelluccia, Arnaud Legout, and Walid Dabbous. One Bad Apple Spoils the Bunch: Exploiting P2P Applications to Trace and Profile Tor Users. In *Proceedings of the 4th USENIX Conference on Large-scale Exploits and Emergent Threats (LEET)*, 2011.

Abdelberi Chaabane, Gergely Acs, and Mohamed Ali Kaafar. You Are What You Like! Information Leakage Through Users' Interests. In *19th Annual Network & Distributed System Security Symposium (NDSS)*, 2012.

Abdelberi Chaabane, Mohamed Ali Kaafar, and Roksana Boreli. Big Friend is Watching You: Analyzing Online Social Networks Tracking Capabilities. In *Proceedings of the 2012 ACM Workshop on online social networks (WOSN)*, 2012.

Abdelberi Chaabane, Emiliano De Cristofaro, Mohamed Ali Kaafar, and Ersin Uzun. Privacy in Content-Oriented Networking: Threats and Countermeasures. *ACM Computer Communication Review (CCR)*, July 2013.

Terence Chen, Abdelberi Chaabane, Pierre Ugo Tournoux, Mohamed-Ali Kaafar, and Roksana Boreli. How Much Is Too Much? Leveraging Ads Audience Estimation to Evaluate Public Profile Uniqueness. In *13th Privacy Enhancing Technologies Symposium (PETS)*. 2013.

Tobias Lauinger, Kaan Onarlioglu, Abdelberi Chaabane, Engin Kirda, William Robertson, and Mohamed Ali Kaafar. Holiday Pictures or Blockbuster Movies? Insights into Copyright Infringement in User Uploads to One-Click File Hosters. In *Research in Attacks, Intrusions, and Defenses (RAID)*. 2013.

Claude Castelluccia, Abdelberi Chaabane, Markus Dürmuth, and Daniele Perito. When Privacy meets Security: Leveraging Personal Information for Password Cracking. *arXiv preprint arXiv:1304.6584*, 2013.

Abdelberi Chaabane, Yuan Ding, Ratan Dey, Mohamed Ali Kaafar, and Keith W. Ross. A Closer Look at Third-Party OSN Applications: Are They Leaking Your Personal Information? *Passive and Active Measurement Conference (PAM)*, 2014.

Abdelberi Chaabane, Mathieu Cunche, Terence Chen, Arik Friedman, Emiliano De Cristofaro, and Mohamed-Ali Kaafar. Censorship in the Wild: Analyzing Web Filtering in Syria. *arXiv preprint arXiv:1402.3401*, 2014.

Chapter 1

Introduction

Contents

1.1 What is Privacy?

The traditional definition of "privacy" is the limit which the law sets for others to penetrate your *private sphere*. As such, privacy is conceived as a way to constrain the power of any entity —particularly the government — by explicitly framing the context in which these entities have the right to snoop into one's private affairs (e.g., enter one's home and/or search one's papers). Accordingly, a possible definition of privacy is the one provided by Warren and Brandeis back in 1890 which states that privacy is "*the right to be left alone*". Hence, from a legal perspective, privacy represents the set of legal restrictions on the power of others to invade your private sphere [1].

This definition might seem awkward as privacy is defined as a function of the private sphere, which is in turn, hard to define. However, back in the 19th century[1], it was easy to define the private sphere as almost all possession were *tangible* and therefore, explicitly subject to law (e.g., the fourth amendment). The process of invading someone's privacy

[1]For instance the Fourth Amendment was introduced in Congress in 1789 by James Madison.

through data gathering is also classified in two classes according to the nature of the collected data: *monitoring* for intangible elements and *searching* for the tangible ones.

Specifically, the monitored part of one public sphere is the part of one's life that others can see (or respond to when asked). It is (i) *transitory* as people will only notice unusual events; (ii) *inaccurate* as multiple people are likely to have a different — and sometimes contradictory — picture of the information; (iii) *costly* as it is hard to collect and process such data.

The searchable part, however, is the portion of our life that is "recorded" and hence easy to query (e.g., diary, hard drive, smartphone). This data is usually not ephemeral, provides an accurate (though potentially incomplete) restitution of the information and can be reviewed at any time.

Throughout our history, the cost and inaccuracy of monitoring was sufficient to prevent people from snooping into one's private life. Only the searchable part of the data gathering process —usually tangible — was regulated. However, digital technologies change this balance *radically*. They do not only make large scale behavioral monitoring possible, but also make individuals' behavior searchable. Nowadays, the same technology is able to monitor, index, make most parts, if not all, of our daily activities searchable on large scale. However, as both monitoring and searching activities are today carried out by machines, their impact on user privacy has to be assessed.

1.1.1 Machines and Privacy

"Machine collection and processing of data, cannot, as such, invade privacy" This is how Judge Richard Poster defended the Bush administration's extensive surveillance of (domestic) communications[2] in a Washington Post article. This argument is recurrent and easily explainable: machines are rational not sentient beings and as such react based on a set of predefined rules. There is no *moral* judgment as it is the case with humans but only a *logic* behavior. Hence, the "monitoring" machine does not care about your problems at work or whether you cheated in the last exam, it only cares about what we (they) asked it for: terrorism in Judge Richard Poster's scenario.

However, does using machines to process private information resolves privacy problems? What would happen if we knew that all, or a significant part, of our activities are monitored? This would certainly challenge our feeling of "being left alone". A nice scenario depicting such a situation was provided by Lawrence Lessig [1]: "Imagine a television that shifted its advertisement as it heard what you are talking about ". Would we accept such a system?

While it is hard to provide a universal answer, it is safe to say that most people will at best feel uncomfortable with such a system. From a psychological point of view, some events are private and are meant to remain so, feeling "monitored" will change our way of thinking and behaving, affecting our interactions permanently.

[2]Regarding NSA warrantless surveillance controversy `http//en.wikipedia.org/wiki/NSA_warrantless_surveillance_(2001%E2%80%9307)`

1.2 Why Should We Care?

The *de facto* answer to this question is the misuse of the data. An untrustworthy person can misuse the system, breaching intentionally, the privacy of others. However, this is not the only way privacy can be violated. In fact, the simple usage of someone's information without his *informed consent* is clearly a violation of his privacy. Defining this informed consent in the context of new technologies/Internet is challenging as there are countless numbers of scenarios where such data is collected and used. Hence, drawing a clear "line" between what can be considered as private information and what is not is a complex problem. A first step towards understanding digital privacy threats is to identify the *goals* behind collecting data.

Surveillance The tip of the data collection iceberg is the one collected legally to either prove or (hopefully) forestall crimes (e.g., video surveillance, phone calls). This lawful data interception — carried out by governments and official institutions — is security-driven. However, the amount of collected data and its usage are controversial. To illustrate this, let us take surveillance cameras (or CCTV) as an example. For proponents, these cameras reduce criminality as they easily allow resolve criminal affairs. The "nothing to hide" slogan is usually used to claim that "honest" citizens have nothing to hide and hence recording them does not represent a serious problem. Opponents of this approach argue that the "surveillance state" is collecting a tremendous amount of data about citizens which represents a threat to their privacy. Moreover, several abuses have been reported showing that these cameras can be easily misused [2]. Hence, the benefits of deploying video surveillance must be balanced against the costs and risks.

Malicious Attacks Various companies build their entire business model on selling user data to others or using it to conduct malicious attacks (e.g., phishing). We believe that such behavior represents a clear violation privacy: first, the gathered data is usually unknown or very hard to assess. Second, the selling market is opaque as information about buyer(s) and prices are not available. Finally, the data outcome (i.e., for which purpose the data is being used for) is completely unknown. This business model is clearly in complete opposition with the principle of informed consent.

Marketing and advertisement This data collection is operated by most companies — either online or offline — to better target their customers (e.g., Walmart shopping card or user preferences in Amazon.com). Gathered data span a wide range of personal, usually sensitive, information. One might argue that a user is trading (a bit) of its privacy for better services: ads tailored to the user needs, discount based on past purchases etc. More importantly, in the online scenario, ads – and indirectly our personal information — allow us to use Facebook, Gmail and many other services free of charge. In short, one can claim that it is a win-win situation.

While the necessity of collecting data to better serve customers is unquestionable, the techniques that are actually used to achieve this goal represent a real threat to our privacy. This danger stems from two characteristics of the current systems: (i) the perfection and coverage of the gathered information and (ii) the (roughly) unlimited life time of data availability. These two properties breach user privacy as they allow *user profiling* at large scale.

1.2.1 User Profiling

Behavioral advertisement, profiling and adver-gaming illustrate how user personal information and social relations have been integrated into the market model. In other words, user information is commodified: the user identity becomes a commodity to be sold and bought. Such radical change allows a major shift in commerce: the offer is tailored to a specific (targeted) customer. However, understanding the how and why of user privacy threats requires a retrospective analysis of the tracking ecosystem.

First Party Tracking The genesis of tracking arises from the stateless nature of HTTP protocol: each request is treated as an independent transaction and the server does not retain any information about the communication nor the status of each request. While the stateless characteristic of HTTP fitted the first needs of the web — displaying static content —, it quickly became a barrier to building more complex transactional systems. In February 1997[3], cookies were proposed to resolve this shortcoming by allowing the first party server to store, temporarily, key-value data in the user browser and create *sessions*. Interestingly, the two first RFCs that standardized cookies (RFC 2109 and 2965) specified that the browser should protect the user privacy by forbidding the share of cookies between servers. However, the newer RFC (RFC 6265) explicitly allows user agents to implement whichever third-party cookie policy they want.

Third-Party Tracking Third parties started to exploit cookies to track users across multiple websites. The approach is quite simple; whenever a user visits a web page with a third party content (e.g., ads), a cookie is set (or updated) — without the user consent — allowing the tracker to trace a significant part of the user navigation [3]. This tracking is highly pervasive and is usually referred to as *behavior tracking* or *user profiling*. This knowledge of the pages visited by a user allows the advertising company to target advertisements to users presumed preferences. The endless race between advertisers gave birth to a multitude tracking techniques, while cookies are still the predominant one, multiple others are being developed and used (e.g., ever-cookie) [4, 5]. Third party tracking and profiling represent a threat to user privacy as not only a tremendous amount of data is collected but also the collection is done by a steadily decreasing number of entities [3].

1.3 What about Online Social Networks?

The advent of Online Social Networks (OSNs) radically changes the ecosystem. These new players, with far more personal data, started to compete in the advertising arena. OSNs exploit the huge amount of data to conduct user profiling and targeted advertisement. The fundamental differences are not only the amount of collected data, but also its coverage. In fact, OSN operators are not only collecting a wide range of personal and sensitive information such as photos, friends, contacts, locations, likes etc. but also deploying tracking mechanisms that can silently and persistently trace users' browsing activities outside the OSN platform (as we will show in Chapter 6).

From a privacy perspective, the profiling has shifted from abstract (i.e., where the targeted user is virtual) to physical (i.e., the OSN is able to link the data to a physical

[3]RFC2109 `http://tools.ietf.org/html/rfc2109`

person). Let us analyze the impact of these worrisome threats according to the *goals* model previously defined.

(Mass) Surveillance From a government perspective OSNs offer an unprecedented access to user private data. This access can be *legally* supported or through *spying activities*. The first approach targets a small range of users that are suspected of criminal activities[4] and relies on subpoena. As OSN operators have to comply with the country legislation, they are obligated to provide user data to government when asked. This data contains most of the user information (demographic, friends, photos, IP, phone number etc.).[5]

However, data collected through lawful interception and subpoena seems to be small compared to the ones collected through spying. In fact, several countries are engaged in large scale monitoring of social networks. In the US for example, numerous agencies are deploying different interception techniques to gather data from OSNs among which: Justice Department, FBI, NSA and others.[6] For instance, PRISM[7] — the NSA framework of systematic and massive data collection of user information — allows the U.S. intelligence community to gain access from nine Internet companies —among which the three major OSN providers — to a wide range of data including: meta-data, emails, social profiles etc.

As OSN data are diverse, personal, sensitive and regularly updated, the threats to user privacy from a non legally supported interception are extremely severe.

Malicious Attacks Malicious attackers can exploit the OSN platform in several ways. They can infer missing or hidden information about the user through e.g., machine learning techniques [6, 7] or exploiting friendship information [8]. Sybil nodes can be created then linked in away to form a *social botnet* aiming at collecting massive amounts of data [9] or sending spams [10]. The attacker can also exploit social applications [11, 12] or mobile social application [12, 13] to collect data. All collected information can be used in other attacks among which social engineering, spear phishing (i.e., spam mails that use the user information to create the illusion that the mail is authentic) or targeted attacks (e.g., Advanced Persistent Threat APT[8]).

Marketing and Advertisement This data collection is done either by the OSN itself, or by third parties. In the first scenario, the amount of provided information as well as its diversity represent a privacy threat to user's privacy. Moreover, this data can be extended through different machine learning techniques. For instance, using user interests, OSN operators can infer information such as user marital status, origin [6], race, IQ, sexuality, substance use and political views [14]. OSN operators also deploy tracking mechanisms to monitor user navigation patterns outside the social network which allows them to create an accurate user profile [15] . Third party websites can also collect several pieces of

[4]Note that some OSNs might also provide data in case of civil litigants.
[5]See `http://goo.gl/bFyvwe` for subpoena of different OSNs.
[6]`http://goo.gl/1S5uBx`
[7]`http://en.wikipedia.org/wiki/PRISM_(surveillance_program)`
[8]`http://goo.gl/8dFUjn`

information about the user either through the OAuth protocol[9] (i.e., the website allows the user to connect with his Facebook login) or when such personal information is leaked from the OSN [16].

1.4 What Can We Do?

In this section we argue that privacy is a multidimensional problem and that any solution addressing a single dimension is at best a partial solution. We show that privacy decisions are complex due to several factors and that *soft paternalism* is a promising approach for protecting user privacy.

1.4.1 The Multiple Dimensions of Privacy

Lawrence Lessig argues in his book *Code 2.0* [1] that there is no single solution to privacy problems on the Internet. He identifies four dimensions to address privacy threats: Law, architecture/code (technology), markets and norms. Law represents legal regulations relative to user privacy. Technology can be used to either create a privacy by design products or to develop privacy enhancing tools. Market can create an incentive for and benefit from privacy. Finally, companies can create norms to enhance user privacy (e.g., norms for sharing data). Lessig shows that any privacy solution should be a mix of at least two modalities and that addressing a single dimension would be at best a partial fix. In the following, we show how the legal or the technological dimension may fail to protect privacy if used separately.

1.4.1.1 Privacy By Law

The most visible privacy artifact on the Internet is *privacy policy*. One simple Google search for the term "privacy policy" returns more than 8 billion hits. The goal of this legal text is to explain which data is being stored and/or processed and to whom it can be transferred. These regulations differ from one country to another and present a wide spectrum of severity regarding what can be collected. If the data collector resides in Europe then it is subject to the "Organization for Economic and Co-operation and Development (OECD) Guidelines on the Protection of Privacy and Trans border Flows of Personal Data". The main goal of these guidelines is to provide a standard definition of privacy to ease the transfer of data with collaborating countries. The EU Data Protection Directive (a.k.a, 95/46/EC) is the implementation of these guidelines in the European community. Each EU member is then responsible for translating this directive into a national law.

These policies are the product of a *traditional* government approach to regulation that hardly bear the specificity of Internet. The government was pushed to solve the problem of Internet privacy. Its solution was to require the user to read and accept the service privacy policy. Let us analyze the possible reasons of the failure of such a legal approach to protect user data.

A first question is *whether users read the privacy policy?* According to the Internet Society's Global Internet User Survey [17], only 16% of internet users read privacy policies. Of those who do, only 20% actually understand them. Multiple reasons explain why users

[9] `http://en.wikipedia.org/wiki/OAuth`

skim through or entirely skip these texts: First, they are on average 2,500 words long requiring more than 10 minutes for reading [18]. Second, the technical writing (legal lexicon) makes them hard to understand. Finally, they are binary: the user has to (fully) accept them in order to use the system.

A second question is *whether privacy laws are enforced?* In 2010, Burghardt *et al.* [19] showed that many data collectors within the EU jurisdiction were in conflict with the EU Data Protection Directive. One hundred service provider privacy policies were inspected to check the obligation to inform the user about data acquisition and handling, data forwarding and automated processing. Results showed that only a single operator specifies to whom the data is transferred, the remaining 99 are in a clear conflict with the law that states that "the privacy policy should explain which data is transferred to whom". Finally, only 35% of them proposed a mechanism of data deletion.

In summary, trying to enforce privacy according to a single dimension (i.e., law) is ineffective. This is mainly the result of a traditional approach to regulation that did not take into consideration the technological side. Such approach would have been more effective if the privacy policy was machine-readable (e.g., P3P[10]).

1.4.1.2 Privacy By Technology

There are numerous examples of Privacy Preserving Technologies tools, however, very few of them are widely adopted. We argue that the main limit to their adoption is that they do not fit the market demands. To illustrate this, let us take the example of privacy preserving social network (e.g., Safebook [20], PeerSon [21]). In order to protect users' privacy from the service provider itself, these systems adopt a decentralized architecture by relying on the cooperation among users. Moreover, friendship relations are built on top of "real" interactions in order to cope with the problem of building trust among virtual friends. While such systems have been proven to provide strong privacy guarantees, they are far from being able to compete with the top (centralized) players.

This lack of adoption is mainly due to the nature of the market. Big players offer unlimited storing space, very easy and user-friendly interface, billions of connected friends and nearly 100% availability time. The *cost* of moving to a secure service, is in most cases, unbearable as it entails a highly limited storing and uptime, a very limited set of friends and usually little or no support.

In summary, trying to enforce privacy solely through a technical approach is hardly achievable. The proposed solution should not only be privacy friendly but should also suit users' demand.

1.4.2 Hurdles in Privacy Decision

Regardless of how privacy-friendly is the software we are using or how restrictive data privacy laws are, our privacy depends on our own behavior. Hence, our ability to make the right, thoughtful privacy decision is vital, yet it remains a complex problem. The right privacy decision balances between our willingness to share information with others and the protection of our private sphere. In short, it should maximize our welfare while minimizing our future regrets [22]. The hurdles in making such decisions stem from a combination of

[10] `http://www.w3.org/P3P/`

factors [23]: competing or opposing interests — the costs and benefits of sharing or hiding information —, inconsistent privacy preferences (e.g., privacy perception changes with age) and most importantly the incomplete and *asymmetric* information about the risks and consequences of sharing data.

In the last few years, several research projects have proposed solutions to overcome such hurdles, allowing users to easily manage their privacy preferences. Such researches tackled the problem of salient information [24–27], control [28, 29] and expressiveness [30, 31].

However, while empowering users may be a desirable goal, neither salient information, nor control, nor expressiveness, if used separately, can achieve the desired privacy goal. Salient information is usually hard to present and can be ignored [32]. Control, when available, can be hard to tune: A coarse grained settings usually offers little *degree of freedom* and might cause lack of access control. On the other hand, a fine grained system might require several layers of menus/tabs which leads to poor privacy decision[33, 34]. Finally, giving the user the ability to expressively determine their previous privacy setting does not guarantee that their future decisions will not be regretted [23].

1.4.3 Soft Paternalism

Paternalistic policies (i.e., legal approach) adopting a *one size fits all* approach cannot handle the complexity of privacy. Rather than mandating a rule and obliging others to follow it, a more subtle and effective approach have been proposed by behavioral economists to achieve the same goal: *Soft Paternalism*[35, 36].[11] As suggested by its name, this approach combines two notions: (i) the soft expresses that the goal is to *convince or nudge* the user to make the right decision (as opposed to oblige) (ii) the paternalism resides in the definition of what is the right decision. This idea might seem to be an oxymoron as it calls to influence the user behavior while at the same time protecting its freedom of choice. However, "the goal is (only) to steer people's choices in welfare-promoting directions"[36] without dictating a specific approach.

To achieve this goal, behavior economists exploit the user biased perception of privacy and turn it in a way that does not impede user freedom of choice but offer him the option of more informed choices. This concept of soft paternalism or nudging the user towards the right privacy behavior is very appealing for policy makers and technologists. We argue that this approach allows to enhance the privacy as it addresses several dimensions:

- **Technology**: Making the right decision requires an adequate software that can be parametrized accordingly. Nudging creates the need of privacy solutions and makes them more visible to the end-user (e.g., AdBlock is the most used extension in Firefox[12])

- **Markets**: Soft Paternalism can create incentive for privacy-friendly solutions. In fact, the shift in user behavior is likely to trigger competition between companies to support and/or protect privacy (e.g., Google+ privacy circles was highly publicized which pushed Facebook to implement a similar feature).

[11] Also referred to as Libertarian Paternalism, Asymmetric Paternalism or Nudging

[12] `https://addons.mozilla.org/en-US/firefox/addon/adblock-plus/`

- **Norms**: Both users and media pressure would create a need of privacy normalization.

This concept goes beyond a simple parametrization of the system, it aims to shift the user behavior by exposing privacy threats and suggesting solutions. Exposing privacy violations allows the user to make *informed consent* by rebalancing the asymmetric data model. Consider for example, OSN users who post their music interests. This information is not sensitive per se, but could lead to privacy breach through the inference of other sensitive information — as we show in Chapter 3. A *paternalistic* (legal) approach would ban the publication of such interests, which is too restrictive as user might have legitimate reasons to share music interests (e.g., to share play-list with friends). A usability/technological approach would design a system where hiding such information is easy. A soft paternalism approach, would expose the risk behind sharing that seemingly harmless information publicly and provide some solution to mitigate this threat.

The work in this thesis adopts this approach by exposing several worrisome issues regarding privacy in OSNs. Our goal is to raise awareness and trigger this *soft* change towards a more privacy respectful behavior. The rationale behind our approach is to show that current systems are not privacy-friendly as different entities can leverage seemingly harmless information to breach user privacy. By drawing users, media and decision makers attention to these threats, we hope to change the *status quo bias* (i.e., the common human resistance to change one's behavior).

1.5 Contributions

Every day, Facebook collects and processes 2.5 billion pieces of content — representing more than 500 terabytes of data — including 2 billion Like actions and more than 300 million photos.[13] However, this data is only the tip of the iceberg as Facebook and similar OSN providers are not only collecting and mining this information but also exploiting other techniques to further increase user monitoring and tracking. In this thesis, we uncover several mechanisms that can be exploited by OSN operators, malicious attackers (or third parties) or governments to acquire more information about users in the context of OSNs. We expose several scenarios of either *deployed* or *possible* attacks to invade user privacy and describe plausible solutions.

This thesis consists of two parts. The first deals with data publication *within* an OSN and how it can be exploited to breach user privacy. More specifically, we show that attributes that are publicly revealed by users can be used to (i) infer other missing or hidden information and (ii) to enhance password cracking.

- **Inferring Hidden or Private Information about Users [6].** We show how seemingly harmless interests (e.g., music interests) can leak privacy-sensitive information about users. In particular, we infer their undisclosed (private) attributes using the public attributes of other users sharing similar interests. In order to compare user-defined interest names, we extract their semantics using an ontologized version of Wikipedia and measure their similarity by applying a statistical learning method. Besides self-declared interests in music, our technique does not rely on any further

[13] `https://www.facebook.com/data`

information about users such as friend relationships or group belongings. Our experiments, based on more than 104K public profiles collected from Facebook and more than 2000 private profiles provided by volunteers, show that our inference technique efficiently predicts attributes that are very often hidden by users. This study calls for a stricter information sharing decision.

- **Exploiting Public Information to enhance password cracking [37].** In a second step, we show the disastrous consequence of privacy breach on security. Specifically, we demonstrate how user information — gathered from an OSN public profile — can be exploited to enhance the password cracking process. From a security perspective, we propose a novel password cracker based on Markov models, which extends ideas used by Narayanan and Shmatikov [38]. Through extensive experiments, we show that it can guess more than 80% of passwords (at 10 billion guesses), more than all probabilistic password crackers we compared against. From a privacy perspective, we systematically analyze the idea that *additional personal information* about a user helps in speeding up password guessing. We show that the gain can go up to 30% for passwords that are actually based on personal attributes. This work highlights the important risk that might arise from weak, ill informed privacy decision regarding published data. A seemingly benign information can be correlated with user password which ultimately can be exploited to hack into the user account. Our results highlight the need of new methods in password strength estimation which take into consideration user information.

Throughout these studies, it becomes clear that publicly disclosed attributes harm privacy. However, measuring the "loss" resulting from the revelation of some attributes remains an open challenge. Our third contribution tries to answer this fundamental question.

- **A Quantitative Privacy Measure [39].** We provide a practical, yet formally proved, method to estimate the uniqueness of each profile by studying the amount of information carried by public profile attributes. Our first contribution is to leverage the "Ads Audience Estimation" platform of a major OSN (Facebook) to compute the uniqueness of public profiles, independently from the used profile dataset. Then, we measure the quantity of information carried by the revealed attributes and evaluate the potential privacy risk of releasing them publicly. Our measurement results, based on an unbiased sample of more than 400K Facebook public profiles, show that the combination of *gender*, *current city* and *age* can identify close to 55% of users to within a group of 20 and uniquely identify around 18% of them. Our approach can enhance privacy in at least two scenarios: First, it provides a *quantitative privacy measure*: by hiding or revealing attributes a user can measure the amount of information he is disclosing. Second, data providers can quantify the risk of re-identification (i.e., identifying a user in a dataset) when disclosing a set of attributes.

The second part of this thesis analyses privacy threats *outside* the OSN. Specifically, we analyzed how the interaction between the OSN platform and third parties can harm

privacy. The two possible scenarios are analyzed: (i) sensitive information that OSNs gather from third parties (i.e., other websites) and (ii) information that third party can gather from OSNs.

- **Information leakage from first party to OSNs [15].** In this study, we shed light on web user tracking capabilities of the three major global OSNs (i.e., Facebook, Google+ and Twitter). We study mechanisms which enable these services to persistently and accurately follow users' web activity, and evaluate to what extent this phenomenon is spread across the web. Through a study of the top 10K websites, our findings indicate that OSN tracking is diffused among almost all website categories, independently from the content and from the audience. We also evaluate the tracking capabilities in practice and demonstrate by analyzing real traffic traces that OSNs can reconstruct a significant portion of users' web profile and browsing history. We finally provide insights into the relation between the browsing history characteristics and the OSN tracking potential, highlighting the high risk properties.

- **Information leakage from OSN to third parties [11].** We examine third-party OSN applications for two major OSNs: Facebook and RenRen. These third-party applications typically gather, from the OSN, user personal information. We develop a measurement platform to study the interaction between OSN applications and fourth parties. We show that several third party applications are leaking user information to "fourth" party entities such as trackers and advertisers. This behavior affects both Facebook and RenRen with varying severity. 22% of tested Facebook applications are transmitting at least one attribute to an external entity with the user ID being the most prominent (18%). On the other hand, RenRen suffers from a major privacy breach caused by the leakage of the *access token* in 69% of the tested apps. These tokens can be used by trackers and advertisers to impersonate the app and query RenRen on behalf of the user.

1.6 Thesis Organization

This dissertation is structured as follows. In the next chapter, we overview privacy in the context of OSN. First, we identify and classify privacy threats, then we describe proposed solutions and expose their strengths and weaknesses. Finally, we compare our work to previous researches and highlight our main contributions.

In Chapter 3, we propose a novel inference technique that leverages publicly disclosed music interests to infer hidden or missing information about the user. Our approach relies on semantic analysis and machine learning techniques to extract discriminative features for user classification.

Chapter 4 is at the intersection of privacy and security. The first part treats the security challenge of password cracking. Specifically, we propose a new Markov chain password cracker and show through extensive experiments that it outperforms all state of the art password crackers. The second part treats the privacy challenge of integrating user information in the cracking process. We describe a novel approach that exploits user information (e.g.,

name, birthday) in password guessing and show that for some passwords, this additional data can boost the cracking success by up to 30%.

Chapter 5 answers the fundamental question — raised by both previous chapters — of *How private is a public profile?*. We provide a methodology to quantify the uniqueness of a public profile. In the first part, we describe a formal approach to measure the Information Surprisal (IS) carried by each attribute in two scenarios: in the first case, we assume independence between attributes (i.e., the decision to show/hide an attribute i is independent from the same decision for the attribute j) whereas in the second, we allow these attributes to be correlated. We describe our algorithm and show how OSN Advertising Platform can be exploited to achieve this goal.

The reminder of this dissertation examines the privacy implications resulting from the interaction between OSN and third parties. Chapter 6 analyses the tracking capabilities of OSN providers through the so called "share buttons". First, we propose a detailed description of how this tracking is implemented and which users are vulnerable. Then, we evaluate the distribution of these buttons on Alexa's[14] top 10K most visited websites and show that these buttons target a wide range of website categories. Third, we investigate the amount of information that OSNs collect and to which extent they can reconstruct user profiles. Our results, based on real traffic dataset, show that this mechanism is highly efficient and can reconstruct up to 70% of some profiles. Finally, we propose several solutions to reduce the tracking.

Chapter 7 investigates the data leakage from third party OSN applications to external entities. We start by describing the OSN's complex application ecosystem, then describe our measurement platform as well as the data we collected form both Facebook and RenRen. Third, we provide general statistics about applications. The following section presents the information leakage and uncover its mechanisms. Lastly, we provide several solutions and describe their strengths and weaknesses.

Finally, we conclude this dissertation by summarizing our main observations and findings and discuss remaining open questions.

[14] `http://www.alexa.com/topsites`

Chapter 2

Literature review

Contents

As the amount of personal information shared on OSNs is rapidly increasing, so do the privacy concerns. In fact, over the past few years, OSNs experienced an exponential growth which made public a tremendous amount of personal (and sensitive) data. This easily accessible information has brought a plethora of attacks affecting both user privacy and anonymity. This chapter surveys the current state of privacy issues as well as the available defenses. We provide a novel classification of privacy threats with two main categories: *structural inference* and *information leakage*. We acknowledge that this is a brief survey and as such does not *fully* cover the ever-growing area of privacy in OSNs. Our main goal is to categorize each privacy threat and provide some "representative" countermeasures. Interested reader might refer to [40–45] for more details.

2.1 Privacy threat classification

OSNs' users suffer from a wide range of privacy threats ranging from identity disclosure to information leakage. These attacks are diverse as they assume different attacker models, have distinctive goals and lead to various privacy impacts. At a high level, such attacks can either rely on the *graph structure* or solely target user information (see Fig. 2.1). Graph

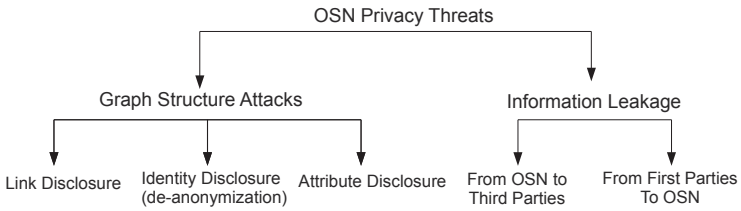


FIGURE 2.1: Categorization of OSN privacy threats

structure attacks aim to infer some graph information such as node attributes (e.g., user age), the existence of a link (e.g., whether two users are friends) or the node identity. We note that these attacks are probabilistic as the adversary output is noisy and hence error-prone. *Information leakage* on the other hand, represents a family of attacks where user information are disclosed either to the OSN (e.g., information about the user navigation pattern is sent to the OSN from an external website) or from the OSN to third parties (e.g., user location is leaked by the OSN). In the following we provide a detailed description of each attack, survey related works and sketch possible solutions.

2.2 Identity disclosure

Identity disclosure refers to the ability of an attacker to map an unknown (possibly "virtual") identity v to a real-world entity p.[1] First, let us define formally the notion of *mapping* in the context of OSNs:

Definition 2.1 (Mapping Query[41])**.** *In a set of online profiles V in a social network G, a mapping query Q finds the profile v that maps to a particular individual p. It returns v if it exists and $\emptyset$ otherwise.*

A first definition of identity disclosure can be expressed as the ability of an attacker to answer a *mapping query* with a full certainty. However, such definition assumes a strong adversary with a background knowledge that uniquely matches p to the corresponding profile in V. In realistic scenarios, such attack is usually hard to achieve. A relaxed definition would introduce uncertainty in the mapping process by modeling the mapping as a distribution over the set of all profiles. More formally, let v_i be a random variable over the set of profiles V and $Pr_p(v_i)$ the probability that the profile i maps to a real entity p. In addition, we introduce a dummy profile v_{dummy} which serves the purpose of absorbing the probability of individual p not having a profile in the network. We assume that p has a unique profile in $V \cup \{v_{dummy}\}$ denoted v_*. Hence, the attacker confidence of mapping v_i to p is therefore $Pr_p(v_i)$. How Pr_p is constructed depends usually on the attacker capabilities (see Section 2.2.1 and 2.2.2). We can now define *identity disclosure* similarly to [41] as:

[1] Real identity represents any Personally Identifiable Information (PII) that can be mapped to a single "physical" identity.

Definition 2.2 (Identity disclosure with a confidence **t**). *In a set of online profiles V in a social network G, identity disclosure with a confidence $\mathbf{t}$ occurs when $Pr_p(v_*) \geq \mathbf{t}$ and $v_* \neq v_{dummy}$*

Identity disclosure represents the "highest" breach to user privacy according to our classification as it allows to identify users. Suppose that an OSN provider releases publicly the friendship graph. To protect the privacy of its users, the provider removes all names and demographic information associated with individual except the gender. This operation is usually referred to as *sanitization*. At first glance, this release seems to be privacy preserving as no personal information can be drown from the released data. However, using the attack described in [46], an adversary can infer the real identity of some nodes in the graph. Moreover, he can exploit the gender data to infer further sensitive information such as the user sexual orientation [47]. Hence, suppressing personal information (i.e., personally identifiable information or PII) is no enough to protect user identity from being disclosed. To this end, more elaborated algorithms aiming at protecting user identity through "anonymization" have been proposed. These protections aim at increasing the attacker uncertainty (i.e., decrease the value of the confidence **t** in Definition 2.2).

2.2.1 Identity disclosure with no external information

In this scenario, the attacker *solely* uses the social graph G without relying on any external information. The rationale behind such attack is that the graph structure can be "finger-printed" which allows the attacker to de-anonymize a targeted user even if the profile attributes have been suppressed.

Such attack can be *passive* or *active*. The former assumes no prior changes (e.g., creating new accounts) to G before it get published while the latter allows the adversary to manipulate G before the anonymization. Both attacks are proposed by Backstrom *et al.* [48] and are based on random graph theory. In the *active* attack scenario, an adversary creates k accounts and links them randomly. The set of targeted account m are then linked in a particular pattern to the set of k accounts to create unique and identifiable subgraphs. After data release, the attacker can efficiently recover the set of k fake nodes and identify the set of m targeted nodes by solving a set of restricted isomorphism problems exploiting the previously created subgraphs. The *passive attack* works similarly by assuming a k colluding nodes which are linked to the m targeted nodes. This linking is also done in particular way that generates "identifiable" subgraphs. At data release, the adversary can identify these subgraphs which allows him to de-anonymize the targeted nodes.

2.2.1.1 Discussion

Active attack demonstrated by Backstrom *et al.* [48] assumes that the adversary is able to modify the graph prior to its release by creating an order of $\mathcal{O}(n \log n)$ Sybil nodes. Carrying out such attack on a large scale is difficult to achieve in practice for multiple reasons. First, most social networks deploy a node admission control mechanism (e.g., requiring the user to link his account with a phone number) which drastically reduces the ability of the attacker to create a large number of accounts. Second, this attack is restricted to online graphs where the adversary is able to create more than one identity: other datasets such as cellphone graphs or location-based social discovery services are

immune as creating Sybil nodes is prohibitively expensive if not impossible. Third, as most legitimate users have no reasons to link back to a Sybil node, an entity with no incoming edges but many outgoing edges will stand out and thus can be easily detected by Sybil defense mechanism such as [49, 50]. Finally, Narayanan *et al.* [46] show that the cut-based attack of [48] which creates a 7-nodes subgraphs containing a Hamiltonian path can easily be detected as only Sybil nodes have this property. Large scale active attacks are unlikely to be feasible as already deployed protection mechanisms significantly reduce the attack effectiveness.

Passive attack is realistic, however, it only breaches the privacy of users that are already connected to either the attacker or the attacker "colluding" friends. Hence, it cannot scale as only users who are "already" friends with malicious nodes can be targeted.

2.2.1.2 Protection mechanism

Two main strategies have been proposed so far to achieve graph anonymization: [2] *k-anonymity* and *generalization* (or clustering). These techniques present the following structure: First, they consider a particular adversary by modeling its *background knowledge* which affects both the privacy guarantee and the effectiveness of the attack. Second, each model is based on a predefined privacy notion (e.g., k-anonymity). Finally, each technique is evaluated based on the utility/privacy trade off that it provides.

k-anonymity: The broader definition of k-anonymity[51] can be expressed as the inability of the attacker to distinguish the record of a user from a $k-1$ other records. In other words, the attacker confidence that a profile v maps to an entity p is no more than $\frac{1}{k}$.

Generalization: The main idea is to group similar users in a single entity called *super-node*. The output of this anonimyzation algorithm is a simplified graph where each group of user is substituted by its super-node. Note that oppositely to k-anonymity, this model does not provide any guarantee on the group size (i.e., the value of k).

We provide here a quick overview of the main graph anonymization techniques. Interested reader may refer to [40, 41, 52] for more details.

Each node has structural properties (i.e., sub-graph signature) that are the same as the ones of a small set of other nodes in the graph, called a candidate set for this node[53]. As this signature is very likely to be unique, it can be used to de-anonymize the node. One example of such structures is the immediate one-hop neighbors (i.e., the node degree). Numerous research papers adopt a modified version of k-anonymity specially tailored to the graph structure. Based on the first hop degree, the notion of *k-degree anonymity* [52] have been proposed to protect against an attacker with the knowledge of the node degree of the users. This algorithm states that each node should have the same degree as at least $k-1$ other nodes in the anonymized network. Formally, let $G(V,E)$ be a graph and $\mathbf{d}_g$ the *degree sequence of* G. That is, $\mathbf{d}_g$ is a vector of size $n = |V|$ such that $\mathbf{d}_g(i)$ is the degree of the i-th node in G. First, let us define k-anonymity for a vector:

[2] Assuming that the adversary background knowledge can be modeled. A third model with no assumption about adversarial knowledge will be presented in Section 2.2.2.

Definition 2.3. *A vector of integers* $\mathbf{v}$ *is k-anonymous, if every distinct value in* $\mathbf{v}$ *appears at least k times.*

For example, vector $\mathbf{v} = [8,8,8,5,5,2,2,2,2]$ is 2-anonymous. We can now define first hop degree k-anonymity for a graph such as:

Definition 2.4. *A graph* $G(V,E)$ *is k-degree anonymous if the degree sequence of G,* $\boldsymbol{d}_g$ *, is k-anonymous.*

To achieve this, authors propose two approaches [52]: a first dynamic-programming algorithm (DP) that solves the problem optimally but has a complexity of $\mathcal{O}(n^2)$, the second algorithm is greedy with a linear complexity of $\mathcal{O}(nk)$.

Zhou and Pei [54] extend the attacker knowledge by assuming that he also knows the link between the one hop neighbors of the targeted node u and refer to it as $Neighbor_G(u)$. To protect the data from such attacker, the anonymized graph should satisfy the stricter notion of *k-neighborhood anonymity* which states that each node in the graph should have its $Neighbor_G$ graph isomorphic to the $Neighbor_G$ graph of at least $k-1$ other nodes.

Generalizing the k-anonimity to 2-hops neighborhood leads to the most general privacy preservation definitions: *k-candidate anonymity* [53] and *k-automorphism* [55]. The latter assumes a very strong adversary with the knowledge of any subgraph signature of a target individual. In 2011, Zhou and Pei [56] proposed an algorithm that goes beyond k-anonymity and achieves the stricter privacy notion of l-diversity. By both adding and removing edges, the anonymized graph achieves both k-anonymity and l-diversity.

Other utility metrics and graph structures have also been considered. Ying and Wu [57] propose two approaches to achieve graph anonymization while minimizing the disruption of the spectral graph properties. Cormode *et al.* [58] introduce a new technique called k,l-grouping to anonymize bipartie graphs.

Bhagat *et al.* [59] present a generalization technique to anonymize OSN data by grouping the entities into classes, substituting each class by a "super-node" and finally masking the mapping between entities. The released graph is composed of the super-nodes and the link between them.

2.2.2 Identity disclosure leveraging external information

Another family of attacks relies on external information to help the adversary de-anonymize the targeted social network. Wondracek *et al.* [60] exploit group membership to create a "membership fingerprint" and de-anonymize the user. It is a two steps attack: First the attacker crawls all groups of the targeted OSN and creates a *membership directory (MD)*. Note that the identity of each user in this directory is known. The de-anonymizing step consists in stealing the victim v history — when visiting the attacker website for example — and extracting the social groups that she belongs to (referred to as grp_v). Note that only v history is known but not its identity. The attacker can now use the *MD* to map v to its identity by checking which user in *MD* has (approximately) the same set of links grp_v. The attack success depends on the entropy of the "membership fingerprint" which have been experimentally shown to have enough entropy to identify a significant proportion of users. In particular, authors show that for 42.06% of Xing[3] OSN users participating in at

[3]XING is a social network for professionals.

least one group (more than 753K users) this fingerprint is unique which implies that the user can be uniquely identified.

Narayanan and Shmatikov [46] were the first to propose a re-identification algorithm targeting anonymized social network graphs. The algorithm relies *solely* on the network structure (i.e., node degree), and demonstrates the feasibility of large-scale, passive de-anonymization of real-world social graphs. The authors show that a third of the users who had accounts in both Flickr and Twitter could be re-identified in the anonymous Twitter graph with only 12% error. The same algorithm — with only minor modifications– have been successfully applied to de-anonymize a partial crawl of Flickr during the Kaggle Social Network Challenge [61].

2.2.2.1 Discussion

Narayanan's[46] attack is the most severe since it can be carried out on a large scale and with a high accuracy. It raises worrisome privacy threats as it does not assume any access to the graph before its release nor any Sybil nodes. It shows that by *solely* relying on the graph topology and assuming that the target graph is properly anonymized (i.e., removing all user information), an attacker is still able to re-identify a large fraction of users by using an external source of information. This technique answers a fundamental question about graph *isomorphism* by showing that for real graphs an algorithm can *easily and efficiently* finds an approximate mapping between two instances (i.e., answering whether two graphs are isomorphic). In fact, graph isomorphism is believed to be a hard problem especially that the best know algorithm for resolving it (in a general scenario) is from Babai and Lucks [62] with a complexity of $\exp O(\sqrt{n \log n})$. However, Narayanan's[46] work shows that mapping two real graphs (e.g., social graphs) is easier and can be done efficiently.

The take home messages from Narayanan's algorithm are not only technical, as they proof, once again, a fundamental aspect of privacy: privacy guarantees are dynamics. For years, graph isomorphism was thought to be intractable — which is the case for general graphs — however, Narayanan's algorithm proved that for social graphs, this observation does not hold allowing large-scale de-anonymization.

2.2.2.2 Protection mechanism

Narayanan's[46] attack relies on external information. In such scenario, k-anonymity falls short as it is impossible to model *arbitrary* external information that an attacker might use to de-anonymize the graph. The quest for a formal model preventing such attack with a quantified privacy bound gave birth to *Differential Privacy*. This model proposed by Dwork [63] frames the privacy in terms of *indistinguishability*: the result of a differentially private algorithm $\mathscr{A}$ for a dataset S_i containing a user i is statistically very close to (and hence "indistinguishable" from) the output of the same algorithm $\mathscr{A}$ for a dataset $S_{\bar{i}}$ where the user i is not part of the dataset. Formally:

Definition 2.5 (Differential Privacy)**.** *A randomized algorithm $\mathscr{A}$ is ε-differentially private if for all datasets D_1 and D_2 that differ on a single element (i.e., data of single user), and all $S \subseteq \text{Range}(\mathscr{A})$,*

$$\Pr[\mathscr{A}(D_1) \in S] \leq e^{\varepsilon} \times \Pr[\mathscr{A}(D_2) \in S]$$

where the probability is taken over the coins of the algorithm and Range($\mathscr{A}$) *denotes the output range of the algorithm* $\mathscr{A}$

This approach shifts the privacy problem from modeling and/or comparing prior and posterior beliefs about individuals before and after data publication to bounding the risk of joining the database. Note that differential privacy is a condition on the release mechanism (i.e., the sanitization algorithm $\mathscr{A}$) and not on the dataset. This privacy definition provides not only guarantees that are *independent* of the background information but also does not bound the adversary computational power.

However, achieving differential privacy is challenging in the context of OSNs. This difficulty stems from the fact that graph structure, when converted to the traditional tabular dataset representation (i.e., each edge has a "source" and "destination" attribute), requires an excessive amount of noise to fulfill the privacy requirements [64]. Hence, naive approaches adopting anonymizing algorithms from statistical databases are impractical as the generated graph has no utility. Therefore, specific approaches have been recently proposed to achieve differential privacy for some of the graph properties such as the degree distribution [65], the joint degree distribution (and assortativity) [66], triangle counting [67] and some generalization of triangles [68] or any combination of these [69]. Based on these statistics, the algorithm outputs a synthetic graph respecting these properties.

2.3 Link disclosure

Link disclosure occurs when an attacker is able to guess whether two entities in the OSN are connected. Links can be categorized in two classes: *social* and *affiliate*. The former defines a link between two individuals (e.g., friendship) while the latter expresses the affiliation of an individual to a group (e.g., the user is a member of a band). However, in most cases, link prediction algorithms do not make any assumption about the link category. Link prediction task is often considered as a binary classification problem: for any two — potentially linked – entities o_i and o_j, predict whether $l_{i,j}$ is 1 or 0. This task is not trivial as most linked data set are highly sparse. As pointed out in many studies [70–72], one of the main difficulties of building statistical models for such a classifier is that the prior probability (i.e., the probability that two entities are linked) is usually quite small. This causes several difficulties, in particular, quantifying the level of confidence in the prediction.

One possible approach to link prediction is to *solely* rely on the graph structure e.g., graph proximity measures [42] or to exploit the attribute information e.g., the structured logistic regression proposed by Popscul *et al.* [73]. Prediction can also be made collectively by defining a probabilistic model over the set of links, attributes and edges (e.g., Markov Random Fields (MRF)). In its simplest scenario, the set of entities O with their corresponding attributes X and links E can be modeled by a MRF describing the joint distribution over the set of edges $P(E)$, or a distribution conditioned on the node' attributes $P(E|X)$. Direct graph models can also be applied such as the algorithm described by Getoor *et al.* [74] for handling link uncertainty in probabilistic relational models. Finally, Yin *et al.* [75] propose a random walk based algorithm that integrates network structure and node attributes to perform both link prediction and attribute inference.

2.3.0.3 Discussion

Link inference attacks are usually seen as a stepping stone towards identity de-anonymization and, as such, are rarely the final goal of a privacy breach. For instance, Wondracek *et al.* [60] show that by identifying the groups a victim belongs to, an attacker can identify its victim (See Section 2.2.2 for the attack description). Moreover, we note that for scenarios like the one presented by Zheleva and Getoor [8] where an attacker exploits the social links to infer user private attributes or the *navigation attack* [76] where an adversary navigates through the OSN to uncover a hidden profile are ineffective when social links are partially hidden, link prediction can then be used to overcome such issues.

2.3.0.4 Protection mechanism

From a user perspective, the first protection is to minimize the amount of information he shares. While this measure does not completely solve the problem, it is a necessary step towards the solution. Friends should also behave similarly as most OSNs do not provide any mechanism to prevent "friends" from sharing their social links with others, breaching their friends privacy. Friendship relations can also be exposed by the social network itself through a variety of other means (e.g., mail searching capabilities, friends suggestion etc.).

In the data release scenario, Zheleva *et al.* [77] propose both *randomization* and *aggregation* algorithm to protect users social links. The former randomly removes some existent links while the latter groups multiple links into a single class. Note that all approaches presented in section 2.2.1.2 (i.e., k-anonymity and generalization) and 2.2.2.2 (i.e., differential privacy) can also be used to mitigate link disclosure attacks.

2.4 Attribute disclosure

While users tend to share a tremendous amount of information about themselves, a significant amount of this information is supposed to be private or shared with a small circle of friends. Attribute disclosure refers to privacy breach where an attacker infers such private information. Attributes can be categorized based on their observability: an *observable* attribute is one that the user sets in his profile (e.g., age). Oppositely, a *latent* attribute refers to an attribute that is not part of the profile and as such cannot be explicitly set by the user (e.g., user mood or sentiment). To infer missing or hidden attributes several strategies are possible.

Homophily. A first natural approach borrowed from social science is based on homophily [78] which states that linked users tend to share similar attributes suggesting that network structure should inform attribute inference. Furthermore, other evidence [79, 80] show that user sharing the same attributes are likely to be linked. Zheleva and Getoor [8] were the first to study the impact of friends' attributes on the privacy of a user. They try to infer private user attributes based on the groups users belong to. For that purpose, they compare the inference accuracy of different link-based classification algorithms (e.g., Friend-aggregate model, Collective classification model, Groupmate-link model etc.). In [81], authors build a Bayes network from links extracted from a social network. Although they crawl a real OSN (LiveJournal), they use hypothetical attributes to analyze their

learning algorithm. A further step was taken by [82] who proposes a modified Naive Bayes classifier that infers political affiliation (i.e., a binary value: liberal or conservative) based on user attributes, user links or both. Becker and Chen [83] infer many different attributes of Facebook users, including affiliation, age, country, degree of education using the most popular attribute values of the user's friends.

User attributes. Some publicly available attributes are correlated with the missing ones. An adversary can learn these correlations e.g., by using machine learning techniques. In [84, 85] authors use supervised learning and text processing techniques to extract information from both user name and user generated text to infer ethnicity and political orientation. Burger *et al.* [86] construct a large multilingual dataset of tweets with the corresponding user gender. Based on this dataset, they explore several different classifiers to infer user gender and show that the classifier accuracy depends on tweet volumes as well as whether or not various kinds of profile metadata are included in the models. Pennacchiotti *et al.* [87] use a Gradient Boosted Decision Trees to learn class of labels (e.g., political affiliation) for each tweet which is then used to infer the user profile (political affiliation detection, ethnicity identification and detecting affinity for a particular business). Dey *et al.* [88] leverage the college/university information to infer the user age.

Virtual Graph. Rather than relying on self declared or existing graphs, an attacker can build a virtual graph where *similar* users are linked together. Such similarity might be the affiliation to a group (e.g., same university), similar taste (e.g., like the same music), graph based metric (e.g., conductance). Note that this similarity metric is a parameter which depends both on the targeted OSN and the target attributes. For instance, Mislove *et al.* [7] build a "virtual" communities based on a metric called *Normalized conductance*. Using two Facebook datasets of both Rice and New Orleans universities, they infer few attributes such as college, matriculation year and department. While they achieve good performance for both department and school, the result for the matriculation year is much worse. This behavior have been reported by Amanda *et al.* [89] who explain that community-based inference is data dependent. In fact, through an in depth study of community detection algorithms for social networks and after comparing the results of 100 different social graphs (provided by Facebook), Amanda *et al.* [89] conclude that a common attribute of a community is a good predictor only in small set of social graphs and cannot be generalized.

Latent attribute inference. The main goal of latent attribute inference algorithms is to derive some unknown information about the user. The main difference with previous approaches is that there is (usually) no ground truth to confirm the inferred data. Latent information can be the user taste (music, films, books), user "mood" (e.g., happy, sad) or any other variable that describes the user (psychological) state.

One possible approach to infer user mood or opinion relies on *sentiment analysis*. In this scenario, the system takes as input a sentence and outputs the user mood or opinion. While a wide range of moods can be captured from a sentence, most research studies focus on *polarity*: inferring whether the message is positive, negative or neutral. Such analysis has numerous applications especially for real time monitoring of public opinions (e.g.,

predicting the stock Market based on users mood [90]). Sentiment analysis approaches can be classified in two groups: machine learning and lexical based.

Machine learning mostly relies on supervised techniques [91–94] where the problem is expressed as binary classification task (i.e., positive or negative). The learning phase of the algorithm requires a *labelled* dataset to train the classifier. Various learning algorithms have been tested such as Support Vector Machines (SVMs), Naive Bayes, and Maximum Entropy classifiers, using a variety of features, such as unigrams and bigrams, part-of-speech tags and term frequency [91]. The main advantages of learning models is their adaptability to a specific context. However, the scarceness of labelled data impacts both accuracy and the applicability of the learned model to new data.

Lexical based approaches make use of predefined lists of words or dictionary where each word is associated with its emotional weight. The created models depend on the used dictionary. For instance, "Linguistic Inquiry and Word Count" (LIWC) [95] or "General Inquirer" (GI) [96] are usually used in the context of formally written English, while PANAS-t [97] and POMS-ex [98] are used in the context of psychometric. While lexical based techniques do not rely on labelled data and do not require a training phase, it is hard to have a specific dictionary for each specific situation (e.g., dictionary for slang).

Gonçalves *et al.* [99] compare 8 different sentiment analysis algorithms — both machine learning and lexical— in terms of coverage (i.e., the fraction of messages whose sentiment is identified) and agreement (i.e., the number of message that are correctly classified). Their results show that these algorithms have a varying degrees of coverage (ranging from 4% to 95%) as well as agreement (ranging from 33% to 80%). From a privacy perspective, their newly proposed algorithm — a mixture of the 8 others — achieves a coverage of 95% with and F-measure of 70%. Using such approach, an attacker can learn the user opinion and mood with very high accuracy. He can also monitor his victim mood fluctuation [100], opinion about a product or even the victim willingness to buy it.

2.4.0.5 Discussion

Homophily based inference techniques represent the "natural" approach to infer user data as they exploit the main data in OSN: friendship information. While these algorithms have been proven to perform well, they suffer from at least two shortcomings. First, they are usually OSN dependent as the semantic of friendship depends on the OSN (e.g., friends in Facebook are different from those in LinkedIn). We note that this behavior has been reported by Zheleva and Getoor [8] as they admit that their approach is not suitable for Facebook especially with multi-valued attributes such as political views. Second, these algorithms rely on attributes that are (publicly) revealed by friends, and hence, heavily depend on the amount of revealed information. For instance, Zheleva and Getoor [8] make the assumption that at least *50%* of a user's friends reveal the private attribute. However, as we will show in Chapter 3, this is not realistic, since users tend to massively hide their information from public access. For instance, only 18% of users on Facebook disclose their relationship status and less than 2% disclose their birth date. Hence, the performance of such approaches in today's scenario is questionable.

Attribute based inference techniques are powerful but affects only few attributes. In fact, most previous works targeted either the user gender or the political affiliation. The

performance of such algorithms depends on the amount of publicly revealed information which has been proven to be quickly diminishing [6, 101]. However, the fast progress in both text mining and entity recognition might revive such attacks.

Virtual graph based inference and latent attribute inference are severe attacks for several reasons. First, they usually exploit information that is seemingly *harmless* such as the music interests or user name. Second, they are highly modular and hence can be easily extended (e.g., using other interests). Finally, their machine learning building-blocks are continuously improved.

2.4.0.6 Protection mechanism

The aforementioned privacy leaks stem from the lack of content control. An extensive body of research is dealing with such protection ranging from providing access control to user resources independently from the specific OSN to more stronger fine grained access control where even the OSN provider is prevented from accessing the data.

Tootoonchian *et al.* tackle image privacy problem by providing a Firefox extension called *ImageLock* [102] that substitutes the original picture by a fake and stores the former in a trusted server (under the user control). When authorized users request the picture, ImageLock downloads it from the trusted server and replace the fake one. An improved version was proposed few years later [103, 104] and encompasses *any* type of updated content. Similarly to the first version, the updated content is stored in a trusted server along with an access control list that specifies which user (i.e., social relation) can have access to the content. A public-key-based cryptographic credentials called *social attestations* are created and shared among users to specify the kind of relation between users. The main drawback of both approaches is that users still have to trust a third party (i.e., the server where the data is stored).

To overcome the unnecessary trusted third party , Luo *et al.* propose *FaceCloak* [105], an application that substitutes the users personal information with fake data while storing the genuine information, encrypted, on a third party server. The fake data is then used as index to decrypt the genuine information as long as the user knows the corresponding mapping (the key).

Lucas and Borisov present *flyByNight* [106], a platform for content protection using encryption. All operations are executed through a Facebook application as follows: the application is used to update user profile and define the set of users who can access this update. The information is then encrypted using the (set) receiver public key. Proxy re-encryption [107] is used to reduce the storing and communication overhead in case of one to many communication. This approach, however, has at least one drawback: it is insecure against active attacks carried out by the OSN provider as it has access to the encryption keys.

Guha *et al.* design None Of Your Business (NOYB) [108], a tool to encrypt the user information and prevent the OSN provider from reading and even detecting the encryption. As traditional encryption schema are easily detectable (e.g., by checking the entropy), the authors propose a novel substitution encryption. NOYB partitions the users' uploaded text into atoms. These atoms are then pseudo randomly substituted by other NOYB's users atoms. Each substitution is performed based on a dictionary, which is indexed by the

encrypted genuine atoms. Therefore, by decrypting the index (if the user has the proper key to decrypt it) the user obtains the genuine atom.

Attribute based encryption (ABE) [109] is the basis of many content protection approaches handling multiple users. In a nutshell, each generated key has a single or a set of attributes, for which it can both cipher and decipher the content. In the OSN case, an attribute can represent a category of users such as "relatives", "co-workers" or "close friends". Each social link has then an associated key with the corresponding attribute, the user (i.e., data owner) can then encrypt the information according to the group of users with whom he wishes to share it. *Persona*, proposed by Baden *et al.* [110], is an instantiation of such approach.

Beato *et al.* proposed a Firefox extension called *Scramble!* [111] which, through access control list, helps the user to enforce access control over his data. Both data integrity and confidentiality are enforced using hybrid-encryption (OpenPGP). Diaspora,[4] an open source social network, also uses OpenPGP to ensure data confidentiality. However, it relies on its own distributed infrastructure and, as such, does not support the existing (and highly popular) centralized OSNs.

2.5 Information leakage

Web 2.0 have brought a new concept: connecting people in a rich participatory network. Leveraging OSNs and federated identity platform, Internet migrated from a ready only (or fetch only) media to a write/contribute one. While this evolution brought countless advantages, it also rises several privacy issues as social interactions imply sharing personal, and usually sensitive, information. They also complexify the interactions which heavily reduces, or even impedes, any control over the flow of transmitted data. Data leakage happens when user personal information is transferred, without the user consent, from a first party website (i.e., the website user is browsing) to third parties. This information is usually referred to as "Personally Identifiable Information" (PII) and stands for any piece of data that can distinguish or trace an individual's identity either alone or when combined with other information that is linkable to a specific individual [16]. Information leakage in the context of social networks can be categorized in two classes: (i) PII leaked from OSN platform to external websites, and (ii) website leaking PII to OSNs.

2.5.1 Information leakage from OSN to external websites

Krishnamurthy and Wills were the first to study the potential leakage of PII from OSNs to third party entities. In their work [16], they show that PII are indeed leaked from OSN platforms to third parties through a combination of HTTP header information (i.e., Referer and URI) and cookies. They identify four types of PII leakage. First, the *transmission of the OSN identifier to third party servers from the OSN*: they showed that out of the 12 studied OSNs, 9 transferred the user identifier to third parties through the Referer, 5 through the request URI, and 2 through cookies. In all, 11 out of the 12 OSNs are leaking the user identifier to third parties. The second class considered the *transmission of the OSN identifier to third-party servers via popular external applications* and was not deeply

[4] https://diasporafoundation.org

investigated. In fact, a single example of the "ilike" Facebook application was given to prove the existence of such leakage. The third class dealt with the *transmission of specific pieces of PII to third-party servers*: results showed that only 2 OSNs directly leaked pieces of PII to third parties via the Request URI, Referer header and cookies. Recently, Xia *et al.* [112] present *Tessellation* a framework to correlate user identity –using OSN ID extracted from the social network traffic – to its online behavior.

Protection Mechanism To restrict apps access to user data, multiple solutions have been proposed. Egele *et al.* [113] proposed a client-side proxy that executes in the user's web browser, which makes requests for private data explicit to user and allows her to exert fine-grained access control. Multiple solutions adopt a similar (proxy) based design [114]. While such solutions reduce the leakage, they cannot prevent the transmission of user information to third party. Felt *et al.* [115] propose an alternative platform design that prevents third parties from seeing real user data. *The privacy-by-proxy approach* imposes a radical change as applications display information to users using special markup tags that are (then) substituted by the real value (i.e., real user attribute) by the proxy.

2.5.2 Information leakage from first party to OSNs

While at first glance information leakage from first parties to OSNs seems impossible, there exists a growing phenomena that can be considered as information leakage: *tracking*. In fact, while OSNs are freely accessible, the viability of their business model relies on collecting users' personal data for targeted advertising and in some cases data monetization [116]. Hence, there is a huge incentive for OSNs to gather as much information as possible as a mean to build accurate advertising profiles. In this quest of the perfect profile, user browsing history represents a highly valuable data, and tracking, in general, is more pervasive than ever [3]. Nevertheless, and to the best of our knowledge, a single study [117] analyses this privacy threat of social tracking. To bridge this gap, our work [15] (see Chapter 6) studies the tracking capabilities of OSNs and the potential data leakage it can cause.

Protection Mechanism This technique is a cookie-based tracking mechanism, solutions such as cookies deletion after each session or private mode browsing can be a first approach to mitigate the threat. However these countermeasure suffer from at least two drawbacks. First, navigation patterns belonging to one session are still tracked. Second, more aggressive tracking techniques (e.g., through IP address and browser fingerprint [118]) can still be applied. A more suitable solution consists in blocking the OSN "iframes" connections, and as such all connections to the OSN servers. Tools like Ghostry [119] implement such a mechanism as a browser Plugin.

2.6 Positioning

In this section, we sketch the main differences between our work and the aforementioned literature.

Attribute Inference [6]. We infer users' undisclosed (private) attributes using the public attributes of other users sharing similar interests. Our approach creates a *virtual graph* based on users' music taste. It contrast with previous works as: First, it only relies on users' interests and does not exploit any link information (i.e., homophily) . Second, it uses a (large) public dataset rather than relying on a private dataset as for [7, 8] and hence does not make any (unrealistic) assumption about the availability of data. Finally, it assume a different attacker model than [7, 8] as our attacker has only access to the public profile.

Exploiting Public Information to enhance password cracking [37]. Our work presents two main contributions. First, we propose a novel password cracker based on Markov models, which extends ideas used by Narayanan and Shmatikov [38]. Our algorithm outputs passwords in decreasing probability order which allows it to outperform all password crackers we compared against. Our second contribution analyses how *additional personal information* about a user can help in speeding up password guessing. Through a formal analysis, we provide a method to integrate such data and assess experimentally its effectiveness. We show that the gain is significant and can go up to 30% for passwords that are actually based on personal attributes. To the best of our knowledge, we are the first to systematically study the relationship between chosen passwords and users' personal information in the context of password cracking.

User Re-identification [39]. This work addresses the important goal of quantifying the uniqueness of a public profile by measuring the information carried by each (public) attributes. Sweeney [120] was the first to show that seemingly coarse-grained information such as birth date or ZIP code, if combined, can uniquely identify their owners. Following studies such as [121] emphasize the same finding that: "few characteristics are needed to uniquely identify an individual." Our work complements these by proposing a way to measure the uniqueness of every public profile in a large OSN (e.g., Facebook). Moreover, these studies differ from ours in at least two aspects. First, studied datasets are released by a third-party (e.g., government) who decides which attributes to disclose for all users. This implies "a one rule fits all" approach, where all users are subject to the same privacy policy. Our work considers a dataset where each user has significant control over the revealed data. Second, the targeted populations are significantly different, i.e. US census data versus our world-wide and online population. Hence, our work can be viewed as a new technique to quantify user anonymity in a dynamic environment (such as OSNs), where both self-selected and crowd-driven privacy policies impact user anonymity. Moreover, while our approach does not explicitly address user re-identification and record linkage, it can be leveraged to assess the feasibility of such attacks. Specifically, the magnitude of the information surprisal value of a user's profile (in the attacked dataset) can be directly related to the level of user's vulnerability to linkage. Our approach relies on *entropy* which has been used at various levels: to measure the fingerprint size of a web browser [122], of a host [118] or to track users across multiple services based on their usernames [123].

Information leakage from first party to OSNs [15]. Despite the fact that tracking user activities represents a major threat to user privacy, still it is widely used. The most

comparable work to ours is [3] where the authors study web activity tracking by third parties. Our work focuses on web activity tracking by OSNs, which was not considered in [3], and presents a comprehensive analysis of tracking by the major OSN service providers. Authors in [117] describe the mechanism deployed by Facebook to track user navigation. The paper focuses on the legal issues of tracking and provides a succinct technical description of the mechanism. Our work extends their research results by including an in-depth description of how the tracking mechanism is deployed by the three major OSNs, analyzing the presence of OSN tracking in Alexa top 10K web sites and providing tracking statistics based on real traffic measurements.

Information leakage from OSN to third parties [11]. We examine third-party (OSN) applications for two major OSNs: Facebook and RenRen by developing a measurement platform to study the interaction between OSN applications and fourth parties. While OSN privacy have been extensively studied [16, 124], few works dealt with social apps. The first analysis of Facebook apps permissions is reported by Chia *et al.* [13] where they show that community rating is not reliable and that most applications are requesting more permissions than needed. A fine grained analysis of 150 Facebook apps permissions by Felt *et al.* [115] reveals that only 9% of the evaluated applications need to access personal data to work correctly. Frank *et al.* [12] extend this work and show that Facebook permissions follow a predefined pattern and that malicious application deviate from it. Finally, Xia *et al.* propose *Tessellation* [112] a framework to correlate user identity – using various OSN identifiers extracted from the social network traffic – to its online behavior. Our approach is complementary as it shows that OSN identifiers can be also extracted from other sources (i.e., traffic between third party applications and external entities). None of these works examine the flow of personal information from the third-party apps to fourth-party entities. Our work, to our knowledge, is the first to explore indirect privacy leakage to external entities via third-party applications.

Conclusion

In this chapter we provided a systematic analysis of privacy threats in OSNs. Through a novel classification, we presented a detailed description of each threat, described several attacking scenarios and present some available solutions. We then highlight the main differences between our work and related works.

Part I

Privacy Threats within OSNs

Chapter 3

Information Leakage Through Users' Interests

Contents

3.1 Introduction

Among the vast amount of personal information, user interests or *likes* (using the terminology of Facebook) is one of the highly-available public information on OSNs. Our measurements show that 57% of about half million Facebook user profiles that we collected *publicly* reveal at least one interest among different categories. This wealth of information shows that the majority of users consider this information harmless to their privacy as they do not see any correlation between their interests and their private data. Nonetheless, interests, if augmented with semantic knowledge, may leak information about its owner and thus lead to privacy breach. For example, consider an unknown Facebook user who has an interest "Eenie Meenie". In addition, there are many female teenager users who have interests such as "My World 2.0" and "Justin Bieber". It is easy to predict that the unknown user is probably also a female teenager: "Eenie Meenie" is a song of "Justin Bieber" on his album "My World 2.0", and most Facebook users who have these interests are female teenagers. This example illustrates the two main components of our approach: (1) deriving *semantic correlation* between words (e.g., "My World 2.0", "Eenie Meenie", and "Justin Bieber") in order to link users sharing similar interests, and (2) deriving statistics about these users (e.g., Justin Bieber fans) by analyzing their public Facebook profiles. To the best of our knowledge, the possibility of this information leakage and the automation of such inference have never been considered so far. We believe that this lack of exploitation is due to several challenges to extract useful information from interest names and descriptions.

First, many interests are *ambiguous*. In fact, they are short sentences (or even one word) that deal with a concept. Without a semantic definition of this concept, the interest is equivocal. For example, if a user includes "My World 2.0" in her Art/Entertainment interests, one can imply that this user is likely to be interested in pop as a genre of music. Without a knowledge of what "My World 2.0" is, the information about such an interest is hidden, and hence unexploited.

Second, drawing *semantic link* between different interests is difficult. For example, if a user includes in her public profile "My World 2.0" and another user chooses the interest "I love Justin Bieber", then clearly, these two users are among the Justin Bieber funs. However, at a large scale, automating interest linkage may not be possible without semantic knowledge.

Finally, interests are *user-generated*, and as such, very *heterogeneous* items as opposed to marketers' classified items (e.g., in Amazon, Imdb, etc.). This is due to the fact that OSNs do not have any control on how the descriptions and titles of interests are constructed. As a result, interest descriptions as provided by users are often incorrect, misleading or altogether missing. It is therefore very difficult to extract useful information from interests and classify them from the user-generated descriptions. Particularly, interests that are harvested from user profiles are different in nature, ranging from official homepage links or ad-hoc created groups to user instantaneous input. In addition, interest descriptions, as shown on public profiles, either have coarse granularity (i.e., high level descriptions of classes of interests such as "Music", "Movies", "Books", etc.), or they are too fine-grained to be exploited (e.g., referring to the name of a singer/music band, or to the title of a recent movie, etc.). Finding a source of knowledge encompassing this huge variety of concepts is

challenging.

Therefore, *linking users sharing semantically related interests* is the pivot of our approach. The main goal of our work is to show how seemingly harmless information such as interests, if augmented with semantic knowledge, can leak private information. As a demonstration, we will show that *solely based* on what users reveal as their music interests, we can successfully infer hidden information with more than 70% of correct guesses for some attributes in Facebook. Furthermore, as opposed to previous works [7, 8, 82], our technique *does not need further information*, such as friend relationships or group belongings.

Technical Roadmap

Our objective is to find out interest similarities between users, even though these similarities might not be clearly observed from their interests. We extract semantic links between their seemingly unrelated interest names using a semantic-based generative model called Latent Dirichlet Allocation (LDA) [125]. The idea behind LDA is to learn the underlying (semantic) relationship between different interests, and classify them into groups (called Interest Topics). The output of LDA is the probabilities that an interest name belongs to each of these topics.

To identify latent (semantic) relations between different interests, LDA needs a broader semantic description of each interest than simply their short names. For instance, LDA cannot reveal semantic relations between interests “Eenie Meenie” and “My World 2.0” using only these names unless they are augmented with some text describing their semantics. Informally, we create a document about “Eenie Meenie” and another about “My World 2.0” that contain their semantic description and then let LDA identify the common topics of these documents. These documents are called Interest Descriptions. In order to draw semantic knowledge from the vast corpus of users’ interests, we leverage on the ontologized version of Wikipedia. An interest description, according to our Wikipedia usage, is the parent categories of the most likely article that describes the interest. These represent broader topics organizing this interest. For instance, there is a single Wikipedia article about “Eenie Meenie” which belongs to category “Justin Bieber songs” (among others). In addition, there is another article about “My World 2.0” that belongs to category “Justin Bieber albums". Therefore, the descriptions of interests “Eenie Meenie" and “My World 2.0" will contain “Justin Bieber songs" and “Justin Bieber albums", respectively, and LDA can create a topic representing Justin Bieber which connects the two interests. An interesting feature of this method is the ability to enrich the user’s interests from, say a single item, to a collection of related categories, and hence draw a broader picture of the semantics behind the interest of the user. We used two sets of 104K public Facebook profiles and 2000 private profiles to derive the topics of all the collected interests.

Knowing each user’s interests and the probabilities that these interests belong to the identified topics, we compute the likelihood that users are interested in these topics. Our intuition is that users who are interested roughly in the same topics with “similar" likelihood (called interest neighbors) have also similar personal profile data. Hence, to infer a specific user’s hidden attribute in his profile, we identify his interest neighbors who publicly reveal this attribute in their profile. Then, we guess the hidden value from the neighbors’ (public)

attribute values.

We postulate and verify that interest-based similarities between users, and in particular their music preferences, is a good predictor of hidden information. As long as users are revealing their music interests, we show that sensitive attributes such as Gender, Age, Relationship status and Country-level locations can be inferred with high accuracy.

Organization We describe our attacker model in Section 3.2. Section 3.3 presents related work in the area of personalizing retrievals. Our algorithm is detailed in Section 3.4 and both of our datasets are described in Section 3.5. Section 3.6 is devoted to present our inference results. We motivate the usage of each building block of our algorithm and discuss possible alternatives in Section 3.7 and finally we conclude.

3.2 Attacker Model

Before defining our attacker model, we describe user profiles as implemented by Facebook.

Facebook implements a user profile as a collection of personal data called attributes, which describe the user. These attributes can be binary such as Gender or multi-valued such as Age. The availability of these attributes obeys to a set of privacy-settings rules. Depending on these privacy settings, which are set by the user, information can be revealed exclusively to the social links established on OSN (e.g., friends in Facebook[1]) or partially (e.g., to friends of friends) or publicly (i.e., to everyone). In this work, we demonstrate the information leakage through users' interests by inferring the private attributes of a user. We consider two binary attributes (Gender: male/female and Relationship status: married/single) and two multi-valued attributes (Country-level location and Age).

As opposed to previous works [8, 82], we consider an attacker that *only* has access to self-declared, publicly available music interests. Hence, the attacker can be *anyone* who can collect the Facebook public profile of a targeted user. In fact, earlier attacks considered a dataset crawled from a specific community such as a university. Hence, the crawler being part of this community had access to attributes that are only visible to friends which impacts data availability. Indeed, as we will show in Section 3.5, data availability is different whether we deal with public data (data disclosed to anyone) or private data (data disclosed to friends only). Thus our attacker is more general compared to [8, 82], since it relies only on public information.

This characteristic allows us to draw a broader attacker. For example, our technique can be used for the purpose of profiling to deliver targeted ads. Advertisers could automatically build user profiles with high accuracy and minimum effort, with or *without* the consent of the users. Spammers could gather information across the web to send extremely targeted spam (e.g., by including specific information related to the location or age of the targeted user).

[1]Facebook has recently added a feature to split friends into sub-lists in order to make some attributes accessible to a chosen subset of friends.

3.3 Related Work

In the following we briefly summarize related work in the area of Personalizing Retrieval as we use techniques from this field in order to link users sharing similar interests. Inferring private attributes and de-anonymizing users are two privacy problems related to this work, however, we do not detail them here as they have already been treated in Chapter 2 in Section 2.4 and Section 2.2 respectively.

Our work shares techniques with the area of personalizing retrieval where the goal is to build personalized services to users. This can be derived from the user "taste" or by interpreting his social interactions. This is an active research domain and a broad range of problems were resolved and used in e-commerce, recommendation, collaborative filtering and similar. This knowledge extraction entails the analysis of a large text corpora from which one can derive a statistical model that explains latent interactions between the documents. Latent semantic analysis techniques provide an efficient way to extract underlying topics and cluster documents [126, 127]. Latent Dirichlet Allocation (LDA) [125] has been extended by Zhang *et al.* [128] to identify communities in the Orkut social network. The model was successfully used to recommend new groups to users. In addition, Zheleva *et al.* [129] used an adapted LDA model to derive music taste from listening activities of users in order to identify songs related to a specific taste and the listeners who share the same taste.

Similarly to these works, we also use LDA to capture the interest topics of users but instead of recommending content, our goal is to link users sharing *semantically-related* interests to demonstrate information leakage.

3.4 From Interest Names to Attribute Inference

3.4.1 Overview

While a human can easily capture the semantics behind different interest names (titles or short descriptions), this task cannot be easily automated. In this section, we present how we can extract meaningful knowledge from users' interests and then classify them for the purpose of attribute inference.

Our technique consists of four main steps as illustrated by Figure 3.1:

1. Creating Interest Descriptions: Interest descriptions are the user-specified interest names augmented with semantically related words which are mined from the Wikipedia ontology.

2. Extracting semantic correlation between interest descriptions using Latent Dirichlet Allocation (LDA). The output represents a set of topics containing semantically related concepts.

3. Computing Interest Feature Vectors (IFV). Based on the discovered topics, LDA also computes the probability that an interest I belongs to $Topic_i$ for all I and i (Step 3a). Then, we derive the IFV of each user (Step 3b) which quantifies the interest of a user in each topic.

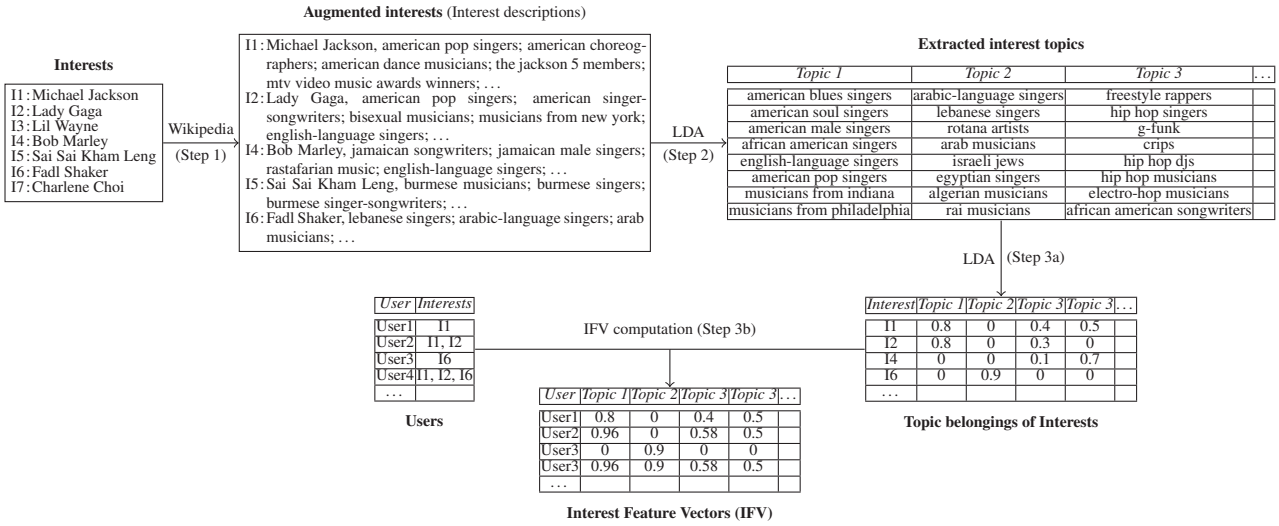


FIGURE 3.1: Computing interest feature vectors. First, we extract interest names and augment them using Wikipedia (Step 1). Then, we compute correlation between augmented interests and generate topics (Step 2) using LDA. Finally, we compute the IFV of each user (Step 3).

4. Computing the neighbors of each user in the feature space (i.e., whose IFVs are similar in the feature space) to discover similar users, and exploiting this neighborhood to infer hidden attributes.

3.4.2 Step 1: Augmenting Interests

Interest names (shortly interests) extracted from user profiles can be single words, phrases, and also complex sentences. These text fragments are usually insufficient to characterize the interest *topics* of the user. Indeed, most statistical learning methods, such as LDA, need a deeper description of a given document (i.e., interest) in order to identify the semantic correlation inside a text corpora (i.e., set of interests). Moreover, the diversity and heterogeneity of these interests make their description a difficult task. For instance, two different interests such as “AC/DC” and “I love Angus Young" refer to the same band. However, these strings on their own provide insufficient information to reveal this semantic correlation. To augment interest names with further content that helps LDA to identify their common topics, we use an ontology, which provides structured knowledge about any unstructured fragment of text (i.e., interest names).

3.4.2.1 Wikipedia as an Ontology

Although there are several available ontologies [130, 131], we use the ontologized version of Wikipedia, the most up-to-date and largest reference of human knowledge in the world. Wikipedia represents a huge, constantly evolving collection of manually defined concepts and semantic relations, which are sufficient to cover most interest names. Moreover, Wikipedia is multilingual which allows the augmentation of non-english interest names. We used the Wikipedia Miner Toolkit [132] to create the ontologized version of

Wikipedia from a dump made on January, 2011 with a size of 27 Gb.

Wikipedia includes *articles* and *categories*. Each article describes a single concept or topic, and almost all Wikipedia's articles are organized within one or more categories, which can be mined for broader (more general) semantic meaning. AC/DC, for example, belongs to the categories Australian hard rock musical groups, Hard rock musical groups, Blues-rock groups etc. All of Wikipedia's categories descend from a single root called *Fundamental*. The distance between a particular category and this root measures the category's generality or specificity. For instance, AC/DC is in depth 5, while its parent categories are in depth 4 which means they are more general and closer to the root.
All articles contain various hyper links pointing to further (semantically related) articles. For example, the article about Angus Young contains links to articles AC/DC, musician, duckwalk, etc. The anchor texts used within these links have particular importance as they can help with disambiguation and eventually identifying the most related article to a given search term: e.g., if majority of the "duckwalk" links (i.e., their anchor texts contain string "duckwalk") is pointing to Chuck Berry and only a few of them to the bird Duck, then with high probability the search term "duckwalk" refers to Chuck Berry (a dancing style performed by Chuck Berry). Indeed, the toolkit uses this approach to search for the most related article to a search string; first, the anchor texts of the links made to an article are used to index all articles. Then, the article which has the most links containing the search term as the anchor text is defined to be the most related article.

3.4.2.2 Interest Description

The description of an interest is the collection of the parent categories of its most related Wikipedia article (more precisely, the collection of the name of these categories). To create such descriptions, we first searched for the Wikipedia article that is most related to a given interest name using the toolkit. The search vocabulary is extensive (5 million or more terms and phrases), and encodes both synonymy and polysemy. The search returns an article or set of articles that could refer to the given interest. If a list is returned, we select the article that is most likely related to the interest name as described above. Afterwards, we gather all the parent categories of the most related article which constitute the description of the interest. For example, in Figure 3.1, *User3* has interest "Fadl Shaker". Searching for "Fadl Shaker" in Wikipedia, we obtain a single article which has parent categories "Arab musicians", "Arabic-language singers" and "Lebanese male singers". These strings altogether (with "Fadl Shaker") give the description of this interest.

3.4.3 Step 2: Extracting Semantic Correlation

To identify semantic correlations between interest descriptions, we use Latent Dirichlet Allocation (LDA) [125]. LDA captures statistical properties of text documents in a discrete dataset and represents each document in terms of the underlying topics. More specifically, having a text corpora consisting of N documents (i.e., N interest descriptions), each document is modeled as a mixture of latent topics (interest topics). A topic represents a cluster of words that tend to co-occur with a high probability within the topic. For example, in Figure 3.1, "American soul singers" and "American blues singers" often co-occur and thus belong to the same topic ($Topic_1$). However, we do not expect to find "Arab musicians"

in the same context, and thus, it belongs to another topic ($Topic_2$). Note that the topics are created by LDA and they are not named. Through characterizing the statistical relations among words and documents, LDA can estimate the probability that a given document is about a given topic where the number of all topics is denoted by k and is a parameter of the LDA model.

More precisely, LDA models our collection of interest descriptions as follows. The topics of an interest description are described by a discrete (i.e., categorical) random variable $\mathscr{M}(\phi)$ with parameter ϕ which is in turn drawn from a Dirichlet distribution $\mathscr{D}(\alpha)$ for each description, where both ϕ and α are parameter vectors with a size of k. In addition, each topic z out of the k has a discrete distribution $\mathscr{M}(\beta_z)$ on the whole vocabulary. The generative process for each interest description has the following steps:

1. Sample ϕ from $\mathscr{D}(\alpha)$.
2. For each word w_i of the description:
   (a) Sample a topic z_i from $\mathscr{M}(\phi)$.
   (b) Sample a word w_i from $\mathscr{M}(\beta_{z_i})$.

Note that α and $B = \cup_z\{\beta_z\}$ are corpus-level parameters, while ϕ is a document-level parameter (i.e., it is sampled once for each interest description). Given the parameters α and B, the joint probability distribution of an interest topic mixture ϕ, a set of words W, and a set of k topics Z for a description is

$$p(\phi, Z, W|\alpha, B) = p(\phi|\alpha)\prod_{\forall i} p(z_i|\phi)p(w_i|\beta_{z_i}) \tag{3.1}$$

The observable variable is W (i.e., the set of words in the interest descriptions) while α, B, and ϕ are latent variables. Equation (3.1) describes a parametric empirical Bayes model, where we can estimate the parameters using Bayes inference. In this work, we used collapsed Gibbs sampling [133] to recover the posterior marginal distribution of ϕ for each interest description. Recall that ϕ is a vector, i.e., ϕ_i is the probability that the interest description belongs to $Topic_i$.

3.4.4 Step 3: Interest Feature Vector (IFV) Extraction

The probability that a user is interested in $Topic_i$ is the probability that his interest descriptions belong to $Topic_i$. Let V denote a user's interest feature vector, $\mathbb{I}$ is the set of his interest descriptions, and ϕ_i^I is the probability that interest description I belongs to $Topic_i$. Then, for all $1 \leq i \leq k$,

$$V_i = 1 - \prod_{\forall I \in \mathbb{I}} (1 - \phi_i^I)$$

is the probability that the user is interested in $Topic_i$.

For instance, in Figure 3.1, *User4* has interests "Lady Gaga", "Michael Jackson", and "Fadl Shaker". The probability that *User4* belongs to $Topic_1$, which represents American singers, is the probability that at least one of these interests belongs to $Topic_1$. This equals $1 - ((1-0.8)(1-0.8)) = 0.96$.

3.4.5 Step 4: Inference

3.4.5.1 Neighbors Computation

Observe that an IFV uniquely defines the interest of an individual in a k-dimensional feature space. Defining an appropriate distance measure in this space, we can quantify the similarity between the interests of any two users. This allows the identification of users who share similar interests, and likely have correlated profile data that can be used to infer their hidden profile data.

We use a chi-squared distance metric. In particular, the correlation distance $d_{V,W}$ between two IFV vectors V and W is

$$d_{V,W} = \sum_{i=1}^{k} \frac{(V_i - W_i)^2}{(V_i + W_i)}$$

In [134], authors showed that the chi-squared distance gives better results when dealing with vectors of probabilities than others. Indeed, we conducted several tests with different other distance metrics: Euclidean, Manhattan and Kullback-Leibler, and results show that the chi-squared distance outperforms all of them.

Using the above metric, we can compute the ℓ nearest neighbors of a user u (i.e., the users who are the closest to u in the interest feature space). A naive approach is to compute all $M^2/2$ pairwise distances, where M is the number of all users, and then to find the ℓ closest ones for each user. However, it becomes impractical for large values of M and k. A more efficient approach using k-d tree is taken. The main motivation behind k-d trees is that the tree can be constructed efficiently (with complexity $O(M \log_2 M)$), then saved and used afterwards to compute the closest neighbor of any user with a worst case computation of $O(k \cdot M^{1-1/k})$.

3.4.5.2 Inference

We can infer a user's hidden profile attribute x from that of its ℓ nearest neighbors: first, we select the ℓ nearest neighbors out of all whose attribute x is defined and public. Then, we do *majority voting* for the hidden value (i.e., we select the attribute value which the most users out of the ℓ nearest neighbor have). If more than one attribute value has the maximal number of votes, we randomly choose one.

For instance, suppose that we want to infer *User4*'s country-level location in Figure 3.1, and *User4* has 5 nearest neighbors (who publish their locations) because all of them are interested in $Topic_2$ with high probability (e.g., they like "Fadl Shaker"). If 3 out of these 5 are from Egypt and the others are from Lebanon then our guess for *User4*'s location is Egypt.

Although there are multiple techniques besides majority voting to derive the hidden attribute value, we will show in Section 3.6.2 that, surprisingly, even this simple technique results in remarkable inference accuracy.

3.5 Dataset Description

For the purpose of our study, we collected two profile datasets from Facebook. The first is composed of Facebook profiles that we crawled and which we accessed as "everyone"

(see Section 3.5.1). The second is a set of more than 4000 private profiles that we collected from volunteers using a Facebook application (see Section 3.5.2). Next, we describe our methodology used to collect these datasets. We also present the technical challenges that we encountered while crawling Facebook. Finally, we describe the characteristics of our datasets.

3.5.1 Crawling Public Facebook Profiles

Crawling a social network is challenging due to several reasons. One main concern is to avoid sampling biases. A previous work [135] has shown that the best approach to avoid sampling bias is a so called True Uniform Sampling (UNI) of user identifiers (ID). UNI consists in generating a random 32-bits ID and then crawling the corresponding user profile in Facebook. This technique has a major drawback in practice: most of the generated IDs are likely to be unassigned, and thus not associated with any profile (only 16% of the 32-bits space is used). Hence, the crawler would quickly become very resource-consuming because a large number of requests would be unsuccessful. In our case, inspired by the conclusions in [135], and avoiding sampling bias that might be introduced by different social graph crawls (e.g., Breadth-First Search), we follow a simple, yet efficient two-steps crawling methodology as an alternative to UNI.

First, we randomly crawled a large fraction of the Facebook Public directory[3]. As a result, a total of 100 Million (and 120 thousands) *URL*s of searchable Facebook profiles were collected (without profile data). This technique allows to avoid the random generation of user identifiers while uniformly (independently from the social graph properties) collecting *existing* user identifiers.

Second, from this list of candidate URLs of profiles, we crawled a set of randomly selected 494 392 profiles out of the 100 millions. The crawled dataset is called RawProfiles.

Finally, the entire RawProfiles dataset was sanitized to fit our validation purposes. Two restrictions were considered: (1) non Latin-written profiles were filtered out from the dataset and (2) only profiles with at least one music interest with its corresponding Wikipedia description were kept. Therefore, we obtained a set of 104 401 profiles. This data set, called PubProfiles, is then used as an input of our inference algorithm (see details in Section 3.4.4).

Technical challenges

As noted above, we crawled profiles to collect public information that are available to everyone. However, Facebook, as most OSNs operators do, protects this data from exhaustive crawling by implementing a plethora of anti-crawler techniques. For instance, it implements a request rate limit that, if exceeded, generates a CAPTCHA to be solved. To bypass this restriction and to be cautious not to DoS the system, we set a very slow request frequency (1 per minute). In addition, we distributed our crawler on 6 different machines that were geographically spread. In addition, it is worth noting that one of the trickiest countermeasures that Facebook implements to prevent easy crawling is the rendering of the web page. In particular, rather than sending a simple HTML page to the client browser, Facebook embeds HTML inside JavaScript, thus, the received page is not a

[3]available at:`http://www.facebook.com/directory/`

valid HTML page but a JavaScript code that has to be interpreted. Unfortunately, most publicly available crawling libraries do not interpret JavaScript. Thus, we developed our own lightweight web browser, based on the Qt Port of WebKit [136], which is capable of interpreting JavaScript. This allows our crawler to be served with easy-to-parse HTML page.

3.5.2 A Facebook Application to Collect Private Attributes

We developed a Facebook application to gather private attributes from users. The application was distributed to many of our colleagues and friends on Facebook, and was surprisingly used by more users than expected. Users volunteered to install the application, and hence, their private information was collected by our tool. We collected private attributes from 4012 profiles out of which 2458 profiles have at least one music interest. These anonymized private profiles, collected from April 6 to April 20 in 2011, represent our private dataset (called VolunteerProfiles).

The usage of this dataset is motivated by our need to understand how data availability varies between public and private datasets, and to verify whether it impacts the results of our algorithm.

3.5.3 Ethical and Legal Considerations

In order to comply with legal and ethical aspects in crawling OSNs data, we were cautious not to inadvertently DoS the Facebook infrastructure (as mentioned in Section 3.5.1). Also cautionary measures were taken to prevent our crawler from requesting off-limit information. In other words, our crawler is compliant with the Robots Exclusion Protocol [137]. Even though we accessed publicly available information, we anonymized the collected data by removing user names and all information which were irrelevant to our study.

The Facebook application needed more sanitization to ensure users' anonymity.

3.5.4 Dataset Description

In the following, we provide statistics that describe the datasets used in this study. First, Table 3.1 summarizes the statistics about the availability of attributes in the three datasets (i.e., in RawProfiles, PubProfiles and VolunteerProfiles).

Attributes	$Raw(\%)$	$Pub(\%)$	$Volunteer(\%)$
Gender	79	84	96
Interests	57	100	62
Current City	23	29	48
Looking For	22	34	-
Home Town	22	31	48
Relationship	17	24	43
Interested In	16	26	-
Birth date	6	11	72
Religion	1	2	0

TABLE 3.1: The availability of attributes in our datasets.

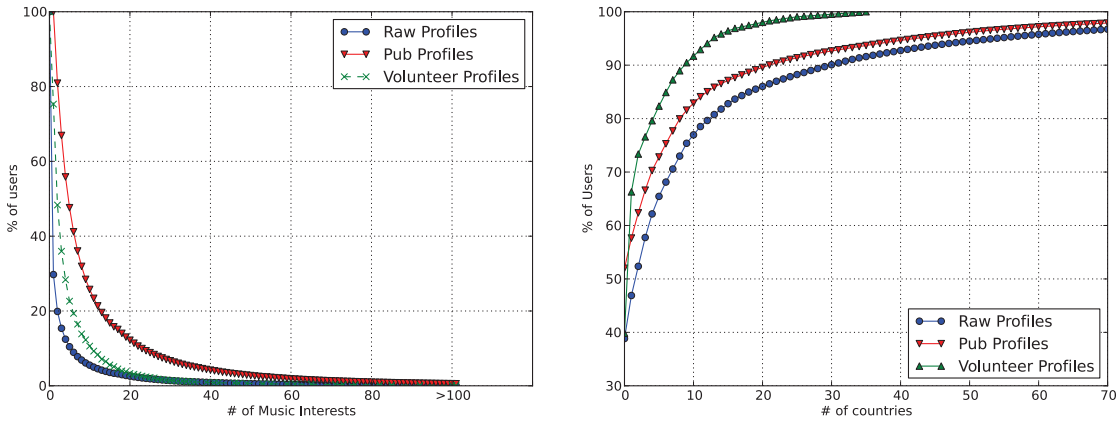


FIGURE 3.2: Left: Complementary Cumulative Distribution Function of Music Interests. Right: Cumulative Distribution Function (CDF) of Country-level Locations (retrieved from the *CurrentCity* attribute)

We observe that Gender is the most common attribute that users publicly reveal. However, three attributes that we want to infer are largely kept private. The age is the information that users conceal the most (89% are undisclosed in PubProfiles). Comparing the availability of the attributes in PubProfiles and VolunteerProfiles is enlightening. We can clearly note that users tend to hide their attribute values from public access even though these attributes are frequently provided (in their private profiles). For instance, the birth date is provided in more than 72% in VolunteerProfiles, whereas it is rarely available in PubProfiles (only 1.62% of users provide their full birth date). The current city is publicly revealed in almost 30% of the cases, whereas half of all volunteers provided this data in their private profile. Recall that the attributes we are interested in are either binary (Gender, Relationship) or multi-valued (Age, Country-level location). Finally, note that, as it is shown in Table 3.1, the public availability of attributes in PubProfiles and in RawProfiles are roughly similar.

Also note that the availability of interests slightly changes from RawProfiles (57%) to VolunteerProfiles (62%), yet still relatively abundant. This behavior might have at least two explanations: (1) by default, Facebook sets Interest to be a public attribute, (2) users are more willing to reveal their interests compared to other attributes. Figure 3.2 (left) depicts the complementary CDF of music interests publicly revealed by users in the three datasets. Note that more than 30% of RawProfiles profiles reveal at least one music interest. Private profiles show a higher ratio which is more than 75%.

3.6 Experimentation Results and Validation

In the following, we validate our interest-based inference technique using both VolunteerProfiles and PubProfiles. We evaluated the correctness of our algorithm in terms of inference accuracy, i.e. the fraction of successful inferences and the total number of trials. An inference

[3]Using random guessing instead of maximum likelihood decision

Attribute	*Overall marginal distribution (OMD)*		*Inference accuracy on* VolunteerProfiles	
	PubProfiles	*Facebook statistics*	PubProfiles OMD	*Facebook statistics OMD*
Gender	62% (Female)	51% (Male)	39.3%	60.7%
Relationship	55% (Single)	Unknown	36.7%	50%[3]
Age	50% (18-25)	26.1% (26-34)	33.9%	57.9%
Country	52% (U.S)	23% (U.S)	2.3%	2.3%

TABLE 3.2: Baseline inference using different marginal distributions. Inference of VolunteerProfiles based on Facebook OMD is better than PubProfiles OMD.

is successful if the inferred attribute equals to the real value. In particular, for both PubProfiles and VolunteerProfiles datasets and for each attribute to be inferred, we select users that provide the attribute and then we compute the inference accuracy: we hide each user's attribute, compute the nearest neighbors of the user, do a majority voting as described in Section 3.4.5.1, and then verify whether the inference yields the real attribute.

Before discussing our validation results, in the following, we introduce a maximum likelihood-based inference technique that we consider as a baseline technique with which we compare our method.

3.6.1 Baseline Inference Technique

Without having access to any friendship and/or community graph, an adversary can rely on the marginal distributions of the attribute values. In particular, the probability of value val of a hidden attribute x in any user's profile u can be estimated as the fraction of users who have this attribute value in dataset U:

$$P(u.x = \mathsf{val}|U) = \frac{|\{v \,|\, v.x = \mathsf{val} \wedge v \in U\}|}{|U|}$$

Then, a simple approach to infer an attribute is to guess its most likely value for all users (i.e., the value x for which $P(u.x = \mathsf{val}|U)$ is maximal).

To compute $P(u.x = \mathsf{val}|U)$, an adversary can crawl a set of users and then derive the Overall Marginal Distribution (OMD) of an attribute x from the crawled dataset (more preciselly, U is the subset of all crawled users who published that attribute). However, this OMD is derived from public attributes (i.e., U contains only publicly revealed attributes), and hence, may deviate from the real OMD which includes both publicly revealed and undisclosed attributes.

To illustrate the difference, consider Table 3.2 that compares the real OMD of the four attributes to be inferred, as provided by Facebook statistics (composed of both private and public attributes [138]), with the OMD derived from our public dataset PubProfiles. The two distributions suggest different predominant values which highly impacts the inference accuracy when the guess is based on the most likely attribute value. For instance, PubProfiles conveys that the majority of Facebook users are female which contradicts Facebook statistics (with a significant difference of 11%). Similarly, the age of most users according to PubProfiles is between 18 and 25-years old, while the predominant category of ages, according to Facebook, is 26-34.

In fact, all public datasets (e.g., PubProfiles) are biased towards the availability of attributes (not to be confused with the bias in sampling discussed in Section 3.5.1). Recall that, as shown in Table 3.1, some attributes (in particular Age, Relationship status and Country) are publicly available for only a small fraction of users (see the PubProfiles column). Put simply, the difference between the two OMDs is mainly due to the mixture of private and public attributes in Facebook statistics and the absence of private attributes in PubProfiles. Whether revealing attributes is driven by some sociological reasons or others is beyond the scope of this work.

To illustrate how the bias towards attribute availability impacts inference accuracy, we conduct two experiments. First, we infer the attributes in VolunteerProfiles using the OMD derived from PubProfiles. In the second experiment, we infer the same attributes using the OMD computed from Facebook statistics. As shown in Table 3.2, the second approach always performs better. The results show that using the Facebook statistics we obtain an inference accuracy gain of 21% for the gender and 25% for the age. Since Facebook does not provide statistics about the relationship status of their users, we used random guessing instead (i.e., we randomly chose between single and married for each user). Surprisingly, even random guessing outperforms the maximum likelihood-based approach using PubProfiles OMD. Therefore, we conclude that the maximum likelihood-based inference performs better when we use the OMD derived from Facebook statistics. Accordingly, in our performance evaluation, we also used this in our baseline inference technique.

Finally, note that previous works [8, 82] computed the inference accuracy using private data (i.e., their dataset is a crawl of a community, and thus, they could access all attributes that can only be seen by community members). Hence, these results are obtained with different attacker model, and the assumption that 50% of all attributes are accessible, as suggested in [8], is unrealistic in our model.

3.6.2 Experiments

In order to validate our interest-based inference technique, we follow two approaches. First, for each attribute, we randomly sample users from PubProfiles such that the sampled dataset has the same OMD as the real Facebook dataset [138]. Then, we measure the inference accuracy on this sampled dataset. Second, we test our technique on the VolunteerProfiles dataset where both private and public attributes are known. Since we know the attribute values in the collected profiles, we can check if the inference is successful or not. In particular, we infer four attributes in both approaches: Gender, Relationship status, Age, and the Country of current location. We run experiments to infer an attribute a in PubProfiles as follows:

1. From all users that provide a in PubProfiles, we randomly sample a set of users (denoted by S) following the OMD of Facebook. The size of S for each attribute is tailored by (i) Facebook OMD and (ii) the number of available samples in PubProfiles. Table 3.3 shows the size of S.

2. For this sampled set, we compute the inference accuracy as it has been described in Section 3.6.2.

3. We repeat Steps 2 and 3 fifteen times and compute the average of all inference accuracy values (Monte Carlo experiment).

For VolunteerProfiles we proceed as for PubProfiles, but since the attributes are a mix of public and private attributes, there is no need to do sampling, and we skip Step 1.

Attribute	Size of S
Gender	1000
Relationship	400
Country	1000
Age	105

TABLE 3.3: Size of the Randomly Sampled Set of Users S

Parameter Estimation Recall from Section 3.4.5.1 that our algorithm is based on majority voting. Hence, estimating the number of neighbors that provides the best inference accuracy for each attribute is essential. Figure 3.3 depicts the inference accuracy in function of the number of neighbors. This figure clearly shows that each attribute has a specific number of neighbors that results in the best inference accuracy. Note that, as discussed at the beginning of this section, we rely on repeated random sampling to compute the results, and hence, the computed parameters are independent from the input data. Age inference requires two neighbors. This can be explained by the limited number of users that disclose their age which causes the IFV space to be very sparse: the more neighbors we consider the more likely it is that these neighbors are far and have different attribute values. For other attributes, the optimal number of neighbors is between 3 and 5. We tested different IFV sizes (i.e., k the number of topics). Notably, best results were achieved with $k = 100$. In the sequel, we will use these estimated numbers of neighbors as well as $k = 100$ which yield the best inference accuracy.

Table 3.4 provides a summary of the results for PubProfiles. The information leakage can be estimated to 20% in comparison with the baseline inference. Surprisingly, the amount of the information is independent from the inferred attribute since the gain is about 20% for all of them. These results show that music interest is a good predictor of all attributes.

Gender Inference Table 3.4 shows that the gender can be inferred with a high accuracy even if only one music interest is known in the PubProfiles. Our algorithm performs 18%

Attribute	Baseline	Random guess	IFV Inference
Gender	51%	50%	69%
Relationship	50%	50%	71%
Country	41%	10%	60%
Age	26%	16.6%	49%

TABLE 3.4: Inference Accuracy of PubProfiles

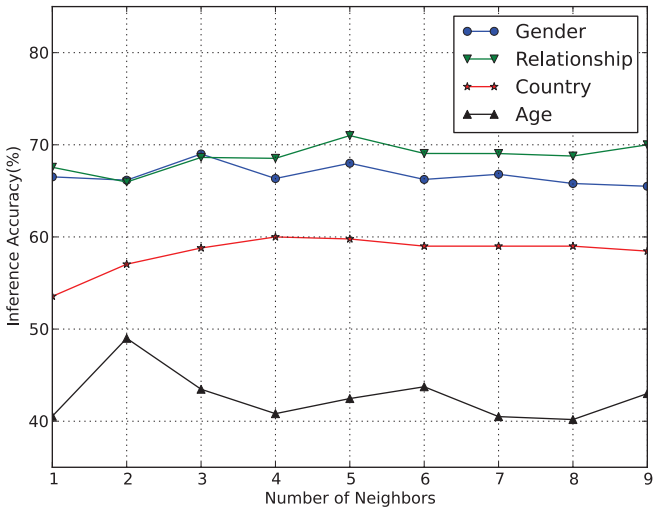


FIGURE 3.3: Correlation between Number of Neighbors and Inference Accuracy

Attribute \ Inferred	Male	Female
Male	53%	47%
Female	14%	86%

TABLE 3.5: Confusion Matrix of Gender

better than the baseline. Recall that the baseline guesses male for all users (Table 3.2). To compare the inference accuracy for both males and females, we computed the confusion matrix in Table 3.5. Surprisingly, Female inference is highly accurate (86%) with a low false negative rate (14%). However, it is not the case for male inference. This behavior can be explained by the number of female profiles in our dataset. In fact, females represent 61.41% of all collected profiles (with publicly revealed gender attribute) and they were subscribed to 421685 music interests. However, males share only 273714 music interests which represents 35% less than woman. Hence, our technique is more capable of predicting females since the amount of their disclosed (music) interest information is larger compared to males. This also confirms that the amount of disclosed interest information is correlated with inference accuracy.

Attribute \ Inferred	Single	Married
Single	78%	22%
Married	36%	64%

TABLE 3.6: Confusion Matrix of Relationship

Relationship Inference Inferring the relationship status (married/single) is challenging since less than 17% of crawled users disclose this attribute showing that it is highly sensitive. Recall that, as there is no publicly available statistics about the distribution of this attribute, we do random guessing as the baseline (having an accuracy of 50%). Our algorithm performs well with 71% of good inference for all users in PubProfiles. As previously, we investigate how music interests are a good predictor for both single and married users by computing the confusion matrix (Table 3.6). We notice that single users are more distinguishable, based on their IFV, than married ones. The explanation is that single users share more interests than married ones. In particular, a single user has an average of 9 music interests whereas a married user has only 5.79. Likewise in case of gender, this confirms that the amount of disclosed interest information is correlated with inference accuracy.

Country of Location Inference As described in Section 3.5, we are interested in inferring the users' location in the top 10 countries in Facebook. Our approach can easily be extended to all countries, however, as shown by Figure 3.2, more than 80% of users in PubProfiles belong to 10 countries and these countries represent more than 55% of all Facebook users according to [138]. As the number of users belonging to the top 10 countries is very limited in VolunteerProfiles, we do not evaluate our scheme on that dataset. Table 3.4 shows that our algorithm has an accuracy of 60% on PubProfiles with 19% increase compared to the baseline (recall that, following Table 3.2, the baseline gives U.S. as a guess for all users). Figure 3.4 draws the confusion matrix[5] and gives more insight about the inference accuracy. In fact, countries with a specific (regional) music have better accuracy than others. Particularly, U.S. has more than 94% of correct inference, Philippine 80%, India 62%, Indonesia 58% and Greece 42%. This highlights the essence of our algorithm where semantically correlated music interests are grouped together and hence allow us to extract users interested in the same topics (e.g., Philippine music). Without a semantic knowledge that specifies the origin of a singer or band this is not possible. As for Gender and relationship, the number of collected profiles can also explain the incapacity of the system to correctly infer certain countries such as Italy, Mexico or France. In particular, as shown in Table 3.7, the number of users belonging to these countries is very small. Hence, their interests may be insufficient to compute a representative IFV which yields poor accuracy.

Age Inference Finally, we are interested in inferring the age of users. To do that, we created four age categories[6] that are depicted in Table 3.8. Recall that the baseline technique always predicts the category of 26 and 34 years for all users. Table 3.4 shows that our algorithm performs 23% better than the baseline attack. Note that our technique gives good results despite that only 3% of all users provide their age (3133 users in total) in PubProfiles. We investigate how music interests are correlated with the age bin by computing the confusion matrix in Table 3.8. We find that, as expected, most errors come

[5] We removed Brazil since all its entries (2) were wrongly inferred. This is caused by the small number of Brazilians in our dataset.

[6] We created six categories but since in PubProfiles we have only few users in the last 3 bins, we merge them together. For VolunteerProfiles we have six bins.

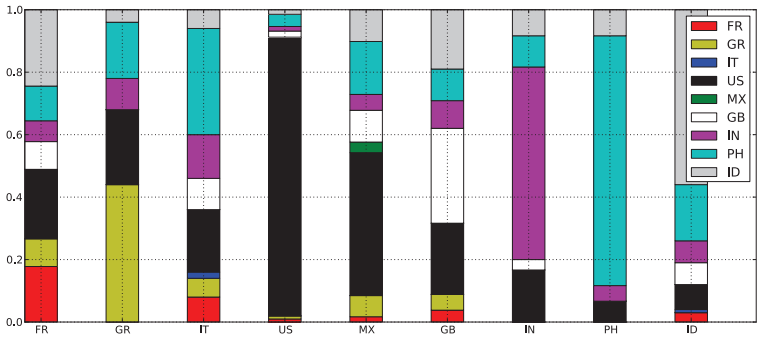


FIGURE 3.4: Confusion Matrix of Country Inference

Country	% of users
US	71.9%
PH	7.80%
IN	6.21%
ID	5.08%
GB	3.62%
GR	2.32%
FR	2.12%
MX	0.41%
IT	0.40%
BR	0.01%

TABLE 3.7: Top 10 countries distribution in PubProfiles

Att \ Inferred	13-17	18-24	25-34	35+
13-17	58.33%	30%	11.6%	0%
18-24	17%	67%	3.4%	1.3%
25-34	15.38%	46.15%	38.4%	0%
35+	0%	100%	0%	0%

TABLE 3.8: Confusion Matrix of Age Inference

from falsely putting users into their neighboring bins. For instance, our method puts 30% of 13-18 years old users into the bin of 18-24 years. However, note that fewer bins (such as teenager, adult and senior) would yield better accuracy, and it should be sufficient to many applications (e.g., for targeted ads). Observe that we have an error of 100% for the last bin. This is due to the small number of users (3 in PubProfiles) who belong to this bin (we cannot extract useful information and build a representative IFV for such a small number of users).

3.6.2.1 VolunteerProfiles **Inference**

Attribute	Baseline	Random guess	IFV Inference
Gender	51%	50%	72.5%
Relationship	50%	50%	70.5%
Age	26%	16.6%	42%

TABLE 3.9: Inference Accuracy for VolunteerProfiles

As a second step to validate our IFV technique, we perform inference on VolunteerProfiles. Table 3.9 shows that our algorithm also performs well on this dataset. Notice that Age

inference is slightly worse than in PubProfiles. Recall from Section 3.6.2 that we had only a few users in the last three bins in PubProfiles, and hence, we merged these bins. However, in VolunteerProfiles, we have enough users and we can have 6 different age categories. This explains the small difference in inference accuracy between VolunteerProfiles and PubProfiles. Regarding other attributes, the accuracy is slightly worse for Relationship (-0.5%) and better for Gender (+3.5%). This small variation in inference accuracy between PubProfiles and VolunteerProfiles demonstrates that our technique has also good results with users having *private* attributes: in PubProfiles, we could compute the inference accuracy only on users who published their attribute values, while in VolunteerProfiles, we could also test our method on users hiding their attributes.

3.7 Discussion

Topic modeling We used LDA for semantic extraction. Another alternative is to use Latent Semantic Analysis (LSA) [139]. As opposed to LDA, LSA is not a generative model. It consists in extracting a spatial representation for words from a multi-document corpus by applying singular value decomposition. However, as pointed out in [140], spatial representations are inadequate for capturing the structure of semantic association; LSA assumes symmetric similarity between words which is not the case for a vast majority of associations. One classical example given in [140] involves China and North Korea: Griffiths *et al.* noticed that, generally speaking, people have always the intuition that North Korea is more similar to China than China to North Korea. This problem is resolved in LDA where $P(\text{occurence of } word1 | \text{occurence of } word2) \neq P(\text{occurence of } word2 | \text{occurence of } word1)$. In addition, [140] showed that LDA outperforms LSA in terms of drawing semantic correlations between words.

Collaborative Filtering Our algorithm is based on discovering latent correlations between user interests in order to cluster users. An alternative approach could be to employ model-based collaborative filtering (MBCF) that avoids using semantic-knowledge. In MBCF, each user is represented by his interest vector. The size of this vector equals the number of all defined interest names, and its coordinates are defined as follows: a coordinate is 1 if the user has the corresponding interest name, otherwise it is 0. Since interest names are user-generated, the universe of all such names, and hence the vector size can be huge. This negatively impacts the performance.

In particular, collaborative filtering suffers from a “cold-start" effect [141], which means that the system cannot draw correct inferences for users who have insufficient information (i.e., small number of interests). Recall from Section 3.5 that 70% of users in PubProfiles have less than 5 interests and it is more than 85% in RawProfiles. Hence, the sparseness of users' interest vectors is very high (the average density[7] is 0.000025). Moreover, [142] has studied the effect of cold-start in recommendation systems (for both item-based and collaborative-based) on real datasets gathered from two IP-TV providers. Their results show that a well-known CF algorithms, called SVD [143], performs poorly when the density is low (about 0.0005) with a recall between 5% and 10%. Additionally,

[7]The density of this vector is the number of coordinates equal one divided by the vector size.

the number of new users and interests is ever growing (on average, 20 millions new users joined Facebook each month in the first half of 2011 [138]). This tremendous number of new users and interests keeps the system in a constant cold-start state. In addition, users, in MBCF typically evaluate items using a multi-valued metric (e.g., an item is ranked between 1 and 5) but it must be at least binary (e.g., like/dislike), whereas in our case, only "likes" (interests) are provided. In fact, the lack of an interest I in a user profile does not mean that the user is not interested in I, but he may simply not have discovered I yet. In these scenarios, when users only declare their interests but not their disinterest, MBCF techniques (e.g., SVD [143]) are less accurate than nearest neighbor-like approaches that we employed [144].

OSN independence One interesting feature of our technique is that it is OSN independent. In particular, it does not rely on any social graph, and the input data (i.e. interest names) can be collected from any other source of information (e.g., deezer, lastfm, or any other potential sources).

No need for frequent model updates (stability) One may argue that our LDA model needs frequent updates since user interests are ever-growing. Nevertheless, recall from Section 3.4.2 that our technique uses the parent topics of the user interests (according to Wikipedia) to augment the semantics knowledge of each interest. There are substantially fewer higher-level parent categories than leaf categories in the Wikipedia hierarchy, and they change less frequently. Thus, there is no need to update the LDA model, unless the considered interest introduces a new parent category in the running model. Hence, our approach is more stable than MBCF; once the IFV vector is extracted and similarity is computed, we can readily make inference *without* having to retrain the system.

Targeted advertising and spam Using our technique, advertisers could automatically build online profiles with high accuracy and minimum effort *without* the consent of users. Spammers could gather information across the web to send targeted spam. For example, by matching a user's Facebook profile and his email address, the spammer could send him a message containing ads that are tailored to his inferred geo-localization, age, or marital status.

3.8 Conclusion

This chapter presents a semantics-driven inference technique to predict private user attributes. Using only Music Interests that are often disclosed by users, we extracted unobservable Interest topics by analyzing the corpus of Interests, which are semantically augmented using Wikipedia, and derived a probabilistic model to compute the belonging of users to each of these topics. We estimated similarities between users, and showed how our model can be used to predict hidden information. These disclosed attributes can be exploited to carry out numerous malicious attacks among which social aware phishing (i.e., spear phishing) is probably the most visible (i.e., 40% of social network accounts are used

for spam[8]) and the most studied too [10, 145]. Such attack have been demonstrated to be highly effective as stolen information from victims is hardly detectable [146].

Next chapter extends these works by demonstrating a *novel* attack that exploits user leaked or inferred attributes to speed up password cracking process. In this context, the consequences of the privacy breach are no more "hypothetical" but have an immediate impact on the security of the user accounts.

[8] http://www.businessweek.com/articles/2012-05-24/likejacking-spammers-hit-social-media

Chapter 4

When Security meets privacy: Faster Password Guessing Leveraging Social Information

Contents

4.1 Introduction

The previous chapter presented an inference algorithm to infer hidden or missing attributes from a user profile. We showed that our algorithm is able to infer various personal information with a high accuracy by solely relying on public data. In this chapter, we present a novel attack that exploits these information to break the user security. Specifically, we construct a password cracker that leverage user information to speed up the process of password guessing. We target users' password as password-based authentication is the most widely used form of user authentication, both online and offline. Passwords will likely remain the predominant form of authentication for the foreseeable future, due to a number of advantages: passwords are highly portable, easy to understand for laypersons, and easy to implement for the operators. Despite the weaknesses passwords have, they still are and will be in use for some time. The reason can be found in [147], which lists a large number of criteria that user authentication mechanism may fulfill, and evaluates the quality of a large number of user authentication mechanisms. While alternative forms of authentication can replace passwords in specific scenarios, they have not been able, so far, to replace them on a large scale.

User chosen passwords have a number of potential problems, most importantly they are vulnerable to guessing attacks. In this scenario, the attacker has access to an oracle (either online or offline) that verifies whether a password guess is correct. The number of guesses could be bounded by a number of factors: 1) in an online attack, it could be bound by the server limiting the number of attempts per unit of time; 2) in an offline attack, the limit lies in the amount of time and resources the attacker is willing to commit to the task of correctly guessing the password.

In this work, we concentrate on offline guessing attacks, in which the attacker can make a number of guesses bounded only by the time and resources she is willing to invest. While such attacks can be improved by increasing the speed with which an attacker can verify guesses (e.g., by using specialized hardware and large computing resources [148, 149]), we concentrate here on techniques to reduce the number of guesses required to crack a password. Hence, our approach reduces the attack time independently of the available resources.

The optimal strategy for password cracking (both offline and online) is to enumerate passwords in decreasing order of likelihood, i.e., trying more frequent passwords first and less frequent passwords later. Moreover, human chosen passwords frequently have a rich structure which can be exploited to generate candidate guesses (e.g., using a dictionary and a set of concatenation rules).

Recent work [38, 150] has shown ways to improve cracking performance by enumerating guessed passwords based on their likelihood. These password crackers have been shown to outperform JtR in certain conditions. The first insight of our work will be to build upon and improve on the performance of these probabilistic password crackers. Furthermore, while previously proposed password crackers outperform John the Ripper (JtR) – one of the most used password cracker –, they do not consider any *user specific* information. This means that the guesses outputted by each of these tools are fixed and do

not depend upon additional information about the user.[1] However, as a tremendous amount of personal information about users is available — either publicly or through attacks such as the one presented in the previous chapter — one might exploit this data to speed up the cracking process.

Common sense would suggest that guessing a password can be done more efficiently when personal information about the victim is known. For example, one could try to guess passwords that contain the victim's date of birth or the names of their siblings. However, this raises the question, how do we include this personal information to in the guessing? Shall we include all the information about the victims or only restrict the attack to a subset of that information? To the best of our knowledge, these questions have not been thoroughly explored so far and, as we will show, have a serious impact on the security of passwords. This is especially true since the steadily increasing use of social networks gives attackers access to a vast amount of public information about their victims for the purpose of password cracking.

Chapter organization We review some basics on password guessing, commonly used password guessers, as well as more related work in Section 4.2. In Section 4.3 we describe the *Ordered Markov ENumerator* (OMEN) and provide several experiments for selecting adequate parameters. Section 4.4 compares OMEN performance to other password guessers. Next section, details the idea of exploiting personal information in password guessing, provides some basic statistics about the data we use, and presents a detailed description of our algorithm. Experimental results are depicted in Section 4.6. We conclude with a brief discussion in Section 4.7.

4.2 Related Work In Password Cracking

One of the main problems with passwords is that many users choose *weak* passwords. These passwords typically have a rich structure and thus can be guessed much faster than with brute-force guessing attacks. Best practice mandates that only the hash of a password is stored on the server, not the password, in order to prevent leaking plain-text when the database is compromised. Furthermore, additional *salting* is used to avoid pre-computation attacks. Let H be a hash function and pwd the password, choose a random bit string $s \in^{\mathscr{R}} \{0,1\}^{16}$ as salt and store the pair $(s, h = H(pwd \mid\mid s))$. In this work we mainly consider *offline guessing attacks*, where an attacker is given access to the pair (s,h), and tries to recover the password pwd. The hash function is frequently designed for the purpose of slowing down guessing attempts [151]. This means that the cracking effort is *strongly dominated by the computation of the hash function* making the cost of generating a new guess relatively small. Therefore, we evaluate all password crackers based on the number of attempts they make to correctly guess passwords.

4.2.1 John the Ripper

John the Ripper (JtR) [152] is one of the most popular password crackers. It proposes different methods to generate passwords: In *dictionary* mode, a dictionary of words is

[1] JtR does some very limited guessing depending on the username

provided as input, and the tool tests each one of them. Users can also specify various mangling rules. When mangling rules are provided, JtR applies each rule to each word in the input dictionary. In real attacks, the dictionary mode using simple mangling rules works surprisingly well, especially when the input dictionary is derived from large collections of leaked passwords. Similarly to [153], we discover that for relatively small number of guesses (less than 10^8), JtR in dictionary mode produces best results. However, we focus on attacks with larger number of attempts, for which simple, non probabilistic approaches fall short.
In Incremental mode (JtR-inc) [152], JtR tries passwords based on a (modified) 3-gram Markov model. Specifically, JtR-inc computes not only the probability of each 3-gram but also the probability that this particular 3-grams appears at certain indices. In this way, this attack takes into account the structure of the password (e.g., upper case appears usually at the front while numbers at the end). However, two key differences are to be stressed: first, as for Narayanan and al. algorithm [38], the guesses are not generated in the *true* probability order (i.e., form the most probable password to the less probable) and second, the Markov chain is modified so that it can cover the entire key-space.

4.2.2 Password Guessing with Markov Models

Markov models have been proven to be useful for computer security in general and for password security in particular. They are an effective tool to crack passwords [38], and can likewise be used to accurately estimate the strength of new passwords [154].

The underlying idea is that adjacent letters in human-generated passwords are not independently chosen, but follow certain regularities (e.g., the 2-gram `th` is much more likely than `tq` and the letter `e` is very likely to follow `th`). In an n-gram Markov model, one models the probability of the next character in a string based on a prefix of length $n-1$. Hence, for a given string $c_1, \ldots, c_m$, a Markov model estimates its probability as

$$\begin{aligned} & P(c_1, \ldots, c_m) \\ & \approx P(c_1, \ldots, c_{n-1}) \cdot \prod_{i=n}^{m} P(c_i | c_{i-n+1}, \ldots, c_{i-1}). \end{aligned} \tag{4.1}$$

For password cracking, one basically learns the initial probabilities $P(c_1, \ldots, c_{n-1})$ and the transition probabilities $P(c_n | c_1, \ldots, c_{n-1})$ from real-world data (which should be as close as possible to the distribution we expect in the data that we attack), and then enumerates passwords in order of descending probabilities as estimated by the Markov model (according to Equation 4.1).

To make this attack efficient, we need to consider a number of details. First, one usually has a limited dataset when learning the initial probabilities and transition probabilities. The limited data entails that one cannot learn frequencies with arbitrarily high accuracy, i.e., *data sparseness* is a problem. The critical parameters are the size of the alphabet Σ and the parameter n, which determines the length of the n-grams.

Second, one needs an algorithm that *enumerates the passwords* in the right order. While it is easy to compute the probability for a given password, it is not clear how to enumerate the passwords in decreasing probability. To overcome this problem, [38] provides a

method to enumerate all passwords that have probability larger than a given threshold λ, but not necessarily in descending order. Hence, *all* passwords that have a probability (approximately) higher than λ are produced in output, where λ is an input parameter. This is sufficient for attacks based on pre-computation (rainbow tables), but not for "normal" guessing attacks where guessing passwords in the correct order can drastically reduce guessing time.

Notably, an add-on to JtR has been released with an independent implementation of the algorithm presented in [38]. The implementation is available at [152]. In the evaluation section, we use this implementation as a comparison (and refer to as JtR-Markov).

4.2.3 Probabilistic Grammars-based Schemes

Probabilistic context-free grammar (PCFG) schemes make the assumption that password structures have different probabilities [150]. In other words, some structures, such as passwords composed of 6 letters followed by 2 digits, are more frequent than others. The main idea of these schemes is therefore to extract the most frequent structures, and use them to generate password candidates.

More precisely, in the *training phase*, different structures are extracted from lists of real-world passwords, where each structure indicates the positions of lower and upper case letters, numerical, and special characters, as well as the associated probabilities. In the *attack phase (or password generation phase)*, an algorithm outputs the possible structures for the grammar with decreasing probabilities. From this output, which describes positions of the four classes of characters, password guesses are generated as follows: Numerical and special characters are substituted by those that have been observed in the training phase and in decreasing order of probability, and letters are substituted with appropriate words from a dictionary. This gives the final password candidate list.

4.2.4 Password Strength Estimation

A problem closely related to password guessing is that of *estimating the strength of a password*, which is of central importance for the operator of a site to ensure a certain level of security. In the beginning, password cracking was used to find weak passwords [155]. Since then, much more refined methods have been developed. One used so-called pro-active password checkers to exclude weak passwords [156–160]. However, most pro-active password checkers use relatively simple rule-sets to determine password strength, which have been shown to be a rather bad indicator of real-world password strength [154, 161, 162]. The influence of password policies on password strength is studied in [163], and [164] proposes new methods for measuring password strength and applies them to a large corpus or passwords. More recently, Schechter *et al.* [165] classified password strength by counting the number of times a certain password is present in the password database. Finally, Markov models have been shown to be a very good predictor of password strength while being provably secure [154].

4.3 OMEN: An Improved Markov Model Based Password Cracker

In this section we present our efficient implementation of password enumeration based on Markov models, which is the first contribution of this chapter. Our implementation improves previous work based on Markov models by Narayanan *et al.* [38] and JtR [152]. Note that the indexing algorithm presented by Narayanan *et al.* [38] combines two ideas: first, it uses Markov models to index only passwords that have high probability, and second, it utilizes a hand-crafted finite automata to accept only passwords of a specific form (e.g., eight letters followed by a digit). Our algorithm is solely based on Markov models to create guesses. However, it could be combined with similar ideas as well.

4.3.1 An Improved Enumeration Algorithm

Narayanan *et al.* 's indexing algorithm [38] has the disadvantage of not outputting passwords in order of decreasing probability, however, guessing passwords in the right order can substantially speed up password guessing (see the example in Section 4.4). We developed an algorithm, the *Ordered Markov ENumerator* (OMEN), to enumerate passwords with (approximately) decreasing probabilities.

On a high level, our algorithm discretizes all probabilities into a number of bins, and iterates over all those bins in order of decreasing likelihood. For each bin, it finds all passwords that match the probability associated with this bin and outputs them. More precisely, we first take the logarithm of all n-gram probabilities, and discretize them into levels (denoted η) similarly to Narayanan *et al.* [38], according to the formula $lvl_i = \text{round}\left(\log(c_1 \cdot prob_i + c_2)\right)$, where c_1 and c_2 are chosen such that the most frequent n-grams get a level of 0 and that n-grams that did not appear in the training are still assigned a small probability. Note that levels are negative, and we adjusted the parameters to get the desired number of levels(nbLevel), i.e., the levels can take values $0, -1, \ldots, -(\text{nbLevel}-1)$ where nbLevel is a parameter. The number of levels influences both the accuracy of the algorithm as well as the runtime: more levels means better accuracy, but also a longer running time.

For a specific length ℓ and level η, enumPwd(η, ℓ) proceeds as follows:

1. It identifies all vectors $\vec{a} = (a_2, \ldots, a_l)$ of length $\ell - 1$ (when using 3-grams we need $\ell - 2$ transition probabilities and 1 initial probability to determine the probability for a string of length ℓ), such that each entry a_i is an integer in the range [0,nbLevel-1] , and the sum of all elements is η.

2. For each such vector $\vec{a}$, it selects all 2-grams x_1x_2 whose probabilities match level a_2. For each of these 2-grams, it iterates over all x_3 such that the 3-gram $x_1x_2x_3$ has level a_3. Next, for each of these 3-grams, it iterates over all x_4 such that the 3-gram $x_2x_3x_4$ has level a_4, and so on, until the desired length is reached. In the end, this process outputs a set of candidate passwords of length ℓ and level (or "strength") η.

A more formal description is presented in Algorithm 1. It describes the algorithm for $\ell = 4$. However, the extension to larger ℓ is straightforward.

Algorithm 1 Enumerating passwords for level η and length ℓ (here for $\ell = 4$).[2]

function enumPwd(η, ℓ)

1. for each vector $(a_i)_{2 \le i \le \ell}$ with $\sum_i a_i = \eta$
   and for each $x_1x_2 \in \Sigma^2$ with $L(x_1x_2) = a_2$
   and for each $x_3 \in \Sigma$ with $L(x_3 \mid x_1x_2) = a_3$
   and for each $x_4 \in \Sigma$ with $L(x_4 \mid x_2x_3) = a_4$:
   (a) output $x_1x_2x_3x_4$

Example We illustrate the algorithm with a brief example. For simplicity, we consider passwords of length $\ell = 3$ over a small alphabet $\Sigma = \{a, b\}$, where the initial probabilities have levels

$$L(aa) = 0, \quad L(ab) = -1,$$
$$L(ba) = -1, \quad L(bb) = 0,$$

and transitions have levels

$$L(a|aa) = -1 \quad L(b|aa) = -1$$
$$L(a|ab) = 0 \quad L(b|ab) = -2$$
$$L(a|ba) = -1 \quad L(b|ba) = -1$$
$$L(a|bb) = 0 \quad L(b|bb) = -2.$$

- Starting with level $\eta = 0$ gives the vector $(0,0)$, which matches to the password `bba` only (the prefix "aa" matches the level 0, but there is no matching transition with level 0).

- Level $\eta = -1$ gives the vector $(-1,0)$, which yields `aba` (the prefix "ba" has no matching transition for level 0), as well as the vector $(0,-1)$, which yields `aaa` and `aab`.

- Level $\eta = -2$ gives three vectors: $(-2,0)$ yields no output (because no initial probability matches the level -2), $(-1,-1)$ yields `baa` and `bab`, and $(0,-2)$ yields `bba`.

- and so on for all remaining levels.

The selection of ℓ (i.e., the length of the password to be guessed) is challenging, as the frequency with which a password length appears in the training data is not a good indicator of how often a specific length should be guessed. For example, assume that are as many passwords of length 7 and of length 8, then the success probability of passwords of length 7 is larger as the search-space is smaller. Hence, passwords of length 7 should be guessed first. Therefore, we use an adaptive algorithm that keeps track of the success ratio of each length and schedules more passwords to guess for those lengths that were more effective. More precisely, our adaptive password scheduling algorithm works as follows:

[2]Here $L(xy)$ and $L(z|xy)$ stand for the level of initial and transition probabilities, respectively.

1. For all n length values of ℓ (we consider lengths from 3 to 20, i.e. $n = 17$), execute enumPwd$(0,\ell)$ and compute the success probability $sp_{\ell,0}$. This probability is computed as the ratio of successfully guessed passwords over the number of generated password guesses of length ℓ.

2. Build a list L of size n, ordered by the success probabilities, where each element is a triple $(sp, level, length)$. (The first element $L[0]$ denotes the element with the largest success probability.)

3. Select the length with the highest success probability, i.e., the first element $L[0] = (sp_0, level_0, length_0)$ and remove it from the list.

4. Run enumPwd$(level_0 - 1, length_0)$, compute the new success probability sp^*, and add the new element $(sp^*, level_0 - 1, length_0)$ to L.

5. Sort L and go to Step 3 until L is empty or enough guesses have been made.

4.3.2 Selecting parameters

In this section we discuss several parameters choices and examine the necessary trade-off between accuracy and performance. In the following, we detail the rational behind the choice of three central parameters: n-gram size, alphabet size and the number of levels for enumerating passwords.

n-gram size The parameter with the greatest impact on accuracy is the size of the n-grams. A larger n generally gives better results as larger n-grams fit better the password distribution. However, it implies a larger runtime, as well as a larger memory and storage requirements. Note also that the amount of training data is crucial as only a significant amount of data can accurately estimate the parameters (i.e., the initial probabilities and the transition probabilities). We evaluated our algorithm with $n = 2, 3, 4$, results are depicted in Figure 4.1.

As expected, the larger n is, the better the results are. Experiments with 5-grams are not depicted in this Figure, but show slightly better results than for 4-grams. As 5-grams implies a much longer running time, memory and storage for a small performance increase, we decided to choose $n = 4$.

Alphabet size The size of the alphabet is another factor that has the potential to substantially influence the characteristics of the attack. Larger alphabet size means that more parameters need to be estimated and that the runtime and memory requirements increase. In the opposite, a small alphabet size means that not all passwords can be generated.

We tested several alphabet sizes by setting $k = 20, 30, 40, 50, 62, 72, 92$ where k represents the most frequent characters from the training set. The results are given in Figure 4.2 and Table 4.1. We clearly see a sharp increase in the accuracy from an alphabet size of 20 to the one of 62. After this threshold (62) , the success cracking rate remains roughly the same. This is mainly explained by the alphabet used in the RockYou dataset where most users favour password with mostly alphanumeric characters rather than using special ones.

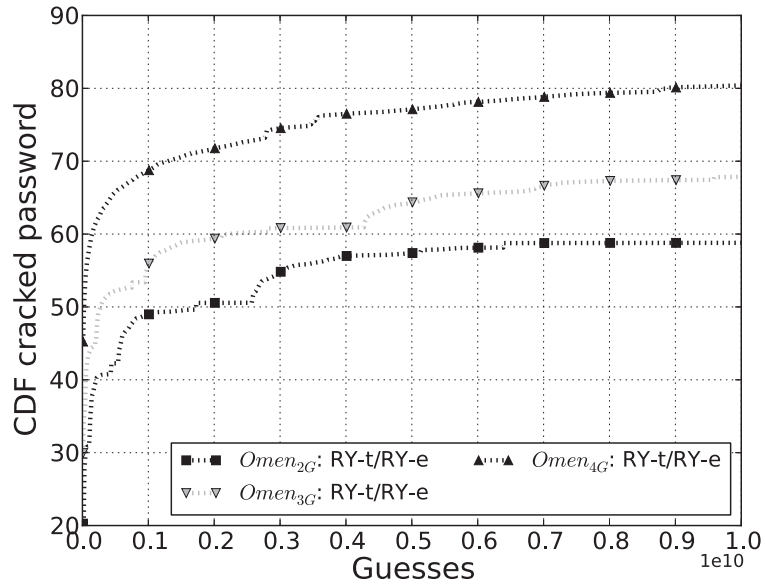


FIGURE 4.1: Comparing different *n*-gram sizes for the RockYou dataset.

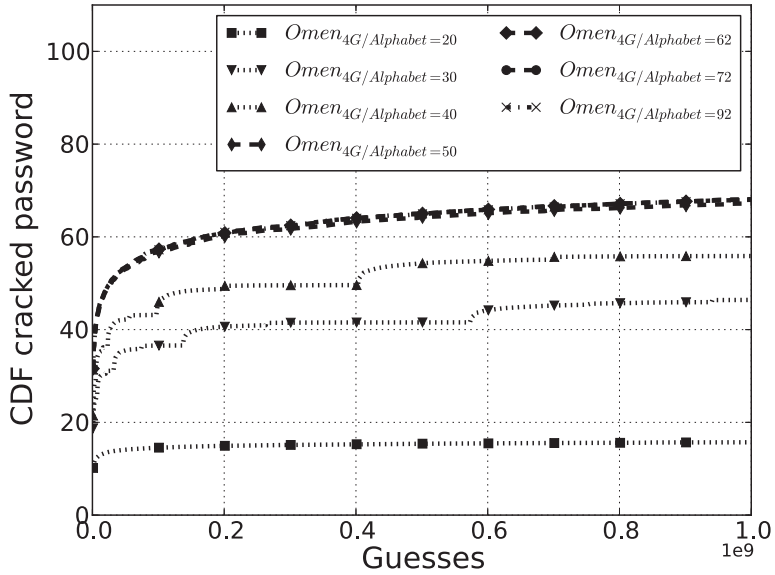


FIGURE 4.2: Comparing different alphabet sizes for the RockYou dataset.

To minimize the impact of the charset while keeping fast computation, we opted for the 72 charset alphabet.

Number of levels A third important parameter is the number of levels that are used to enumerate password candidates. As for previous parameters, higher number of levels can potentially increase accuracy, but it also increases the runtime.

$\|\Sigma\|$	20	30	40	50	62	72	92
	15.7	46.4	55.86	67.18	68.1	68.08	68

TABLE 4.1: Percentage of cracked password for 1B guesses and varying alphabet sizes.

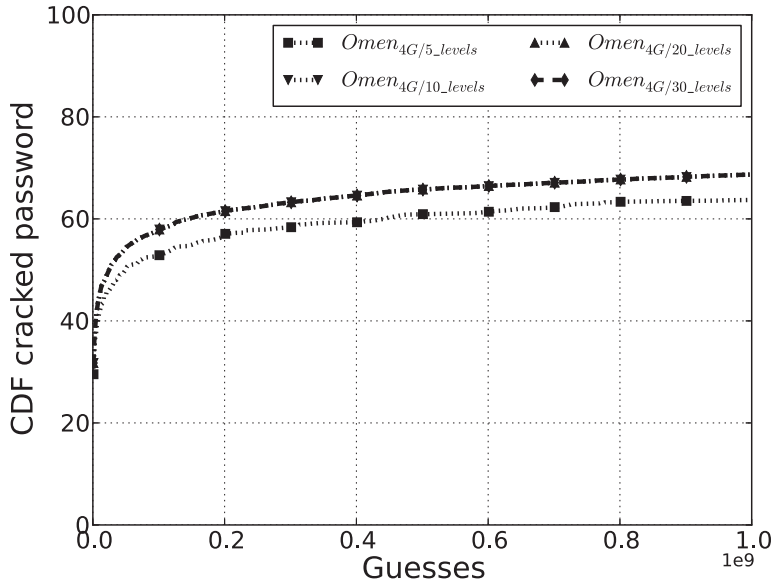


FIGURE 4.3: Comparing different number of levels for the RockYou dataset.

The results are shown in Figure 4.3. We see that increasing the number of levels from 5 to 10 substantially increases accuracy, but further increasing to 20 and 30 does not make a significant difference.

#Levels	5	10	20	30
	63.70 %	68.68 %	68.70 %	68.70 %

TABLE 4.2: Accuracy for 1B guesses and varying number of levels.

Selected parameters Unless otherwise stated, in the following, we use omen with the following parameters: 4-gram, alphabet size of 72 and 10 levels.

4.4 Evaluating OMENs performance

In this section, we present a comparison between our improved Markov model password cracker and previous state-of-the-art solutions.

4.4.1 Datasets

We evaluate the performance of our password guesser on multiple datasets. The largest password list publicly available is the *RockYou list* (RY), consisting of 32.6 million passwords that were obtained by an SQL injection attack in 2009. The passwords were leaked in clear, all further information was stripped from the list before it was leaked to the public. This list has two advantages: first, its large size gives well-trained Markov models; second, it was collected via an SQL injection attack therefore affecting all the users of the compromised service. We *randomly* split the RockYou list into two subsets: a *training set* (RY-t) of 30 million and a *testing set* (RY-e) of the remaining 2.6 million passwords.

The *MySpace list* (MS) contains about 50000 passwords (different versions with different sizes exist, most likely caused by different sanitation or leaked from the servers at different points in time). The passwords were obtained in 2006 by a phishing attack.

Attribute	Availability	Mean	Std
Friends	67.77%	243.6	388
EduWork	46.28%	2	0.9
Contacts	20.94%	1.15	0.65
Siblings	20.44%	5.8	6.1
Birthday	0.97%	-	-
Current City	16.2%	-	-
Home Town	13.93%	-	-

TABLE 4.3: Facebook dataset statistics.

The *Facebook* list (FB) was posted on the pastebin website[3] in 2011. This dataset contains both Facebook passwords and associated email addresses. It is unknown how the data was obtained by the hacker, but most probably was collected via a phishing attack. We complemented this list by collecting the public information associated to the Facebook profiles connected to the email address. For each profile, we collected the public attributes, which include: first/last name; location; date of birth; friends names; siblings names; education/work names. Note that the access to user profile is governed by a set of privacy policies and as such, not all data is publicly accessible. Table 4.3 provides statistics about attributes availability as well as the mean and standard deviation for multivalued attributes (e.g., Friends list or Education history).

Finally, we used a list of 60000 email addresses and passwords leaked by the group LulzSec (we call this list LZ). The list was publicly released in June 2011 via Twitter[4].

Ethical Considerations Studying databases of leaked password has arguably helped the understanding of users real world password practices and as such, have been used in numerous studies [150, 154, 161]. While these datasets are already available to the public,

[3] `http://pastebin.com/`

[4] `https://twitter.com/#!/LulzSec/status/81327464156119040`

we believe that the information they contain is private; therefore we treat these lists as confidential [166].

4.4.2 Results

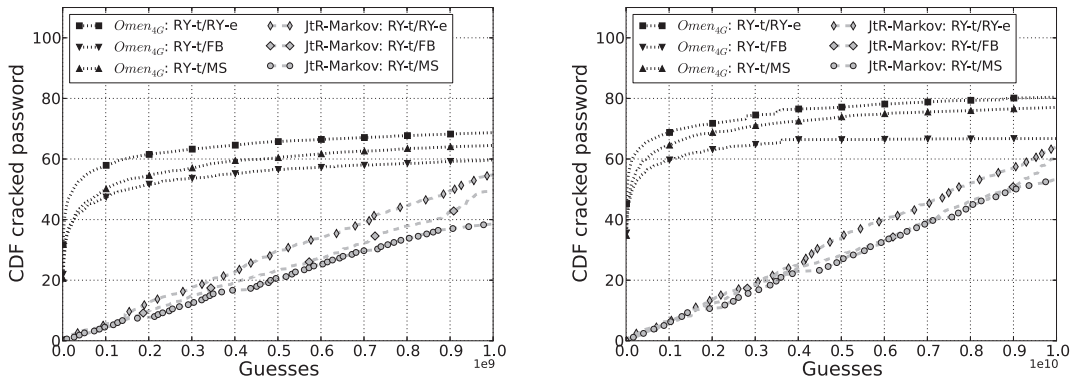


FIGURE 4.4: Comparing OMEN with the JtR Markov mode at 1B guesses (left) and at 10 billion guesses (right).

In this section, we evaluate the efficiency of our password guesser OMEN, and compare it with other password guessers on different datasets. We discover that OMEN has consistently better performance compared to previously proposed algorithms. For the experiments, we trained all algorithms using the RockYou training set RY-t, and evaluated it on the sets RY-e, MS and FB. Table 4.4 provides a summary of the results. The table can also be used as a comparison of all the previously proposed password crackers.

Algorithm	Training Set	Testing Set		
		RY-e	MS	FB
Omen	RY-t (10^{10})	80.40%	77.06%	66.75%
	RY-t	68.7%	64.50%	59.67%
PCFG [150]	RY-t	32.63%	51.25%	36.4%
JtR-Markov [38]	RY-t (10^{10})	64%	53.19%	61%
	RY-t	54.77%	38.57%	49.47%
JtR-Inc	RY-t(10^{10})	54%	25.17%	14.8%

TABLE 4.4: Summary table indicating the percentage of cracked passwords for 1 billion guesses (or 10 billion when specified).

4.4.2.1 OMEN vs JtR's Markov Mode

Figure 4.4 shows the comparison of OMEN and the Markov mode of JtR. JtR's Markov mode implements the password indexing function by Narayanan et al. [38]. Both models are trained on a list of passwords (RY-t). Then, given a target number of guesses T (here 1 billion in the left graph and 10 billion in the right graph), we computed the corresponding

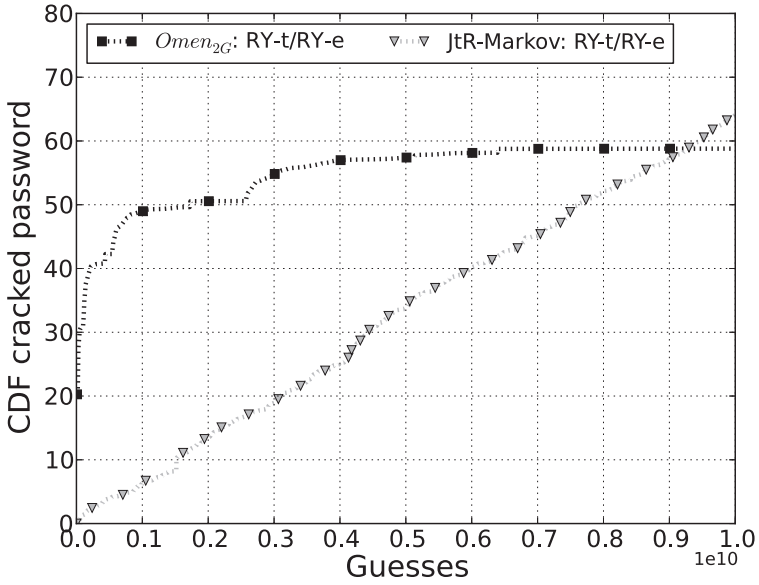


FIGURE 4.5: Comparing OMEN using 2-grams with the JtR Markov mode.

level (η) to output T passwords. The curve shows the dramatic improvement in cracking *speed* given by our improved ordering of the password guesses. In fact, JtR-Markov outputs guesses in no particular order which implies that likely passwords can appear "randomly" late in the guesses. This behaviour leads to the near-linear curves shown in Figure 4.4. One may ask whether JtR-Markov would surpass OMEN after the point T; the answer is *no* as the results do not extend linearly beyond the point T; and larger values of T lead to a flatter curve. To demonstrate this claim, we performed the same experiment with T equals to 10 billion guesses (instead of 1 billion). Figure 4.4 (right) shows how the linear curve becomes *flatter*.

To show the generality of our approach, we compare the cracking performance on three different datasets:RY-e, FB and MS. The ordering advantage allows OMEN to crack more than 40% of passwords (independently of the dataset) in the first 90 million guesses while JtR-Markov cracker needs at least eight times as many guesses to reach the same goal. For the RockYou dataset the result are impressive: OMEN cracks 45.24% of RY-e passwords in the first 10 million guesses (see Figure 4.4 (right)) while JtR-markov achieves this result after more than 7 billion guesses.

In the above comparison, OMEN uses 4-grams (as determined in Section 4.3.2), while JtR-Markov uses 2-grams. To see the effects that this difference has, we provide an additional comparison of OMEN using 2-grams with JtR-Markov, this is given in Figure 4.5. The results are as expected: JtR-Markov still has a straight line, which means that OMEN has a better cracking speed. The speed advantage of OMEN can be seen at 1 billion guesses where OMEN cracks 50% of all passwords while JtR-markov cracks less than 10% of them. At the point T, i.e., when JtR-Markov stops, both algorithms perform roughly the same. Note that since all parameters (i.e., alphabet size, number of levels etc.) of both models are not identical, we have a small difference in the cracking rate at the point

T.

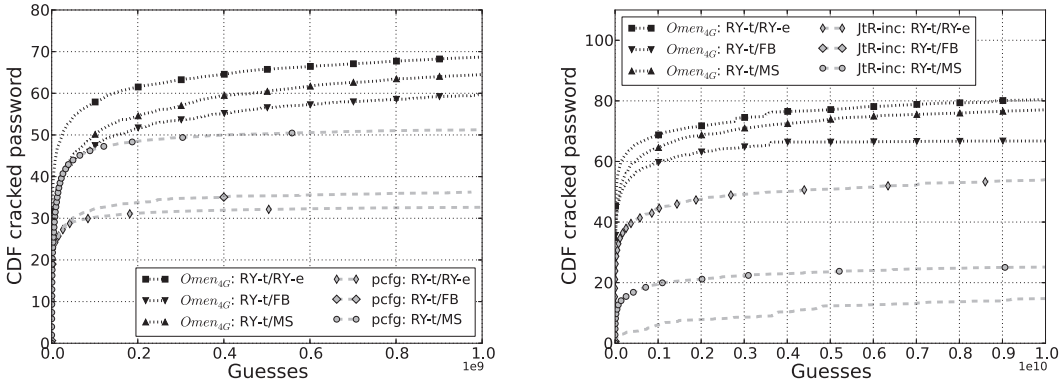


FIGURE 4.6: Comparing OMEN to the PCFG guesser (left) and to JtR incremental mode (right).

4.4.2.2 OMEN vs PCFG

Figure 4.6 compares OMEN to the PCFG password guesser of Weir *et al.* [150], based on the code available in [167]. We run it using the configuration as described in the paper [150]: we use RY-t to extract the grammar and the dictionary dict-0294 [168] to generate candidate passwords.

Figure 4.6 shows that OMEN outperforms the PCFG guesser. After 0.2 billion guesses, OMEN cracks 20% more passwords than PCFG for both MS and FB datasets and 10% more for RY-e. It is interesting to see the impact of the training set on PCFG performance: PCFG performs much better on RY-e than on FB and MS. We believe the reason is that the grammar for PCFG is trained on a subset of the RY list, which is better adapted for guessing the RY list. OMEN achieves roughly the same results for all datasets which proofs the robustness of the learning phase. Finally, note that PCFG mostly plateaus after 0.3 billion guesses and results hardly improve any more, whereas OMEN still produces noticeable progress.

4.4.2.3 OMEN vs JtR's Incremental Mode

We also compare OMEN to JtR in incremental mode (Figure 4.6). Similarly to the previous experiments, both crackers were trained on the RockYou training set of 30 million passwords and tested on RY-e, MS and FB datasets. Clearly, JtR incremental mode produces worse guesses than OMEN. Notably, it also produces worse guesses than any other cracker tested.

4.5 Personal Information and Password Guessing

The results of the previous section show that a significant fraction of passwords can be guessed with a (relatively) moderate number of attempts, compared to the entire possible search space. However, most techniques adopt a *coarse grained* approach that relies on a *generic* probability distribution, which by definition, does not depend on the

password being guessed. Intuitively, exploiting personal information in the password cracking process may enhance the success ratio. Surprisingly, such possibility has not been extensively studied.

While a multitude of personal information can be used, we focus on a realistic scenario where these information can easily be extracted from a *public* source. For instance, an attacker armed with his victim email address, can gather her social network profile and use the collected information to guess her password. Such information includes:

- Information related to the *user's name*, such as first name, last name, username;
- *Social relations* such as friends' and family members' names;
- *Interests* such as hobbies, favorite movies, etc;
- *Location information* like the place of residence;

In the next sections, we explore the relationship between such information, referred to as *hint* in the rest of this chapter, and passwords, as well as its effect on password cracking.

4.5.1 Similarity between Passwords and Personal Information

To asses whether social information can be exploited to improve password cracking, we quantify the correlation between passwords and personal information. We use two different similarity metrics to capture different aspects of the potential overlap.

Longest common substring (LCSS): The LCSS of two strings is the longest string that is a substring of both strings. For example, the LCSS of the two strings `abcabc` and `abcba` is `abc`. We use the length of the LCSS as an indicator for similarity.

(Asymmetric) Jaccard similarity (JS): The Jaccard similarity index compares similarity of two sets. For two sets X and Y, the Jaccard index is $J(X,Y) := \frac{|X \cap Y|}{|X \cup Y|}$. It is a similarity measure on sets, not on strings, but for our application it is very natural to extract the n-grams of the strings and apply Jaccard similarity to the sets of n-grams. There is one drawback of this measure that does not match our application, namely that appending unrelated information to the hint (which degrades the "real" usefulness only for large appended text) rapidly decreases the JS value. Therefore, we use an "asymmetric JS" defined as follows: Given a password P and a *hint* H, and denoting the set of 3-grams that appear in P and H with P_{3g} and H_{3g}, respectively, we define $J^*_{3g}(P,H) := \frac{|P_{3g} \cap H_{3g}|}{|P_{3g}|}$. Figure 4.7 displays the cumulative distribution function (CDF) of the JS between passwords and personal information for FB and LZ datasets. For each password of the Facebook dataset, we compute the JS with each of its corresponding personal attribute.

The lowest (green) plot (FB Max) displays the CDF of the maximum of these values. It basically shows that about 35% of Facebook passwords are somewhat correlated with one of their user's attributes. Note that any non-zero similarity means that at least one 3-gram is shared, which already is a substantial overlap. The correlation becomes stronger for about 10% of users. The grey (FB Username) and black (LZ Username) curves display the CDF of the similarity between passwords and *UserNames* for the FB and LZ datasets, respectively. They both show similar shape, although surprisingly Facebook passwords

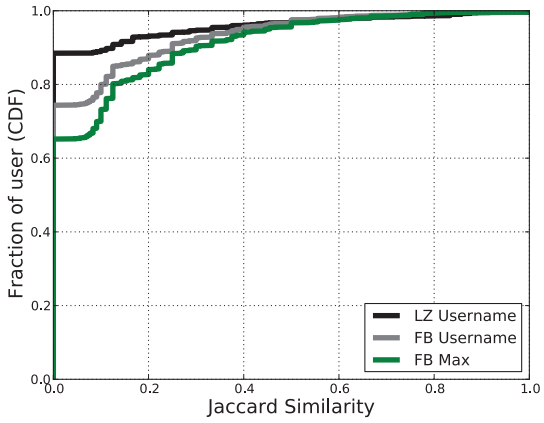


FIGURE 4.7: CDF of Jaccard similarity

seem to be more correlated with *UserNames* than LZ passwords. In both datasets, about 10% of passwords seem to be highly correlated with the *UserNames* attribute.

Attribute	Len	JS	JS(5%)	LCSS	LCSS(5%)
FirstName	5.84	0.02	0.31	0.93	4.34
LastName	6	0.01	0.24	0.71	3.55
Username	10.53	0.07	0.58	1.48	6.31
Friends	147.28	0.06	0.30	1.54	4.15
Edu/Work	40.93	0.02	0.23	1.20	3.5
Contacts	17.67	0.06	0.63	1.44	6.55
Location	25.70	0.01	0.13	1.07	2.94
Birthday	6.84	0.04	0.5	0.87	4
Siblings	94.71	0.04	0.36	1.27	4.96

TABLE 4.5: Mean similarity between passwords and personal information (FB dataset).

Table 4.5 goes into more details and summarizes similarity measures between different attributes and the respective passwords of the FB datasets. As shown by Figure 4.7, more than 60% of passwords have no similarity with personal information attributes. This explains the small similarity values in column JS and LCSS, that correspond to averages of the similarities over the whole dataset (since many similarities are equal to zero, the resulting averages have pretty low values). For this reason, we order the passwords according to their similarity values for the different attributes. We then present, in the columns JS(5%) and LCSS(5%), the average similarity of the top 5% for each attributes.

First we notice how attributes such as *UserNames*, *FirstName* and *Birthday* seem to substantially overlap with the passwords. For instance, for the top 5% users, *UserNames* and *Birthday* share half of the n-grams with the password and have more than 4 and 6.4 common substring with it respectively. Furthermore, we notice that long attributes such as *Friends* or *Education* and *Work* (on average 150 characters long and 50, respectively) have

a high value of LCSS. Finally, the LCSS(5%) results show that the average LCSS value is around 3 which sustains the usage of a 3-grams model (rather than 2-grams or 4-grams).

In Section 4.5.2.1 we will explore how to incorporate these findings to robustly increase the performance of OMEN.

4.5.1.1 Password Creation Policies and Usernames

One surprising fact highlighted by the data in the previous section is that usernames and passwords are very similar in a small, yet significant, fraction of the cases. This fact prompted us to study this specific aspect of password policies in depth, the reader may refer to appendix for more details. The results are worrisome: out of 48 tested sites, 27 allowed *identical* username and password, including major sites such as Google, Facebook, and Amazon, and only 4 sites required more than one character difference between the two. This could lead to highly effective guessing attack.

4.5.2 OMEN+: Improving OMEN Performance with Personal Information

In Section 4.5.1 we showed that users' personal information and passwords can be correlated. However, it is not clear how to use such information when guessing passwords, specifically because using some information that is not sufficiently relevant may have a negative impact on the performance. Let us illustrate this possibility with an example: assume that we possess extensive information about a victim. This information may include name, date of birth, location information, family member names, etc. Intuitively, this information should increase the performance of a password cracker. For example, we could generate a password guess with the name of a sibling concatenated with their year of birth. However, in order to increase the (overall) performance, one must still order password guesses in decreasing probability. Otherwise, the integration of additional information can decrease effectiveness. To show this, for the sake of argument, let us assume that the attacker is only allowed *one* guess. The same argument can be extended to any number of guesses. The attacker should use some personal information for the one guess only if the probability of this password is higher than the most frequent "generic" password, say `123456`. Assuming that no user specific password is, on average, more frequent than `123456`, then, by including personal information, the attacker would decrease her chances of success for the single guess.

Boosting Algorithm It is challenging to decide how to use the additional personal information. In fact, only certain parts of this data overlap with the password. In a dictionary-based attack, we need to decide which substring(s) should be added to the attack dictionary, and choosing the wrong ones one could decrease performance. With Markov models, however, the situation is easier, as n-grams are a canonical target. By increasing (conditional) probabilities of important n-grams (i.e., n-grams that are contained in $hints$), we can increase the probability of passwords that are related to them, and thus improve OMEN's performance. Our boosting algorithm takes as input a parameter $\alpha > 1$ (see Section 4.5.2.1 on how this parameter is chosen), a hint h, and a Markov model consisting of the initial probabilities $p(xy)$ and the conditional probabilities $p(z|xy)$, and outputs

modified conditional probabilities $p^*(z|xy)$.

Let us assume we have a list of N passwords $pwd_1, \ldots, pwd_N$, and for each password pwd_i we have some additional information $hint_i$, which may or may not help us in guessing the password. We want to automatically and efficiently determine if a specific set of hints is useful or not, and how strongly each of the hints should be weighted. Note that since hints are password-specific, trying all possible combinations would be too computationally expensive. Our algorithm works as follows[5]:

1. For each pair $pwd_i, hint_i$, two sets are defined: S_i is the set of 3-grams that appear in both the password and the hint and T_i is the set of 3-grams from pwd_i such that $hint_i$ has a 3-gram that shares the first two letters, but not the third. For instance, if $pwd_i =$ `password` and $hint_i =$ `passabcd` then $S_i = \{$`pas`, `ass`$\}$ and $T_i = \{$`ssw`$\}$.

2. For each 3-gram xyz in S_i, we set the conditional probability

$$p^*(z|xy) := \alpha \cdot p(z|xy)$$

for a given parameter α. Considering the previous example, we boost the 3-grams `pas` and `ass`, as follows: $p^*(s|pa) := \alpha \cdot p(s|pa); p^*(s|as) := \alpha \cdot p(s|as)$

By modifying a conditional probability from $\hat{p}$ to $\alpha \cdot \hat{p}$ we distribute a probability mass of $(\alpha - 1)\hat{p}$ that we need to subtract at another place. The probability $\hat{p}$ is (in practice) much smaller than 1 (we use an alphabet size of $|\Sigma| = 72$), so $1 - \hat{p} \approx 1$. Consequently, if we multiply all remaining (conditional) probabilities except $\hat{p}$ with $(1 - \alpha\hat{p})$, they sum up to approximately 1 again (using the approximation simplifies the calculations):

$$(1 - \alpha\hat{p}) \cdot (\sum_{i \neq z} p(i|xy)) + \alpha\hat{p} \approx (1 - \hat{p}\alpha) \cdot 1 + \alpha\hat{p} = 1.$$

Writing $s_i := |S_i|$ and $t_i := |T_i|$ for the sizes of the two sets, the overall effect on password probabilities is

$$p^*_{pwd_i} = \prod_{i \in (pwd_i)_{3g}} p_i \approx \alpha^{s_i}(1 - \hat{p}\alpha)^{t_i} \cdot p^{old}_{pwd_i}$$

where $p^*_{pwd_i}$ is the "new" probability after boosting the n-grams, and $p^{old}_{pwd_i}$ is the "old" probability before boosting.

4.5.2.1 Estimating Boosting Parameters

The section describes how the optimal boosting parameter α of each $hint_i$ is computed. Recall that OMEN outputs password guesses in descending order of their (estimated) probabilities. Let f denote the function that gives the estimated probabilities $x = f(y)$ for the y-th guess that OMEN outputs. This function can simply be computed by running OMEN and printing the probability estimation of the current password. The inverse function $y = f^{-1}(x)$ gives the number of guesses OMEN needs to output before reaching

	α	$\ln(\alpha)$	boosted
email	1.6	0.5	0(*)
userName	2	0.7	1
firstName	2.3	0.8	1
lastName	1.5	0.4	0
birthday	5	1.6	2
location	1.7	0.5	1
contact	1.5	0.4	0
eduWork	1.1	0.1	0
friends	1.4	0.3	0
siblings	1.7	0.5	1

TABLE 4.6: The estimated values of α and the boosting parameters for the attributes we considered.

passwords with a certain (estimated) probability. In order to simplify the subsequent calculations, we approximate this function as $f^{-1}(x) \approx x^{-1.5}$.

The estimated number of guesses which is required to crack *all* passwords $pwd_1, \ldots, pwd_N$ is consequently defined by $S = \frac{1}{N} \sum_{i=1,\ldots,N} f^{-1}(p_{pwd_i})$.Therefore, for a given $hint_i$, the value α_i to use to boost this hint is the value of α that minimizes the following function: $S^* = \frac{1}{N} \sum_{i=1,\ldots,N} (p^*_{pwd_i})^{-1.5}$.

The following section presents the optimal values of α for different hints.

4.6 Evaluation

4.6.1 Boosting Parameter Estimation

We use the techniques described in the previous section to estimate the boosting parameter α for the different types of hint. For each hint, we compute the sum S^* for different values of α, and select the value that minimizes it. We illustrate the results with three examples:

First Name When plotting S^* as a function of α for the attribute *first name* it shows a minimum at approximately $\alpha = 2.3$, which yields a boosting parameter of 1.

EduWork When plotting S^* as a function of α for the attribute *eduWork*, which is an identifier that contains the persons' education and occupancy, we notice a very small decrease in the beginning, with a minimum around $\alpha = 1.2$, but the overall differences are small and the remainder of the graph is monotonically increasing. The boosting parameter is 0.1, which is rounded to 0.

Birthday When plotting S^* as a function of α for the *birthday* attribute, we have to note that our dataset only contains a small number of profiles with that attribute, so the result

[5]To ease presentation, we only describe the estimation algorithm for 3-grams. The generalization to n-grams is straightforward.

might not be necessarily meaningful. Overall, there were only 7 profiles where the birthday attribute have an effect, and for two of them the effect is positive. These two have enough effect to lead to a great advantage in using the attribute. Overall, we limited the maximal parameter considered to 5.

Email We dropped the attribute *email*. Although it gives an alpha of 1.6 since the *username* is contained in email and achieves better results. The remaining boosting parameters are summarized in Table 4.6.

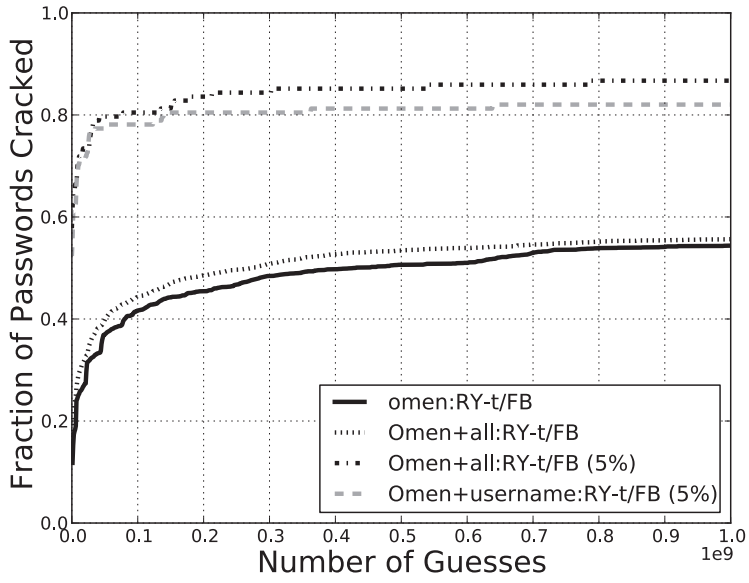


FIGURE 4.8: Comparing OMEN with and without personal information on the FB list

4.6.2 OMEN+ Performance

Once we have estimated the values of the parameters α_i for each $hint_i$, we run OMEN+ on the Facebook (FB) list, consisting of 3140 passwords together with publicly available information about the users (see Table 4.3 for data availability). The results are presented in Figure 4.8. As expected, by including personal information in the Markov model, OMEN+ is able to guess more passwords in absolute terms. We have also conducted different experiments with other values of α_i to test the effectiveness of our estimation code. We confirmed that, when using different values of α_i, the cracking performance either remains the same or slightly decreases. The two lower curves in the Figure 4.8 show the performance of OMEN+ over all passwords with and without using personal attributes. Using personal information can increase the guessed passwords up to 5% (for lower number of guesses of up to 100 million), and around 3% at 1 billion guesses. The limited performance gain is partially explained by the two facts: first, as shown in Table 4.3, few personal information are available in FB dataset (e.g., only 0.98% of all users relieved their birthday), and second, only a small proportion of passwords are based

on personal information (see Section 4.5.1). However, these results are specific to our dataset which assume a realistic – but rather a weak – adversary with a limited background knowledge (i.e., only publicly available information from a *single* source). However, as shown by Brown *et al.* [169] in a survey evaluating the generation and use of passwords, *two* thirds of passwords are designed around one's personal information and more than 50% of all passwords are constructed around proper names and birthdays. As such, our results represent a lower bound for password cracking leveraging personal information.

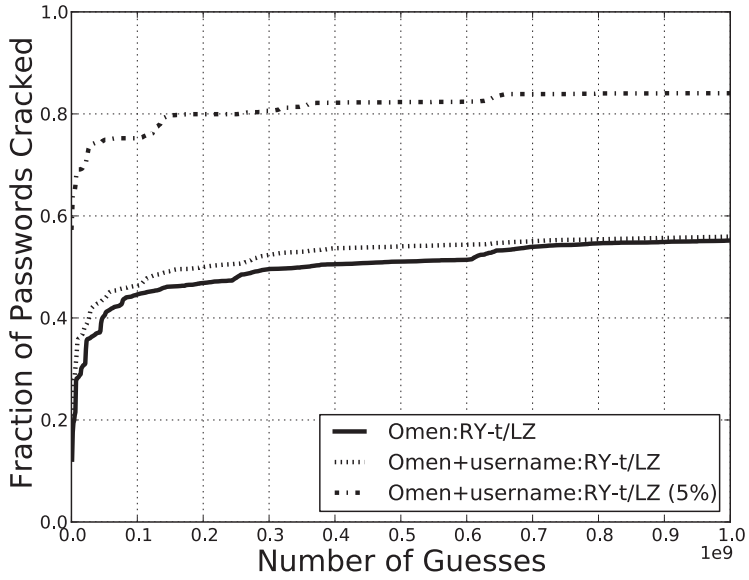


FIGURE 4.9: Comparing OMEN with usernames as hint and without on the LZ/FB list

The two curves in the upper part of the Figure 4.8 display the performance of OMEN+ over the top 5% passwords that are the most correlated with their personal attributes. The achieved performance is much better: About 82% of the passwords are cracked by using usernames only, and more than 88% by using all considered personal attributes. This result is very promising. Figure 4.9 shows a similar experiment performed on the LZ list (60000 passwords). We only had access to the local-part (i.e., the username) of the email associated to each password. Even though the information in this case was more limited, we realized similar gains compared to the previous test on the more extensive FB data.

4.7 Discussion and Conclusion

In this work we have first presented an efficient password guesser (OMEN) based on Markov models, which outperforms all publicly available password guessers. For common password lists we found that we can guess more than 80% of passwords with 10 billion guesses. Subsequently, in our second contribution, we tested if additional personal information about a user can help us better guessing passwords. We found that some attributes indeed help, and we showed how OMEN+ can efficiently exploit this information. We summarize some of the key insights:

- Markov models were long known [38] to be an effective tool in password guessing, but our work shows that they have an even better potential than previously thought, as we could make them guessing "in order", which leads to the improvements shown in Figures 4.4 and 4.6. Moreover, we assessed the impact of different parameters on the accuracy of the algorithm.

- We find that we can guess up to 5% more passwords when exploiting personal information. However this percentage is a lower bound since we had only access to limited amount of data (e.g., only 0.98% of users in our dataset disclosed their birthday see Table 4.3). However, as shown by Brown *et al.* [169], real passwords tend to be correlated with user information in most cases. In such context, the gain of OMEN+ can go up to 30%. This result clearly shows that passwords based on personal information are weaker and should be avoided.

Our work demonstrates how privacy leakages can be exploited in the process of cracking passwords and might lead to the compromise of the user account. This attack highlights a severe security threat resulting from *uninformed consent* of data release which calls for several countermeasures. First, users should be *informed* that using personal information in their passwords is a risky behaviour. Second, password strength measurement tools should provide a mechanism to integrate personal information, especially from social network, and treat passwords based on that information as weak. Finally, a privacy measurement tool for social profiles — while tackling an orthogonal problem — can persuade the user to share less information, decreasing the security risks. In the next chapter, we tackle the last point by proposing a quantitative privacy measure to quantify the privacy loss caused by publicly revealing attributes. Our approach can enhance privacy in at least two scenarios: (i) First, data providers can quantify the risk of re-identification (i.e., identifying a user in a dataset) when disclosing a set of attribute and (ii) Second, it provides a quantitative privacy measure: by hiding or revealing attributes a user can measure the amount of information he is disclosing.

Chapter 5

Leveraging Ads Audience Estimation to Evaluate Public Profile Uniqueness

Contents

5.1 Introduction

The potential to uniquely identify individuals by linking records from publicly available databases has been demonstrated in a number of research works e.g. [120, 121, 170]. In [120] Sweeney reported on the uniqueness of US demographic data based on the 1990 census and showed that, 87% of the US population can be uniquely identified by *gender*, *ZIP code* and *date of birth*. The resulting loss of privacy, i.e. the potential for

re-identification of a person's private data which may exist in any other publicly released dataset, was also demonstrated by the author. A more recent study [121] also produced similar conclusions. Therefore, anonymization of databases (*e.g.* medical records or voting registers) with the aim of protecting the privacy of individual's records when such are publicly released, in reality cannot be successful if the released database contains potentially unique combinations of attributes relating to specific individuals.

Today, with the proliferation of public online data, OSNs are a rich source of information about individuals. For either social or professional purposes, users upload various, in most cases highly personal and up to date information, to their OSN accounts. User's personal data exposure is managed by public profiles, which contain a selected (in some case mandatory) subset of the total information available in their private OSN profiles. In fact, public profiles represent an easily accessible public dataset containing user's personal details which, depending on the OSN, can include their age, gender, contact details for home and workplace, interests, etc (for a full list, see [124]).

The existence of public profiles creates a valuable new source of information that has to be considered when releasing anonymized personal records. Also, the anonymized OSN private (profile) data is being released by OSN's to profiling and advertising companies, including in some cases additional information (e.g., political orientation such as in [171]), thus increasing the number of already available anonymized datasets used e.g. for medical or other research.

These can be henceforth linked to public profiles, allowing the re-identification (and the de-anonymization) of the personal records i.e. the exposure of individual's identities[1].

Previous research has addressed the release of online data in public OSN profiles [16, 124] and re-identification mechanisms aimed at e.g. anonymized OSN graphs [46].

In this chapter, we aim to revisit the study of the uniqueness of demographics, however we consider *online public data* available for individuals. As a first step towards such analysis, we consider the evaluation of the uniqueness of public OSN profiles, consisting of the publicly available attributes e.g. *gender*, *age*, *location*, etc. associated with individual OSN accounts. We use Information Surprisal (IS) and entropy, established information theory metrics for measuring the level of information contained in random variables, to quantify the level of uniqueness. Having a higher IS of the attribute values released in the public OSN profile can be directly related to being more unique in a set of OSN users, and therefore more easily re-identifiable when combining with other publicly available datasets containing the same attribute values. Then, this work also answers the question of the appropriate selection of attributes to be included when releasing anonymized personal records.

The derived measure can also be used as a *quantitative measure of privacy*. In fact, the IS value allows the user the assess the amount of information he is revealing publicly. By hiding or revealing a set of attributes, a user can measure the amount of information he is disclosing, and hence, can take corrective privacy measures to reduce his exposure.

We note that quantifying the user's revealed information is a challenging task, as data that needs to be acquired in order to obtain a reliable estimation of profile uniqueness, is either only partially accessible (private attributes are by definition hidden), protected

[1]The policy of major OSNs is to use real names [172].

by OSNs providers, or of too large a volume for the data collection to be practical. Our study provides a novel probabilistic framework that leverages the global private attribute statistics retrieved from a major OSN Ads platform (Facebook), to obtain an *unbiased* quantification of uniqueness. We present an approach that takes user specific privacy policy into account and allows us to calculate the uniqueness of public profiles, computed over the entire Facebook dataset.

The first contribution of this work is our proposed methodology for computing the uniqueness of public OSN profiles, independently from the dataset on which the analysis is performed. This methodology can, more generally, be applied to any set of attributes that comprise a user's profile. To compute the probability of publicly revealing a combination of attributes and evaluate the measure of uniqueness, we combine statistics derived from a crawled dataset of public profiles and the Ads audience estimation platform. We consider both independence and dependence of the probabilities to reveal different attributes.

Our second contribution is that we evaluate the quantity of information carried in individual attributes and attribute combinations present in user's profiles of a major OSN (Facebook). We show that there is a wide range of values for the amount of identifying information carried by different attributes, with *gender* being the lowest with *1.3 bits* of entropy and *current city* the highest with the entropy of *13.6 bits*.

In our third contribution, we identify the key attribute combinations that contribute to profile uniqueness. Consistent with reported results for linking anonymous US datasets [120, 121] but also applicable globally, we show that the combination of *gender*, *place of residence* and *age* (directly related to date of birth used in [120, 121]) has the highest impact on the potential for re-identification of user's anonymized data. The higher information granularity available in [120, 121] and the difference in the type of community studied (online and global versus US population) results in a lower, although still significant, potential for identification. We show that 55% of Facebook users that reveal this attribute combination (around 7.7 Million) can be identified as a group of 20 and around 18% of such users can be considered unique with an information surprisal of *29 bits*.

Finally, we show the impact of user's privacy policy on the amount of information carried in public profiles and highlight how policy uniqueness contributes to potential for re-identification of users in anonymized datasets. We show that some attributes may allow users to hide in the crowd if revealed, as opposed to hiding them from public access.

The remainder of this chapter is organized as follows. In Section 5.2 we provide a summary of the datasets used for this study. In Section 5.3 we describe the methodology for computing the uniqueness of public profiles. We present results and identify the key attributes that contribute to uniqueness in Section 5.4, followed by the discussion in Section 5.5. We finally conclude in Section 5.6.

5.2 Our data source

For the purpose of our study, we have collected two datasets from Facebook: a set of public user profiles and a set of statistics collected from the Facebook Ads audience estimation platform. In the following, we start by providing a brief description of user's profile as implemented by Facebook, then we describe the methodology used to collect the data. Finally, we describe the characteristics of our datasets.

5.2.1 Facebook Profiles

Recall from Chapter 3 (section 3.2), that a Facebook profile is a collection of attributes that describe the user's personal data. An attribute might be binary (e.g., gender), multi-value from a predefined list (e.g., country) or in free form text (e.g., Interests). According to the privacy settings, an attribute can be visible to anyone, shared with (a set of) user's social links (e.g., friends) or only visible to the owner of this profile. Hereafter, we consider an attribute (resp. a set of attributes) to be *public* if it is visible to anyone and *private* otherwise.

5.2.2 Public Facebook profiles dataset

We use PubProfiles dataset presented in Chapter 3 Section 3.5.1 which was further sanitized to fit our experimentation purposes, e.g. we have used the Google Geocoding API [173] to unify the values of *country of origin* and *current country*.

5.2.3 Facebook Ads Platform dataset

Facebook offers a platform to estimate the audience of targeted Ads campaigns[2]. Advertisers can select different criteria such as user's *locations* (country or city), *gender*, *age* (or range of ages), etc.[3] These criteria can also be combined in a conjunctive manner. According to the selected combination, the Facebook Ads audience estimation platform outputs the ***approximate audience*** which represents the number of Facebook users that match the criteria. Figure 5.1 shows an example of the audience estimation on the Facebook Ads platform. Potential advertiser's selection criteria is for *gender* male, *age* of 25 and *locations* Sydney, Australia, with the corresponding audience size estimation, as highlighted by area A, being 42500. Applying an additional criteria produces a lower estimate highlighted by B in Figure 5.1.

Although there is no full report on how Facebook generates the audience values, Facebook document [174] states that it uses *all* provided information to calculate the audience size for targeted Ads which implies that both public and private attributes are utilized. The only exception is the use of IP address to determine the current location of users (i.e., *current city* and *current country*)[4]. To build the Facebook Ads platform dataset we proceed as follows. We use a subset of six attributes: *gender*, *age*, *relationship status*, *Interested in*, *current city* and *current country*. First, for every Facebook profile in PubProfiles, we extract the set of revealed attribute's values (e.g., male, New York). Then, for each extracted attribute set, we retrieve the corresponding approximate audience size using the Facebook Ads platform. In addition, we collect statistics for each attribute and for all possible attribute values (e.g., all possible locations).

[2] `www.facebook.com/advertising/`

[3] Besides aforementioned attributes, the advertiser can also target user's interests (e.g., beer and wine), interested-in (men or/and women), relationship, language, education (e.g., High School or college/Grad) and workplace.

[4] We note that users connecting to the OSN service through e.g. proxies may introduce errors into the location distributions extracted from Facebook statistics compared to actual users locations. However, we believe these users only represent a small proportion of the overall population, and as such only have a low impact on the location accuracy.

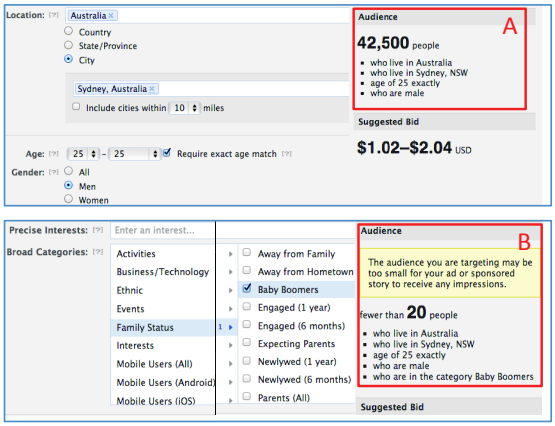


FIGURE 5.1: An example of Ads audience estimation on the Facebook Ads platform, URL: https://www.facebook.com/ads/create.

To collect the statistics from the Ads audience platform, we have developed a customized automated browser based on the Selenium WebDriver [5] which sends requests to the Facebook Ads platform with an acceptable rate. We share our collected dataset with the community on `http://planete.inrialpes.fr/projects/AdsStatistics`.

Finally, it is interesting to note that Facebook deliberately reduces the granularity of estimated audience size by only returning "fewer than 20" for audience numbers lower than 20 users (see area B in Figure 5.1). In our methodology presented in the following section, we conservatively consider "fewer than 20" as being exactly 20 users.

5.3 Methodology for public profile uniqueness computation

This section presents our proposed method to leverage the Ads platform audience estimation provided by OSNs operators (focusing on Facebook) to estimate the uniqueness of users profiles. Uniqueness of a variable is related to the amount of information a variable carries and is commonly measured by information surprisal and entropy. These are probability based metrics, therefore to compute the IS or entropy associated with a user's profile, we need a way to estimate the probability to observe the set of attribute's values comprising the profile, independently from the population of profiles we consider.

We first introduce the required theoretical background and notations used in this chapter, followed by the description of our mechanism to estimate the profile uniqueness.

5.3.1 IS and entropy computation for OSN profiles

Table 5.1 introduces the notations. We denote *Tot* as the set of *all* user profiles of a given OSN. Every user profile $u^{\mathscr{A}}$ in *Tot* comprises a set of k attributes $\mathscr{A} = (a_1, .., a_i, .., a_k)$. The profile $u^{\mathscr{A}}$ and all the associated variables may refer to a private, priv or a public, pub profile.

[5] http://seleniumhq.org/

$\mathscr{A}$	A set of attributes $(a_1, a_2, ...)$.
$V(a_i)$	The values of attribute a_i.
$u^{\mathscr{A}}$	A profile defined over the attributes in $\mathscr{A}$.
pub, priv	Denote the public and private OSN profiles.
$\varnothing^{a_i}$	The set of profiles in which an attribute a_i is not available.
$P^{\varnothing}(a_i)$	Probability that the attribute a_i is not present in a profile.
$P^{rev}(\mathscr{A})$	Probability to publicly reveal every attribute in $\mathscr{A}$ knowing that they are present in the private profile.

TABLE 5.1: Notations

An attribute a_i can be seen as a random variable, X^{a_i}, with values in $V(a_i) = \{x_1^{a_i}, x_2^{a_i}, .., x_n^{a_i}\}$ which follow a discrete probability function $P(a_i = x_j^{a_i})$. Similarly, a user's profile $u^{\mathscr{A}}$ defined on a set of k attributes $\mathscr{A}$ can be seen as the outcome of the k-dimensional random vector $(X^{a_1}, X^{a_2}, .., X^{a_k})$.

5.3.1.1 Information surprisal and entropy

IS or self-information measures the amount of information contained in a specific outcome of a random variable. IS of a user profile u which includes a set of attributes $\mathscr{A}$ is given by

$$IS(u^{\mathscr{A}}) = -log_2(P(u^{\mathscr{A}})) \tag{5.1}$$

with $P(u^{\mathscr{A}}) = \frac{|u^{\mathscr{A}}|}{|Tot|}$ *i.e.* the proportion of users having the values of $u^{\mathscr{A}}$ for the set of attributes $\mathscr{A}$. IS is measured in bits and every bit of surprisal adds one bit of identifying information to a user's profile and thus halves the size of the population to which $u^{\mathscr{A}}$ may belong.

Entropy, denoted $H(\mathscr{A})$, on the other hand, quantifies the amount of information contained in a random variable (here a multi-dimensional random vector). Entropy and IS are closely related, as entropy is the expected value of the information surprisal, i.e. $H(\mathscr{A}) = E(IS(u^{\mathscr{A}}))$. The entropy of a set of attributes $\mathscr{A}$ is given by

$$H(\mathscr{A}) = -\sum_{u^{\mathscr{A}} \in V(\mathscr{A})} P(u^{\mathscr{A}}) IS(u^{\mathscr{A}})$$

and can be seen as the amount of information carried by the attributes in $\mathscr{A}$. E.g. a user in our public dataset of $4.45 \cdot 10^5$ profiles is unique if IS reaches *19 bits*. For the Facebook population estimate, we use the value provided by the Facebook Ads platform of 722 Million users, therefore a user profile is unique with an IS of *29 bits*.

In the following, we focus on the use of the IS and entropy as a convenient way to measure the uniqueness of $u^{\mathscr{A}}$ amongst the OSN user profiles, which can be further utilized to derive the related level of anonymity of user profiles e.g. by using k-anonymity [51].

5.3.1.2 The freq method – Is PubProfiles enough

A naive approach to compute the uniqueness of profiles is to rely on an unbiased sample of the dataset of entire OSN's profiles, such as PubProfiles, and adopt a frequency-based

approach (denoted freq) to provide a rough approximation of the probability $P(u^{\mathscr{A}})$, used to compute IS and entropy. Assuming we have a dataset of $|Tot|_{crawl}$ profiles, we can then estimate the probability of each profile simply as $\frac{|u^{\mathscr{A}}|}{|Tot|_{crawl}}$ if u^A belongs to PubProfiles, and 0 otherwise, where $|u^{\mathscr{A}}|$ represents the number of occurrences of $u^{\mathscr{A}}$ in PubProfiles. In the following, we will refer to the frequency-based computation of IS as IS_{freq}, computed by:

$$IS_{\mathsf{freq}} = -log_2(\frac{|u^{\mathscr{A}}|}{|Tot|_{crawl}}) \tag{5.2}$$

This approach has at least two drawbacks. Unless all possible combinations of attribute values (as observed in the entire set of profiles Tot) are collected in the PubProfiles dataset, the frequency-based approach would provide a very coarse estimation and the IS value is lower bounded by the sample size of the dataset. Therefore if freq method is used, a maximum value of *19 bits* can be reached, as opposed to the maximum IS value of *29 bits*, based on a full dataset. For the same reason, we would not be able to estimate the uniqueness of profiles corresponding to a set of attribute values that are not in PubProfiles.

Whereas collecting such a large dataset is technically challenging, we propose a new methodology based on the audience estimation provided by the advertising systems of OSNs, which, as per Section 5.2, have access to the full set of private user's profiles.

5.3.2 Computing profile uniqueness from advertising audience estimation

Ideally, to compute IS and entropy of a set of attributes $\mathscr{A}$ that are free from sampling bias and granularity constraints, we need to know the frequency of each profile, i.e. $|u^{\mathscr{A}}|$, in the full dataset Tot. Leveraging the audience size estimation from the OSN Ads platform, we are now able to obtain such statistics that are based on entire set of OSNs profiles. As discussed in Section 5.2.3, the audience size is estimated from both public and private profiles, resulting in overestimation of frequency for public profiles. This is because user's privacy policy limits the amount of information released on public profiles, which is often significantly lower than that available in private profiles and as such $|\varnothing^{\mathscr{A}}|_{pub} \gg |\varnothing^{\mathscr{A}}|_{priv}$.

However, the bias induced by the users' privacy policy can be corrected by noting that:

$$|u^{\mathscr{A}}|_{pub} = |u^{\mathscr{A}}|_{priv} \cdot P^{rev}(\mathscr{A})$$

where $P^{rev}(\mathscr{A})$ is the probability to publicly reveal attributes in $\mathscr{A}$ knowing that they are disclosed in the private profile.

In the following, we propose two methods to compute P^{rev}, trading off accuracy of the IS estimation and measurement costs (reflected by the number of requests to the Ads audience estimation platform) as discussed in Section 5.3.2.2. These methods are respectively denoted indep and dep, as they differ in the assumption regarding the mutual independence of the probabilities to reveal specific attributes.

5.3.2.1 The indep method – assuming independence between the likelihood of revealing specific attributes

Here, we assume the probabilities to reveal selected attributes in user's public profile are mutually independent. The probability to reveal an attribute a_i, $P^{rev}(a_i)$, can then be

obtained as follows.

First, we highlight the fact that the total number of public and private profiles is equal, $|Tot|_{pub} = |Tot|_{priv}$, i.e. there will always exist a corresponding public and private user's profile.

We also observe that the number of public profiles in which an attribute is not present, i.e. $|\varnothing^{a_i}|_{pub}$, strictly depends on the probability that this attribute isn't publicly present, i.e. $P^{\varnothing}_{pub}(a_i)$ is the probability that attribute a_i is not available in public profiles, and as such:

$$|\varnothing^{a_i}|_{pub} = P^{\varnothing}_{pub}(a_i) \cdot |Tot|_{pub}$$

Similarly, we can calculate the probability that an attribute is not disclosed in private profiles as:

$$|\varnothing^{a_i}|_{priv} = P^{\varnothing}_{priv}(a_i) \cdot |Tot|_{priv}$$

The number of profiles which define a_i as a private attribute but in turn hide this attribute from public access can then be obtained from equation (5.3):

$$\begin{aligned} &|\varnothing^{a_i}|_{pub} - |\varnothing^{a_i}|_{priv} \\ &= P^{\varnothing}_{pub}(a_i) \cdot |Tot|_{pub} - P^{\varnothing}_{priv}(a_i) \cdot |Tot|_{priv} \end{aligned} \tag{5.3}$$

On the other hand, we note that $(|Tot|_{priv} - |\varnothing^{a_i}|_{priv})$ accounts for the number of private profiles where a_i is revealed, and that $P^{rev}(a_i) \cdot (|Tot|_{priv} - |\varnothing^{a_i}|_{priv})$ is the total number of public profiles where a_i is revealed. Hence, the difference $(|Tot|_{priv} - |\varnothing^{a_i}|_{priv}) - P^{rev}(a_i) \cdot (|Tot|_{priv} - |\varnothing^{a_i}|_{priv})$ accounts for the number of users who have profiles where a_i is revealed on private but not on public profiles. We can then compute:

$$\begin{aligned} &|\varnothing^{a_i}|_{pub} - |\varnothing^{a_i}|_{priv} \\ &= (1 - P^{rev}(a_i)) \cdot (|Tot|_{priv} - |\varnothing^{a_i}|_{priv}) \end{aligned} \tag{5.4}$$

Hence from equations (5.3) and (5.4) we have:

$$\begin{aligned} &P^{\varnothing}_{pub}(a_i) \cdot |Tot|_{pub} - P^{\varnothing}_{priv}(a_i) \cdot |Tot|_{priv} \\ &= (1 - P^{rev}(a_i)) \cdot (|Tot|_{priv} - |\varnothing^{a_i}|_{priv}) \end{aligned}$$

i.e.

$$P^{\varnothing}_{pub}(a_i) - P^{\varnothing}_{priv}(a_i) = (1 - P^{rev}(a_i)) \cdot (1 - P^{\varnothing}_{priv}(a_i))$$

$$P^{rev}(a_i) = 1 - \frac{P^{\varnothing}_{pub}(a_i) - P^{\varnothing}_{priv}(a_i)}{1 - P^{\varnothing}_{priv}(a_i)} \tag{5.5}$$

Note that $P^{\varnothing}_{pub}(a_i)$ is computed from PubProfiles: $P^{\varnothing}_{pub}(a_i) = \frac{|\varnothing^{a_i}|_{pub}}{|Tot|_{crawl}}$. On the other hand, $P^{\varnothing}_{priv}(a_i)$, the probability that attribute a_i is not available in private profiles, is computed from the Ad platform audience estimation dataset: $P^{\varnothing}_{priv}(a_i) = \frac{|\varnothing^{a_i}|_{priv}}{|Tot|_{priv}}$, where $|\varnothing^{a_i}|$ is not directly available but can be computed by using the aggregate number of profiles queried from the Ad platform for all possible values of the attribute a_i:

$$|\varnothing^{a_i}|_{priv} = |Tot|_{priv} - \sum_{u^{a_i} \in V(a_i)} |u^{a_i}|_{priv}$$

For example, for the attribute *age*, the number of private profiles in which this attribute is not included can be obtained by: $|\varnothing^{age}|_{priv} = |Tot|_{priv} - \sum_{j=13}^{j=65^+} |u^{age=j}|_{priv}$ (*age* can be queried from the Ads platform for a range of values between $13-65^+$, where 65^+ refers to the "Nomax" *age* attribute in the Facebook Ads audience estimation platform).

According to the assumed independence between attributes a_i, the probability to reveal every attribute in $\mathscr{A}$ is obtained by:

$$P^{rev}_{\mathsf{indep}}(\mathscr{A}) = \prod_{a_i \in \mathscr{A}} P^{rev}(a_i)$$

Finally, the IS estimation of public profile $u^{\mathscr{A}}$ using indep method can be computed using equation (5.1) as:

$$IS_{\mathsf{indep}} = -log_2\left(\frac{|u^{\mathscr{A}}|_{priv} \cdot P^{rev}_{\mathsf{indep}}(\mathscr{A})}{|Tot|_{priv}}\right) \tag{5.6}$$

5.3.2.2 The dep method – considering dependence between the likelihood of revealing specific attributes

Although the indep method offers a simple way to compute $P^{rev}(\mathscr{A})$, the estimation of probabilities can be inaccurate if the independence assumption does not hold. To verify this, we evaluate the dependence between the likelihood of revealing specific attributes, based on our PubProfiles dataset. Table 5.2 shows the calculated probabilities to reveal each of the six example attributes: *gender*, *interested in*, *relationship*, *age*, *current city*, and *country* along the rows knowing that another attribute along the columns has been already revealed. Table 5.2 also includes the overall probability to reveal specific attributes $(1 - P^{\varnothing}_{pub}(a_i))$.

$1-P^{\varnothing}_{pub}(a_i)$	0.76	0.15	0.22	0.024	0.21	0.23
	Gend.	Int. In	Rel.	Age	City	Country
Gender	1.00	0.88	0.86	0.86	0.8	0.8
Interested In	0.17	1.00	0.46	0.35	0.24	0.24
Relationship	0.25	0.68	1.00	0.48	0.33	0.33
Age	0.01	0.04	0.03	1.00	0.03	0.03
City	0.23	0.34	0.32	0.41	1.00	0.97
Country	0.23	0.35	0.33	0.43	0.99	1.00

TABLE 5.2: Probabilities to reveal attribute a_1 (rows) knowing that attribute a_2 is shown on public profile, e.g. $P(gender = revealed | age = revealed) = 0.86$

We can observe that there is indeed a correlation between probabilities to reveal specific attributes on public profiles.

To properly assess the correlation between two attributes, the probability $P(a_i = revealed | a_j = revealed)$ must be considered jointly with $1 - P^{\varnothing}_{pub}(a_i)$, the overall probability to publicly reveal a_i. The highest dependence can be observed for users' interest (*Interested In*), where users who reveal this attribute have a much higher probability to reveal any other attributes, *e.g.* the probability to reveal the *relationship status* when *Interested In* is revealed is over three times higher than the overall probability to reveal the *relationship status*.

Our aim is to provide a general framework for the estimation of public profile uniqueness. We note that the values of the probabilities from Table 5.2 may be driven either by information sensitivity and user's privacy awareness, or simply by natural dependency between attributes from a semantic perspective, however the dependency analysis is out of the scope of this work.

In the following, we present a methodology to compute $P^{rev}(\mathscr{A})$ taking into account the dependency between probabilities to reveal attributes.

Addressing the dependency between $P^{rev}(a_i)$ with $a_i \in \mathscr{A}$, requires us to compute the frequency of a disclosed combination of these attributes.

$P^{rev}_{\mathsf{dep}}(\mathscr{A})$ can be computed similarly to equation (5.5), as:

$$P^{rev}_{\mathsf{dep}}(\mathscr{A}) = 1 - \frac{P^{\varnothing}_{pub}(\mathscr{A}) - P^{\varnothing}_{priv}(\mathscr{A})}{1 - P^{\varnothing}_{priv}(\mathscr{A})} \tag{5.7}$$

with $P^{\varnothing}_{pub}(\mathscr{A})$ and $P^{\varnothing}_{priv}(\mathscr{A})$, the probability that a set of attributes $\mathscr{A}$ is not available in a public (resp. private) profile being defined as : $P^{\varnothing}_{pub}(\mathscr{A}) = P(\bigvee_{a_i \in \mathscr{A}} a_i = \varnothing)$ and

$$P^{\varnothing}_{priv}(\mathscr{A}) = \frac{|Tot|_{priv} - \sum_{u^{\mathscr{A}} \in V(\mathscr{A})} |u^{\mathscr{A}}|_{priv}}{|Tot|_{priv}} \tag{5.8}$$

We note that the computation of $P^{\varnothing}_{priv}(\mathscr{A})$, and $P^{rev}_{\mathsf{dep}}(\mathscr{A})$, requires the audience estimation of every value $u^{\mathscr{A}}$ in $V(\mathscr{A})$. This is implemented by requesting every possible set of attributes from the Ads audience estimation platform. For example, to obtain $P^{\varnothing}_{priv}(\mathscr{A})$ where $\mathscr{A}$ = {*Interested In*, *gender*}, we query the Ads audience platform for the number of profiles corresponding to all combinations of *gender* = {man, woman} and *Interested In* = {man, woman, both}.

This represents an overhead in terms of measurement costs for the dep method, as compared to the indep method which requires a fewer number of queries. However, we note that this overhead may not be prohibitive, as the audience size estimation requests are sent to the Ad audience estimation platform only once for any set of attribute values.

The IS of the public profile $u^{\mathscr{A}}$, assuming the dependency of publicly revealing attributes, denoted by IS_{dep}, can be estimated as:

$$IS_{\mathsf{dep}} = -log_2\left(\frac{|u^{\mathscr{A}}|_{priv} \cdot P^{rev}_{\mathsf{dep}}(\mathscr{A})}{|Tot|_{priv}}\right) \tag{5.9}$$

5.4 Findings on public profile attributes

In this section we study the uniqueness of users within the PubProfiles dataset, using the methodology presented in Section 5.3. We first present our results for the uniqueness based on specific single attributes disclosed in a user's public profile, followed by a study on the impact of revealed attribute combinations. Finally, we consider the impact of a user's privacy policy on the resulting uniqueness of the corresponding user. Due to space constraints, we only present a subset of calculated values, noting that these are representative of our overall findings.

We stress that our main focus is on the uniqueness resulting from the presence of specific attributes and attribute combinations in user's public profile, in line with our goal to have a generic mechanism for evaluating uniqueness.

5.4.1 Information surprisal for a single attribute

We first consider the IS and entropy (average IS) for individual attributes, calculated using the freq and indep/dep methods and based on the PubProfiles and Facebook Ads platform datasets. Note that in this section, as we are calculating IS (and entropy) for a single attribute, the IS_{dep} and IS_{indep} (and corresponding entropy) values are identical, and denoted as $IS_{\mathsf{dep/indep}}$.

Figure 5.2 (a)–(l) shows the PDF and CDF of the calculated IS values (y-axis, left and right hand side, respectively). For the sake of clarity, entropy H is included as a numerical value on top of each sub-figure (a)–(l). We also show: the number of users who hide a specific attribute, N (note the number of users who reveal this attribute is $445k - N$); the aggregate of N and the number of users with attribute values that are not available on the Facebook Ads platform, denoted N/A.

Overall, we can observe that there is a considerable difference in the range of IS and entropy values for selected attributes, with *gender* shown in Figure 5.2 (c)–(d) having the lowest and *current city* shown in Figure 5.2 (k)–(l) the highest entropy (and IS) values, respectively *1.3 bits* and *13.6 bits*. This follows the definitions of IS and entropy, which are related to the number of values an attribute may take and the number of users with specific attribute values, so higher information granularity and lower number of users for a specific value both result in a higher uniqueness. However, an absence of an attribute value may also be related to the profile uniqueness. Therefore, we also show the number of users hiding the corresponding attribute a_i and the associated IS as numerical values included above each of the sub-figures 5.2 (a)–(l).

We note the different scale of y-axis in Figure 5.2 (a)–(l), resulting from the varying value for the total number of users in the PubProfiles dataset who have disclosed a specific attribute.

Age Considering the $IS_{\mathsf{dep/indep}}$ values in Figure 5.2 (b), we can observe that over 70% (approximately 5.5k) of users have an IS value higher than *10.5 bits*, corresponding to an identifying user group size of about 500k users. This supports the conclusion that *age* is an identifying attribute which users should be careful about disclosing. In line with this, the users who hide this attribute (representing 98.4% of the total population) are highly

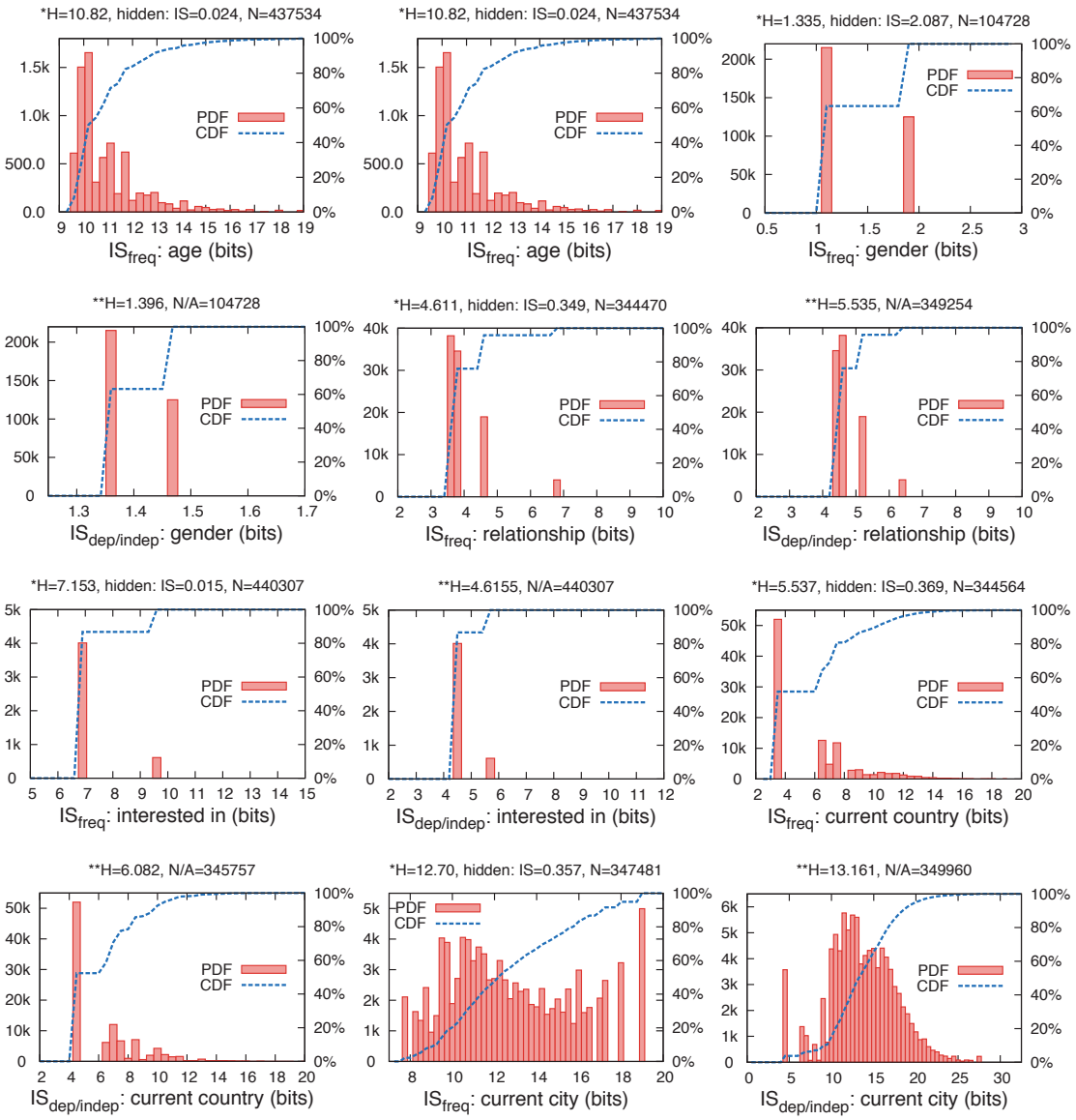


FIGURE 5.2: PDF (left) and CDF (right) of *IS* values and Entropy H (shown on top of the sub-figures) for single attribute computed by IS_{freq} and $IS_{\text{dep/indep}}$ methods (note $IS_{\text{dep}} = IS_{\text{indep}}$ for single attribute). Values are shown for: *age, gender, relationship, interested in, current city* and *country*. ** hidden IS=information surprisal for users hiding this attribute; N: number of users who hide this attribute; ** N/A: Total of N and the number of users for whom the disclosed attribute value is not available on Facebook Ads platform, e.g. age > 65*

anonymous with an IS value of *0.024 bits*.

We remind the reader that each bit of information increase in IS halves the size of the population to which the user represented by their public profile with corresponding IS may belong.

Gender We can observe that users who reveal the *gender* attribute disclose less information (with average $IS_{\mathsf{freq}} = 1.34$ *bits* shown in Figure 5.2 (c) and average $IS_{\mathsf{dep/indep}} = 1.4$ *bits*) shown in Figure 5.2 (d) than the users who consider this information private (with IS of *2.08 bits*). In Section 5.4.4 we will show the impact of hiding a common combination of attributes, including *gender*. Note that this is a highly popular attribute, with around 75% of Facebook users disclosing it in their profiles. Consequently, the population that hides it displays a high IS value for this attribute.

Relationship status The calculated IS_{freq} shows, in Figure 5.2 (e)–(f), that for more than 60% of the users, the *relationship status* reveals a low value of IS, with $IS_{\mathsf{freq}} = 4$ *bits* and $IS_{\mathsf{dep/indep}} = 4.4$ *bits*. Hiding this information has a very low associated IS of 0.35 *bits*. We note that only a subset of IS results are presented here, due to the supported values in the Facebook Ads platform [6].

Interested In We can observe in Figure 5.2 (g)–(h) that OSN users generally consider this attribute as highly sensitive and the vast majority does not disclose it, resulting in a very low IS value for such users 0.24*bits*. The average IS_{freq} values for users who display this attribute are moderate (*7.53 bits*). Similarly, the $IS_{\mathsf{dep/indep}}$ values also do not indicate high user uniqueness, with users being identifiable to within a group of 3.9 Million, only by revealing this single attribute.

Current country There is a wide range of IS values for users who have disclosed this attribute, as can be seen in Figure 5.2 (i)–(j). The average IS values are moderately high, with $IS_{\mathsf{freq}} = 5.54$ *bits* and $IS_{\mathsf{dep/indep}} = 6.08$ *bits*, while hiding this information reveals very little (0.4 *bits*). We note that 210 different countries appear as values for this attribute in our PubProfiles dataset. By examining the data values, we have observed that close to half of the total population (of those who have revealed their *current country*) have US as this attribute value. Therefore, the corresponding IS, for both IS_{freq} and $IS_{\mathsf{dep/indep}}$ methods, is low with a value of around *4 bits*. For all other users with the *current country* attribute set, the calculated IS values for both methods range between a moderate value of *7 bits* to *15 bits*, a significant amount of information which increases the uniqueness of the user resulting in an identifiable group of around 22k users.

When comparing the IS_{freq} and $IS_{\mathsf{dep/indep}}$ values, we can observe a lower IS_{freq} for the US, indicating that the IS_{freq} method overestimates the representation of US in the IS calculation.

Current city The large range of potential values for this attribute and correspondingly high potential to distinguish users intuitively flags it as sensitive personal information. We can observe from Figure 5.2 (k)–(l) that the average IS values are quite high, with $IS_{\mathsf{freq}} = 12.7$ *bits* and $IS_{\mathsf{dep/indep}} = 13.16$ *bits*, while hiding this information reveals very little (*0.4 bits*). Also, more than 75% of the users who display this attribute value lose

[6]The Facebook Ads Platform allows display of *relationship* statistic based only on a subset of values supported in Facebook profiles: single, married, engaged and in a relationship; queries based on divorced and widowed status are not supported.

more than *11 bits* (based on both IS_{freq} and $IS_{\mathsf{dep/indep}}$ values). Note than more than 20% of the users in PubProfiles reveal this information, which makes it a valuable attribute for unique identification.

5.4.2 Expected IS as a function of the number of attributes

We now consider multiple attributes in IS calculations.

Figure 5.3 shows the expected IS and the average entropy values calculated for a varying number of attributes.

We show the minimum, 25^{th} percentile, median, 75^{th} percentile and maximum of the IS values for all users. Both IS and entropy are averaged over all combinations of the selected number of attributes. As can be expected, increasing the number of disclosed attributes results in higher IS and entropy values and the corresponding amount of revealed information about the users. In the following subsection, we will explore in more detail the specific attribute combinations which will result in higher IS values and therefore present a higher privacy risk for users.

Comparing the results obtained using the three calculation methods, in Figure 5.3 we can observe that the values of IS_{freq} are consistently lowest for all attribute combinations, followed by IS_{dep} and IS_{indep}. As previously discussed, IS_{freq} presents a rough calculation of values, which can be used as an indication of the relevance (to privacy) of both attributes and attribute combinations. Increasing the complexity of obtaining data (i.e., the number of required queries from the Facebook Ads platform) increases the accuracy of the result. Consequently, the IS_{indep} values can be calculated for the combinations not present in the collected dataset. However, this method results in higher IS and entropy values than what is obtained by the more precise IS_{dep} method, which in turn requires the highest amount of information from the Facebook Ads platform.

We can observe the most significant difference in the IS and entropy values obtained by different methods when considering the users who have revealed six attributes in Figure 5.3 (b). The IS_{indep} and IS_{dep} values reach an average entropy higher than *25 bits*, representing a corresponding uniqueness within a set of 22 users, while the IS_{freq} value underestimates IS and only reaches *19 bits* of entropy with a significantly lower corresponding unique user set of around 1300 users. Although there may be a number of factors contributing to the low IS_{freq} values, the most relevant one is the dependency of the frequency-based entropy estimation on the used dataset. Regardless of the large size and the unbiased sample of the full dataset that we have used for IS_{freq} calculations, a number of combinations of attributes among the profiles may still be missing and will influence the result.

5.4.3 On the relevance of disclosed attribute combinations

We now consider the IS values for different attribute combinations, enabling us to draw conclusions about the dominant (and less relevant) parameters contributing to privacy loss. Figure 5.4 shows the cumulative distributions function of the IS, for: (a) IS_{freq}, IS_{indep} and IS_{dep} with all six attributes considered; (b)–(d) IS_{freq}, IS_{indep} and IS_{dep} for selected attribute combinations that were shown to have extreme (both low and high) IS values. Similarly to the results shown in Figure 5.3, we can observe that revealing 6 attributes (regardless of their values) results in a high IS value for the majority of users, e.g. observing

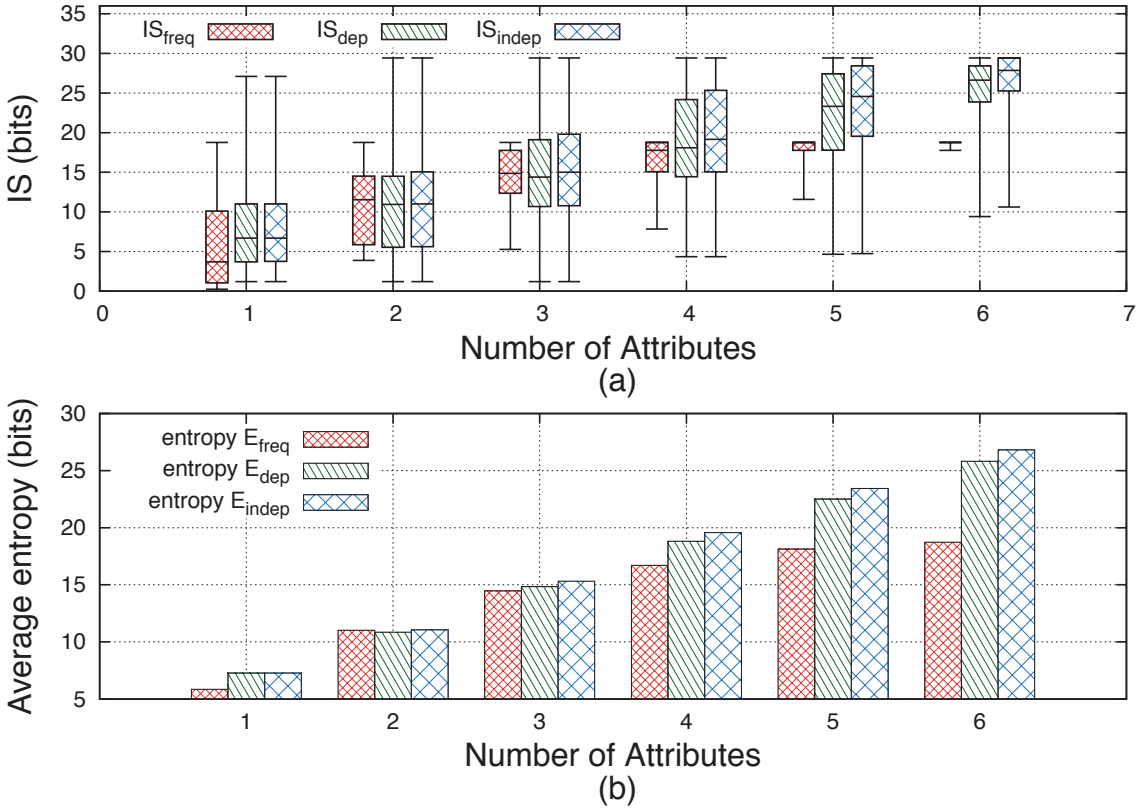


FIGURE 5.3: (a) Information surprisal and;(b) Expected entropy for a varying number of attributes.

the IS_{indep} and IS_{dep} CDF values in Figure 5.4 (a), more than 80% of the OSN population with six disclosed attributes has IS of more than *22 bits*. This represents users uniquely identifiable within a set size of 170.

Considering specific attribute combinations shows the importance of having a carefully considered personal privacy policy with selectively disclosed attributes. Figures 5.4 (b)–(d) indicate that users should be wary of concurrently disclosing the combination of *age*, *gender* and *current city*, as this reveals almost as much information as as the total of six disclosed attributes. Although the granularity of our data is significantly lower (only age is available in the PubProfiles dataset, as compared to full birth date in the dataset used in [120, 121]) and we study a different community (global and online, as compared to US only and based on Census data in [120, 121]), our results are in line with the previously published studies on the uniqueness of demographics [120, 121], which show that the combination of *gender*, *ZIP code* and *date of birth* has a very high uniqueness. We can observe in Figure 5.4 (c) that around 55% of users have IS of around *25 bits* and can therefore be identified in a set of 20 users. Further to this, around 18% of users can be uniquely identified, having the IS value of their public profile at *29 bits*. This represents a significant potential threat, as the corresponding number of Facebook users is around 7.7 Million for being identifiable to within a set of 20 and 2.7 Million for unique identification.

On the other hand, revealing the *gender* and *interested in* may not be harmful from a

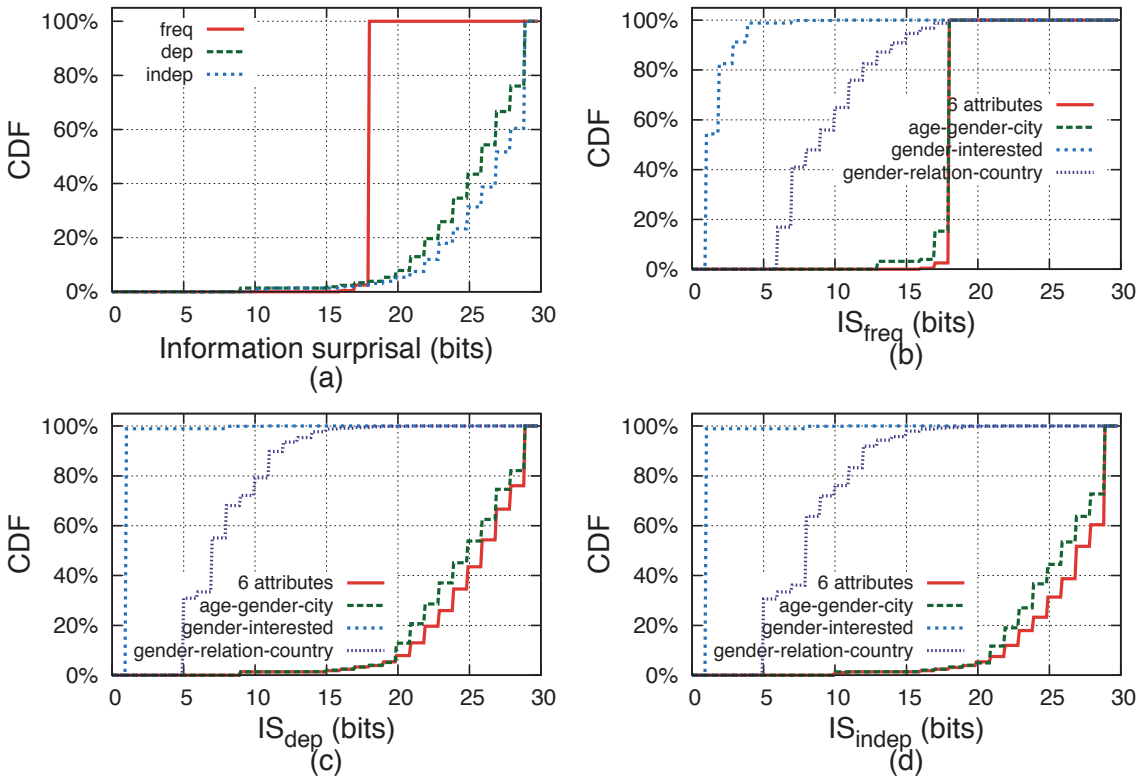


FIGURE 5.4: Multiple attribute information surprisal distribution

privacy perspective for most of the OSN users (IS is less than *5 bits* for 90% of the users). Disclosing the *relationship status* along with the *gender* and *country* of residence reveals a higher amount of information, with more than 70% of users loosing at least *11 bits* of IS.

We note that, although our results have been derived for a sample of the Facebook population, the unbiased nature of the sample (as argued in Section 5.2) makes them applicable to the whole Facebook population.

5.4.4 Impact of privacy policy

This section complements the understanding of the key parameters that influence the resulting IS and entropy values of users' profiles, by studying the potential impact of users' privacy policy.

The likelihood to reveal specific attributes varies significantly amongst user profiles and there are some combinations that users may potentially prefer to hide (the dependency between probabilities to reveal pairs of attributes, shown in Table 5.2 partially illustrates this). One consequence of users' restrictive privacy policies is that other users revealing a rare set of attributes may be more easily identifiable, independently from the values of the attributes (i.e., regardless whether the attribute's values are rare). To verify this claim, we show in Figure 5.5 the information surprisal IS_{dep} as a function of the P^{rev}_{dep} (note the log scale on the x-axis).

As expected, the lower the probability with which users reveal attributes, the higher

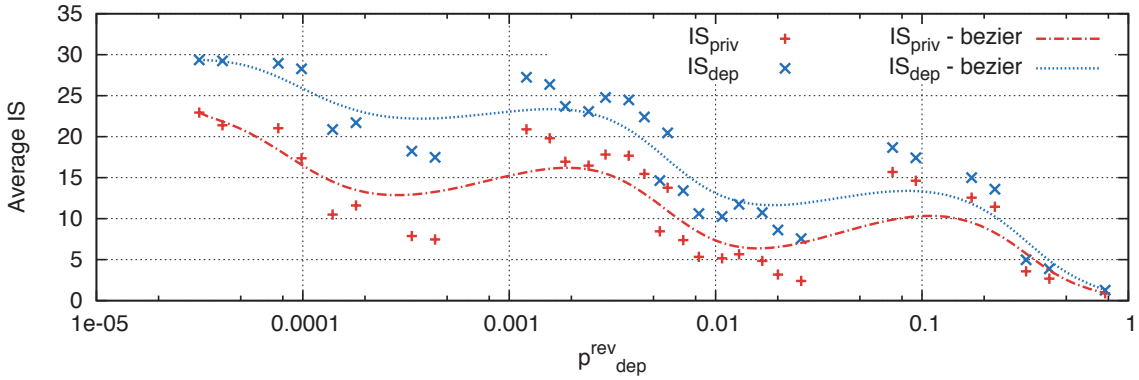


FIGURE 5.5: Average IS as a function of P^{rev}_{dep}

the value of IS_{dep}. However, to understand whether the increase in profile uniqueness is contributed by the attributes' values themselves or by the hiding of these attributes (setting restrictive privacy policies), we also include in the figure the values of IS_{priv} denoting the information surprisal of the user's private profiles (i.e., reflecting the profile uniqueness contributed to by all the attribute values). IS_{priv} is obtained from the Facebook Ads platform statistics, for each set of attributes corresponding to a specific value of IS_{dep}. For improved clarity of the figure, we also plot the Bezier approximation of the data.

We again observe the general trend for both IS_{dep} and IS_{priv}: A decrease in P^{rev}_{dep} values corresponds to an increase in average IS_{dep} and IS_{priv} values, which illustrates that independently from the attributes values, the more a set of attributes is hidden the more unique the corresponding profiles will be. In other words, a restrictive privacy policy is also a good identifier of profiles. On the other hand, we highlight an interesting observation that can be made from Figure 5.5, which shows that the gap between the IS_{dep} and IS_{priv} also increases as P^{rev}_{dep} decreases. This result suggests that the more users tend to hide a combination of attributes, the more identifying this set of attributes will be for other users that do reveal it. The paradox here is that when a combination of attributes becomes too rare, because of a majority of users choice to hide it, it also becomes very identifying when revealed on other profiles. Similarly, when a combination of attributes is very common, hiding it becomes identifying amongst the rest of the crowd.

5.5 Discussion

We show that it is possible to estimate the quantity of information carried by a combination of attributes shared on OSNs profiles. This is accomplished by leveraging the Ads audience size information from the OSN itself, to compute the probability that a selected set of attributes would be revealed publicly by users. In fact, P^{rev} can be seen as a way to measure users' willingness to publicly share attributes they have revealed in their private profiles.

Consider a scenario where a user wanting to protect their privacy creates a pseudonym-based profile, and hides some of the attributes from public access. While this may be thought of as a suitable approach to being anonymous, our findings show that it may not

always be effective, as the level of anonymity will depend on the chosen set of attributes and attribute combinations, which will carry a varying level of identifying information. We also show that revealing selected attributes may be more effective towards achieving anonymity than hiding them, as this approach would allow users to hide in the crowd of similar users.

Applications based on the proposed methodology We envisage a number of applications for our profile uniqueness computation framework. First, we plan to develop an online service where users would submit a profile to be evaluated in regards to the potential for unique identification of the information they would display publicly. The service would provide an indicator to inform users whether the set of attributes they are revealing (or hiding) is identifying. The users would therefore be able to measure the efficiency of the applied privacy settings and choose the desired trade-off with the usability of their public profile.

Second, our framework can be used by OSNs to design better strategies to release datasets of OSNs records to e.g. the research community. To this effect, by understanding which combinations of attributes disclose a high amount of information about users, OSNs operators could carefully select public profiles with low IS, which will in turn have a low likelihood to be linked to external records and thus deanonymized.

Potential extensions As previously mentioned, a frequency-based approach to compute profile uniqueness is dependent of the collected public profiles i.e. the captured set of combinations of attributes. It is therefore impractical, as e.g. it cannot estimate the uniqueness of a profile with a combination of attributes absent from the dataset. An alternative approach could be to adopt a smoothing-based (e.g., Good-Turing [175] or Laplace [176]) frequency estimation technique that can be used to predict the occurrences of the non-observed combinations of attributes, by relying on observed distributions of individual attributes. However, a smoothing-based method still cannot take into account the dependence between the likelihood of revealing specific attributes.

In this work, we have deliberately chosen to concentrate on a fixed set of attributes. We illustrates the extent to which the chosen attributes may identify the users revealing them, and it is important to note that our methodology is general enough to be extended to other attributes that might be shown on users profiles. E.g., users' interests have been shown to be good predictors of other attributes [6] and as such may carry a high level of information about users. We also did not include users' full names in this study, as the application scenarios we consider focus on anonymized profiles where the identity of users is unknown.

5.6 Conclusion

This chapter proposes a novel method to compute the uniqueness of public profiles independently from the dataset used for the evaluation. We exploit the Ads platform of a major OSN, Facebook, to extract statistical knowledge about the demographics of OSN users and compute the corresponding IS and entropy of user profiles. This is used as a metric to evaluate profile uniqueness and hence the magnitude of the potential risk to

cross-link the specific (OSN based) public information about a user with other public data sources. Our findings highlight the relevance of choosing the right combination of attributes to be released in user's public profile and the impact of user's privacy policy on the resulting anonymity.

This chapter concludes the first part of this thesis which treated privacy concerns related to information sharing. In the next part, we study a second aspect of privacy in OSNs namely privacy threats related to the OSN ecosystem. Specifically, we show how OSN interactions with third and fourth parties can cause information leakages and breach user privacy.

Part II

Information Leakage In OSNs

Chapter 6

Big Friend is Watching You: Analyzing Online Social Networks Tracking Capabilities

Contents

6.1 Introduction

The popularity and widespread of OSNs has increased in recent years to a level unparalleled by any other Internet service. While these services are freely accessible, the viability of their business model relies on collecting users' personal data for targeted advertising and in some cases data monetisation [116].

OSN user's profiles represent a rich source of personal information about the users, including the demographic information, their interests and social relations. Throughout the last three chapters, we studied the threats to privacy resulting from this direct exposure of personal information. In fact, by choosing to publicly reveal some attributes, the user might be the target of inference attack that discloses some hidden or missing data. This information can further be used to carry other attacks such as guessing his password.

However, a second source of privacy diffusion and exposure of information about individuals is the third party tracking of user's Internet activity (i.e., the visited web sites). A number of research studies (e.g., [3]) investigate the mechanisms used for web tracking by third party services including the Ads networks, analytics companies and content distribution networks, and the increasing widespread of such mechanisms on the Internet.

In this chapter, we focus on the emerging players in the web tracking arena, the OSN services. The OSN "share" feature provides both an incentive for web site owners to include it on their sites, as it may result in their increased popularity, and a way for OSNs to track user activity on these sites. We investigate in detail the mechanisms for tracking user's visited web sites and the magnitude (widespread) of the tracking activities for the three most popular OSNs: Facebook (FB), Twitter (TW) and Google+ (G+). We demonstrate how the tracking is done even when the users are logged out of a specific OSN or, surprisingly, when they do not have an OSN account. We further demonstrate the risks from such tracking by an experimental study showing how the majority of user's web profile can be constructed based on the information tracked by OSNs.

Our contributions include: (i) Highlighting of the mechanisms by which the three most popular OSNs track visited sites and showing that even without an OSN account a user is still vulnerable to tracking. (ii) Evaluation of the widespread of user tracking by the three major OSNs, based on the Alexa top 10000 sites. (iii) Based on real traffic traces, demonstration of the real potential that an OSN provider may construct a large part of user's web profile; we present examples where up to 77% of user's web profile is constructed by OSNs only by tracking user's activity.

The organization of this chapter is as follows. Section 2 presents the techniques deployed by the three majors OSNs to track users. Section 3 describe the widespread of such mechanism in top10K most visited web site as classified by Alexa. Section 4 presents an analysis on real web traffic and demonstrates the efficiency of tracking system on building accurate user profile.We conclude in section 5 and present ideas for future work.

6.2 Privacy elements

This section describes the implementation of the OSN Tracking Mechanism (we will call hereafter OTM or bugs interchangeably) by presenting its main components. We then describe each technique of the studied OSNs and evaluate each of their tracking capabilities.

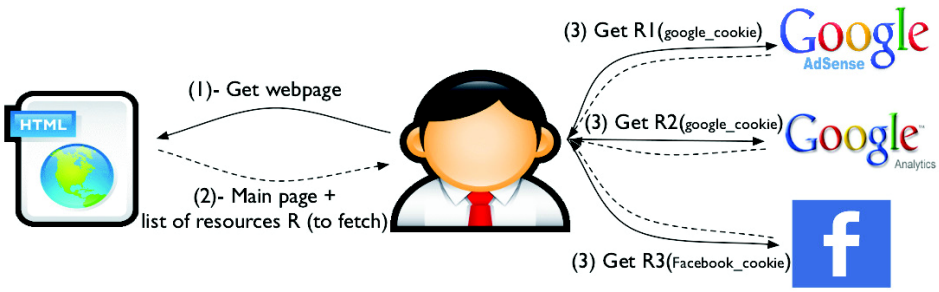


FIGURE 6.1: Retrieving web content

6.2.1 Preliminary Operations

As illustrated in Figure 6.1, when a user visits a web site, the fetched page typically generates several other HTTP connections. These connections, which aim to get additional components into the loaded web page, can be classified in two groups: i) connections belonging to the user's visited web site (usually referred to as first party connections), and used to fetch "legitimate" resources such as images or css (e.g., step 2 in Figure 6.1). ii) other connections established to third party entities for the purpose of content serving, Ads, tracking, analytics measures and statistics computation or a combination of thereof (step3). As such, these third parties can be Ads Networks (e.g., Google's Adsense, Yahoo! yieldmanager, AOL) or analytics companies (e.g., Google analytics, Microsoft/aquantive) or content distribution networks (e.g., akamai, yimg), but also, as this chapter concentrates on, Online Social Networks (e.g., Facebook, Twitter, Google+).

In the case of an OTM, these connection triggers are embedded into the fetching of the "Share" button resource. In fact, an increasing number of websites provide a button to share their content within the visitor's favorite OSN. These buttons, and *without any user specific interaction*, establish connections and send information about the user to the OSN provider. We stress that the user does not necessarily interact with the OTM resource, and without his consent can be tracked by the OSN. As we will show it next, this OTM operates even if the user is not logged-in the OSN and, even if he never registered to that OSN. Similarly to [3] we examine the disclosure of web traffic activities to third parties. However, these "new" kind of third parties have a huge amount of personal information about the user. Therefore, armed with this new knowledge, they can build a better profile.

In general, the OTM is built on well-known building blocks, namely iframes, div or XFBML and cookies. Besides how OTM resources are embedded into the visited webpages, in the following we briefly describe how the webpage's url is transmitted to the OSNs. Then, we examine, for each of the OSNs, its specific implementation of the cookies-based tracking mechanism.

6.2.2 Transmitting the url of the webpage

When loading the OTM resource the user browser typically transfers the current webpage address to the OSN third party either via the referrer attribute of the HTTP request or as a simple url parameter. For example, when fetching `www.cnn.com`, an HTTP request

is sent to Facebook. As shown in the request (below), the url parameter *origin* specifies the origin of the request transmitted to the OSN (the article that the user is accessing).

```
www.facebook.com/extern/login_status.php
?api_key=[...]app_id=[...]&origin=http%3
A%2F%2Fedition.cnn.com
```

Similarly, the referrer record as shown below reports the url of the website the user accesses.

```
GET www.facebook.com/plugins/likebox[...]
Referer: http%3A%2F%2Fedition.cnn.com
```

6.2.3 Cookies-based tracking

In addition to the webpage url, OSNs need to *uniquely* identify the user to bind this information to potential previously gathered data (e.g., from user's profile). Cookies are the *de facto* technique that is typically used to report user's activities while ensuring unique tracking identity. In this section, we examine specific cookies-based tracking mechanism that we observed for three of the major OSN actors: Facebook, Twitter and Google+.

OSN	Facebook		Twitter		Google+	
Cookie name	*d_atr*	*c_user*	*twid*	*guest_id*	SID	PREF(ID)
Logged In	Yes	Yes	Yes	Yes	Yes	Yes
Logged out (But having an account)	Yes	No	No	Yes	No	Yes
No account (But have visited the OSN website)	Yes	No	No	Yes	No	Yes
No account (Never visited the OSN website)	No	No	No	Yes	No	No

TABLE 6.1: Cookies Transmission Strategy

6.2.3.1 Facebook

Facebook stores 16 different cookies for an active session. The role of each one is hard to determine but two of them are made to identify users: *d_atr* and *c_user*. The *d_atr* cookie contains a random string (24 characters) and is set when a user visits Facebook.com. If a user does not have a Facebook account but has once visited Facebook website, this user will also have this cookie set. The cookie is valid for two years, which suggests that Facebook has the ability to track all users that have once visited Facebook (and even without registering to the service) for a relatively long period of time. This information of the "past" could then theoretically be linked to the user upon registration of an account using a unique web browser.

As shown in Table 6.1, when a user is logged in, the user's browser sends both *c_user* and *d_atr* to Facebook and hence the user is uniquely identified. When the user logs out, the *d_atr* cookie is not deleted and the browser keeps on sending it to Facebook when loading OTM resources. By matching the *d_atr* with *c_user*, Facebook is able to track user's activities who have logged out.

To summarize Table6.1's entries for Facebook, we note that we observed that Facebook has the ability to track all users that have visited `facebook.com` at least once, and logging out from the service does not help users to avoid being tracked.

6.2.3.2 Twitter

Twitter stores 15 different cookies for a logged-in user. Again, although it is difficult to determine the role of each cookie, we note two cookies that are used as a user identifier: *twid* and *guest_id*.

guest_id is set independently from whether the user is logged-in or not, as opposed to the *twid* cookie, which is only present during a user's active session. These two cookies can be used to link the activity of logged-out users with his real Twitter ID. Moreover, Twitter shows an interesting behavior when a user who has never visited the Twitter website visits a website which contains a Twitter "Share" Button: Twitter automatically set the *guest_id* cookie and hence may track users that never visited `twitter.com`. This behavior is unique, since none of the studied OSNs behaves similarly.

6.2.3.3 Google+

Google stores 28 different cookies for a logged-in user. These cookies belong to multiple domains and sub-domains of Google. Multiple cookies are used to uniquely identify user, among which a few are sent through an encrypted connections (e.g., *SSID*). We identified and studied two of those that are sent in clear, *SID* and *PREF*. The latter contains a random string variable *ID* that uniquely identifies the user.

When the user disconnects, the *PREF* cookie is not deleted and is still transmitted to Google when the browser loads a Google+ OTM resource. As such, theoretically, Google is still able to track the web activities of logged-out users. To achieve this, Google has to match the *ID* value from the *PREF* cookie to the user's profile, which is possible since both the session-id carrying *SID* cookie and the *ID* carrying *PREF* cookie are together visible to Google while the user is logged-in.

Summary

An interesting finding when digging into the cookies-based OTM is that all the studied OSNs provide a mechanism (theoretically capable of) tracking users who have explicitly logged-out from the service. Moreover, users that did not register into these OSN services are still trackable, since OTM allows also to keep web browsing traces of users prior to their registration to the system, assuming they visited the OSNs domains once. Twitter exhibits a unique, yet surprising tracking behavior and is theoretically designed to track users that do not have accounts, and never visited the twitter domain either.

6.3 OTM Coverage of Alexa top 10000 Sites

In this section, we analyze to which extent OTM is diffused across the web. We consider the top 10K most popular websites in which we evaluate the OTM distribution. We also compare OSNs tracking possibilities to Google tracking mechanism.

We extracted the top 10200 most visited websites according to Alexa [1] from which we removed 128 entries belonging to search engines. The remaining list was crawled and all subsequent HTTP requests recorded. We further removed 138 web pages containing Javascript and/or flash programming bugs. The final list of analyzed websites contains a total of 9933 different urls (which we will refer hereafter as top10k websites), out of which 7173 representing 72% of the considered websites include at least one tracking element in their main page.

We used a modified version of Ghostery [119] to tag tracking and ad companies.

Google [2] tracking system is embedded in nearly half of the websites (43.29%). This presence is mainly due to Google Analytics and Google AdSense. Facebook with both its "Like" button and "Facebook Connect" plugins is second with a surprising coverage of 22.2%. Google+ and Twitter are represented by 10.4% and 7.25% of the webpages, respectively.

6.3.1 Ranking Coverage

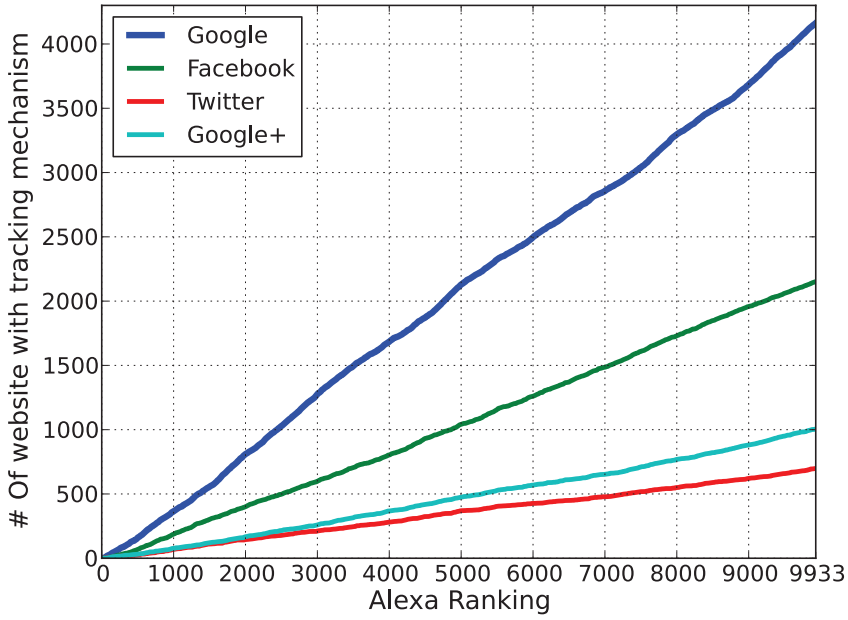


FIGURE 6.2: Ranking Coverage

As a first step to measure the diffusion of OTM, we draw the distribution of the number of detected OTMs as function of the website rank in Figure 6.2. This distribution suggests a uniform diffusion of OTMs through the top10k websites. As observed previously, Facebook has the highest coverage. In particular, it covers nearly 500 websites out of the top 2000 websites whereas Twitter and Google+ cover only 145 and 168 respectively.

[1] `www.alexa.com`

[2] We used en.wikipedia.org/wiki/List_of_acquisitions_by_Google to tag all services that belong to Google

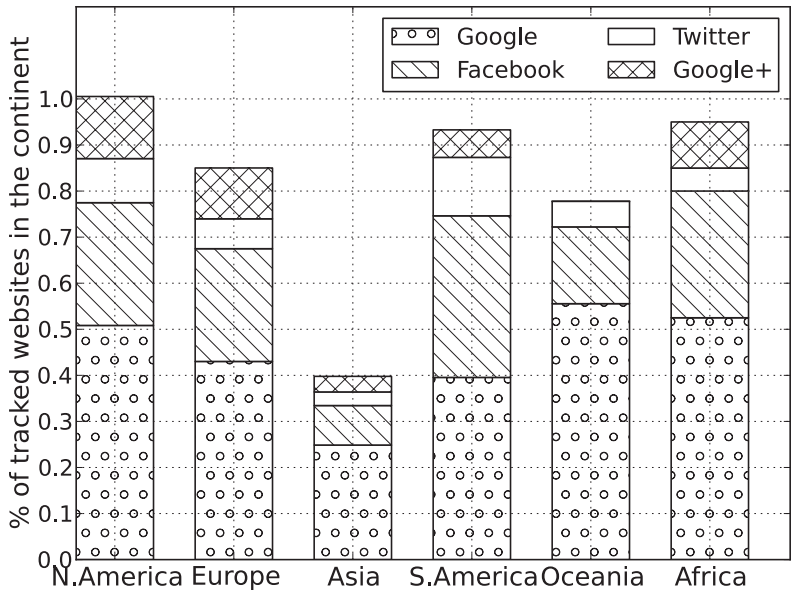


FIGURE 6.3: Proportions of tracking mechanisms per webpages continent-based origin.

Surprisingly, these OTMs are not concentrated in most visited websites. In particular, among the top 1000 sites, only a few dozen of sites embed OTMs. Two reasons may explain this observation. First, the sites geographical origin[3] may have an impact on such a distribution, since the studied OSNs are not as spread in Asia as in Europe and U.S (see Figure 6.3). Second, the website content and categories may be not suitable to embed such OTMs, since OTM main targets are more likely to be interactive online services such as Blogs or News. From a geographic perspective, OTMs target all origins as shown in Figure 6.3. In this Figure, the y-axis represents the percentage of websites belonging to the continent (illustrated in the x-axis) that contain at least an OTM. For example, 50% of North American websites in our dataset contain a Google tracking mechanism whereas this coverage reaches only 22% of Asian websites. Not surprisingly, Asia is poorly targeted by the studied OSNs, which can be explained by the poor popularity of these in Asia due to the existence of local concurrent services (Cyworld in south Korea, or Sina Microblog in China, etc.).

The general trend observed above is still valid, with Google tracking system being the most represented through all the continents, and Facebook leading the OTM diffusion independently of the continent.

6.3.2 Category Distribution

In this section, we study the distribution of OTMs according to the category of the website. We used [177] to extract the websites categories (e.g., `cnn.com` is categorized

[3] We define the site origin as the origin of its main audience (i.e., The geographic location of the majority of its users)

as News). We retrieved 81 categories from the top 10K websites, and then computed the distribution of OTMs as a function of these categories.

We observe that among the total set of categories, Facebook covers the majority with 68 categories (83%), the most prominent being General News (14%), Entertainment (11.6), Internet services (5.8%) and Online shopping (5.5%). Similarly, Twitter covers 60 different categories, where the most illustrated categories are Entertainment (12.5%), General News (12.2%), Internet Services (7%) and Blogs (5.6%). Finally, Google+ is also covering almost 80% of the categories (64), but interestingly it is well represented in different categories such as Pornography (5.6%).

This shows that OTMs are widely spread and not restricted to a small number of categories. This confirms that if collected, OTM-based information would depict an accurate user profile. Hence, the collected data rise concerns about users privacy, since it is collected without neither the user consent nor his knowledge.

6.3.3 SSL-based connections

Transmitting cookies to OSN servers may rise security concerns if the connection is insecure (e.g., unencrypted WiFi connection). Hence, transmitting these cookies over SSL is preferable. We observed that Google+ uses SSL in the vast majority of the cases, and only a few requests (16%) are still sent in clear. Twitter and Facebook OTMs differ significantly from Google+'s behavior with 96% and 96.5% of the traffic containing the transmitted cookies sent in clear, respectively. This behavior may endanger the privacy and security of users since an attacker my easily perform a session hijacking.

6.4 Analyzing Real Traffic Traces

So far, we have studied the diffusion of OTMs, and observed that they are embedded in a wide range of websites and hence can be effectively used to track users outside the OSN sphere. Now, we concentrate on quantifying how much information is actually collected in practice.

6.4.1 Used Dataset

We captured all the HTTP headers (requests and responses) transiting through our lab local network, for a period of a week starting from the 20th of October 2011. We anonymized the observed traffic by hashing the source IP, and constructed two datasets.

- *Dataset1* containing all the traffic with 687 different IP addresses. It contains more than 27 million connections to nearly 55K different destinations.

- *Dataset2* For the purpose of our experiments in Section 6.4.3, we reduced Dataset1 so that the number of overall destinations is reasonably low. In fact, since we are classifying websites into categories, we were unable to categorize all the 55K destinations and hence chose to randomly sample a subset of users from *Dataset1*. Reducing the number of users allows us to keep the history characteristics (length, diversity) whereas reducing the number of destinations to classify. This reduced dataset, called *Dataset2* contains 69 IP addresses that made over 16 million connections to 17539

different destination servers. We further filtered out Ads servers and static content providers, which reduced the set of destinations to 5712 different addresses.

6.4.2 Who's connected?

In the following, we use *Dataset1* to estimate the proportion of users logged-in to the considered OSN services. We capture the transmitted cookies to estimate such statistics. Although, as mentioned in section 6.2, cookies are not reliable to determine whether a user has an account or not, establishing whether a user is connected to the service by observing the session cookies still fit for purpose, since the later is transmitted by the browser only when the user is logged-in.

We observed that 48% of the users in our dataset have a Facebook account and 33% were logged-in at least once during our measurements. The number of users with a Twitter account is significantly lower than the number of Facebook users, with only 195 users (28% of the total number of unique IP addresses) observed to have an account an only a small fraction (4%) being logged-in at least once.

We do not present results for Google+ since we did not observe a significant number of cookies transmitted in clear, due to the usage of SSL by default in most of Google services.

The relatively small proportion of logged-in users we observed, may be explained by the IT awareness of users we monitored in our dataset. Most users here are computer scientists and thus aware of potential tracking threats, and may use different techniques to mitigate such threat by using cookies blocking/cleaning plugins, enforcing HTTPS connections or using private mode browsing.

6.4.3 OSN profiling

So far, we showed that theoretically, OTMs are an efficient way of tracking users. In this section, we aim to study to which extent this mechanism can be used to "construct" users' profiles. From *dataset2*, we construct simple profiles based on either the web history of the user or on the visited websites' categories and then observe how much of these profiles can be reconstructed by the OSN.

6.4.3.1 Methodology

Most profiling techniques rely on classifying websites into categories to overcome the huge amount of information generated by user's web browsing. Based on these categories, a user Profile can be constructed [178].

For the sake of profile construction simplicity, we define a user profile P_u as the union of all websites categories visited by a user u. From *dataset2*, we extract all destination hosts and used [177] to categorize the visited websites, 98.44% of which have been successfully classified. Note that in our experimentation a website may be classified into up to three different categories.

Each website visited by a user u belongs to at least one category $c_j (1 \leq j \leq C)$ (the total number of observed categories is 81 in our dataset) . The set of all visited categories represent the user Profile (P_u) whereas the set of visited websites (i.e., the user web history) is denoted by H_u. Furthermore, HC_u denotes the proportion of H_u retrieved by either Facebook or Twitter using their respective OTMs. PC_u represents the fraction of

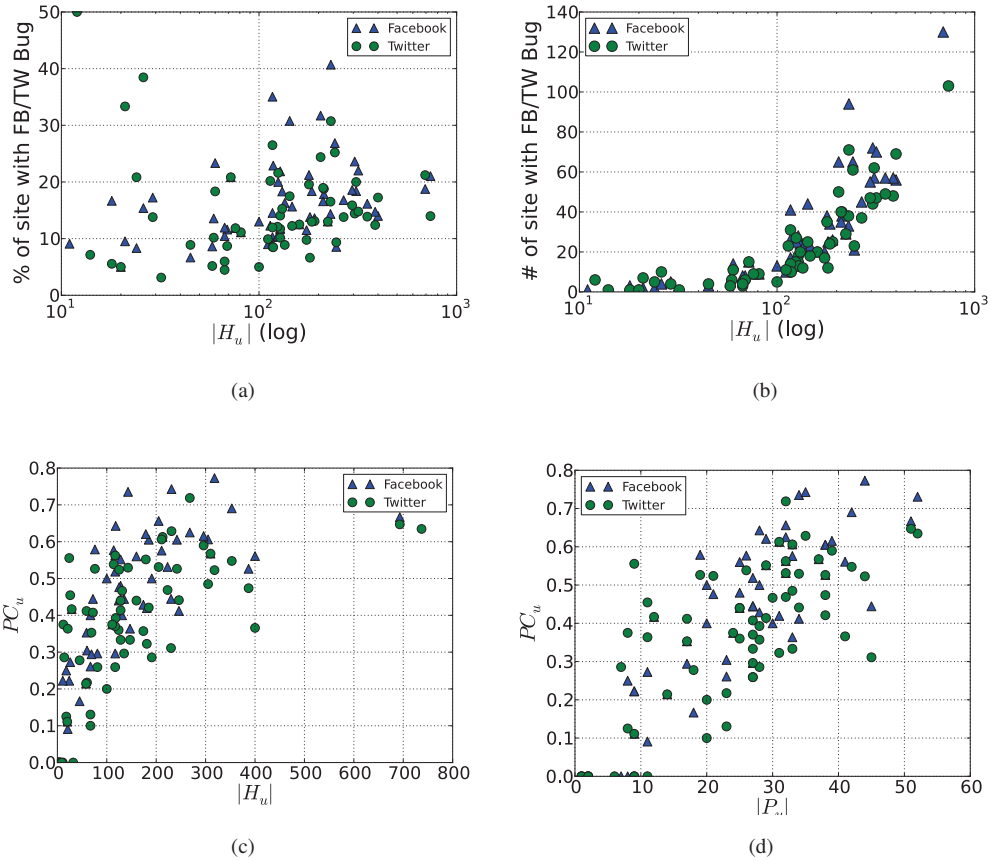


FIGURE 6.4: From left to right (a) History size VS History coverage(%); (b) History size VS History coverage(#); (c) History size Vs Profile coverage; and, (d) Profile size Vs Profile coverage.

P_u that Facebook (resp. Twitter) is collecting through the OTM. For instance, if a user visits `cnn.com`, `kernel.org` and `foxnews.com`, `cnn.com` and `foxnews.com` are classified as News and `Kernel.org` as Software. Since only `cnn.com` has a Facebook OTM, PC_u is 50% whereas HC_u is 33.3%. Figure 6.5(a) shows the distribution of users according to the number of categories in their profiles. We observe that 75% of the profiles have between 10 and 40 different categories.

6.4.3.2 User Web History Analysis

Figure 6.5(b) shows the CDF of HC_u. For example, more than 50% of users have at least 15% of their web history contain an OTM from both Facebook and Twitter. It also shows that at most 10% have more than 25% of their web history tracked by the two OSNs. A surprising result is that the coverage of Twitter and Facebook are slightly different as opposed to our results observed in Section 6.3.1 (Alexa case).

Second, we analyzed the relation that might exist between the size of web history $|H_u|$ and HC_u. Figures 6.4(c) and 6.4(d) depict our finding. Most users have HC_u between 10% and 25% whereas we observe a large variation of HC_u for small history size. Thus, a small

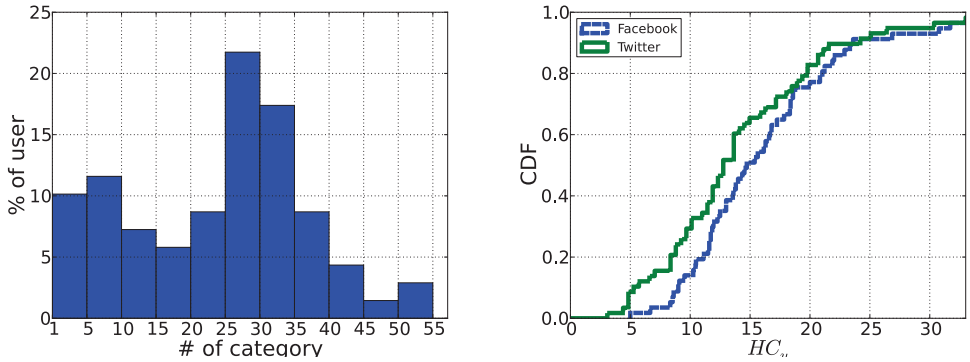


FIGURE 6.5: Profile length distribution (left),CDF of history coverage (right)

history size does not necessarily entail a small coverage but rather depends on the category of visited websites. Moreover, as $|H_u|$ increases, so does HC_u. Thereby, users with large history sizes tend to be more "efficiently" tracked than others.

Finally, we analyzed the correlation between the history size $|H_u|$ and the profile coverage PC_u. As shown by Figure 6.4(c), the larger $|H_u|$ is, the higher PC_u is. These two last results were expected since large history size implies that some of websites contain an OTM with high probability. A second observation is the large variation of PC_u for users having small history sizes. For instance, different users with $|H_u| = 40$, can have as different PC_u value as 0, 22%, 30% or 68%. As for the HC_u case, a small history size does not necessarily implies a small profile coverage. On the other hand, this variation decreases with a larger history size. In fact, most users with $|H_u|$ larger than 200 have a PC_u value higher than 40%.

6.4.3.3 User Profile analysis

Previously, we showed that users with larger web history tend to be more tracked than others. Nonetheless, we also note that even users having small history size are still tracked but with a larger variation. In this section, we examine the correlation that might exist between the profile size $|P_u|$ and the profile coverage PC_u, which is depicted in Figure 6.4(d). It indicates a similar trend to the relation observed for the history size. In essence, users with a large $|P_u|$ tend to be more easily trackable whereas other users with a small $|P_u|$ exhibit a larger variation of PC_u. However, for the the vast majority of users, Facebook and Twitter are able to reconstruct a very accurate category-based profile. In fact, PC_u varies from 0.4 to 0.77.

To dig more into the reasons that would explain our findings, we ask whether users with a high PC_u value browse similar web content (i.e., similar categories). To answer this question, we compute a similarity matrix as follows: for all users, we extract the set of all P_u such that $PC_u \geq \beta$, where β defines the level of reconstructed profile coverage. Then, we compute the Jaccard index between all profiles in that set. Recall that the Jaccard index is computed as $J(i,j) = \frac{|P_i \cap P_j|}{|P_i \cup P_j|}$. For example, a Jaccard value $J(i,j)$ of 0.5 means that user i and user j share the half of their respective profiles P_i and P_i. Figure 6.6 shows the CDF of the Jaccard index for different values of β (i.e., 0.5, 0.6 and 0). First, the red

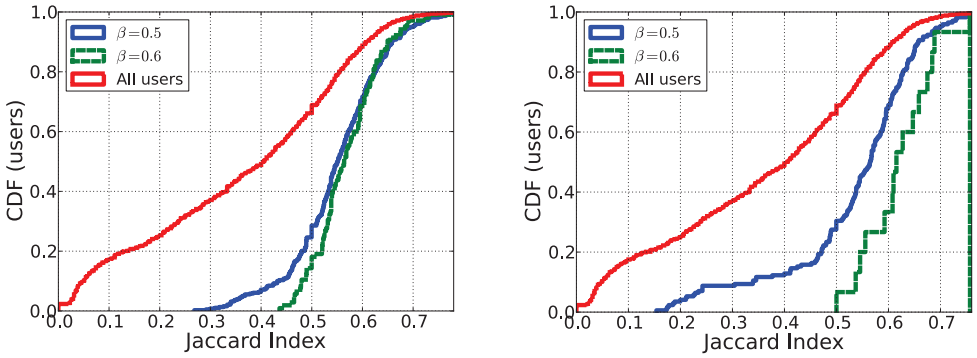


FIGURE 6.6: P_u similarity using Jaccard index (FB left, TW right)

curve depicts Jaccard values for all users: we observe that nearly 80% of the users have a similarity value lower than 0.5 which suggests a weak correlation between user profiles. However, blue ($\beta = 0.5$) and green ($\beta = 0.6$) curves show that there is a high correlation between users who are highly tracked. For instance, all users who have PC_u higher than 0.6 for Twitter have a Jaccard index above 0.5. This means that all these users share at least 50% of their profiles. This clearly indicates that highly tracked users tend to have similar profiles.

6.5 Discussion

Since OSNs do already maintain a tremendous amount of information about their users, it may be argued that tracking mechanisms could barely affect users privacy. Nonetheless, many websites when visited may leak sensitive and personal information about users. For instance, when a user visits `www.sparkpeople.com`, he might not be aware that such an information about his web browsing activity, which can be considered highly sensitive as it may refer to users health, is being reported to Facebook. Among the websites in our Alexa dataset, 34.4% of the websites classified as potentially providing Health information do embed at least one OTM.

On the other hand, users' navigation patterns can be used to divide customers with similar preferences into several segments through web usage mining and then recommended targeted ads as shown in [179].

Since OTM are cookie-based tracking mechanism, solutions such as cookies deletion after each session or private mode browsing can be a first approach to mitigate the threat. However these countermeasure suffer from at least two drawbacks. First, navigation patterns belonging to one session are still tracked. Second, more aggressive tracking techniques (e.g., through IP address and browser fingerprint [118]) can still be applied. A more suitable solution consists in blocking the OSN "iframes" connections, and as such all connections to the OSN servers. Tools like Ghostry [119] implement such a mechanism as a browser Plugin.

6.6 Conclusion

This chapter presents insights about a new tracking mechanism that can be used by OSN providers to collect web browsing information about their current users, and, as we have demonstrated, about potential future users. We observed that these increasingly popular mechanisms cover a broad range of content ranging from Blogs, Health to Government websites. Specifically, we draw attention to the potential privacy threat that may rise from this accumulation of private data that may be utilized for user profiling, without the user consent. We showed that this data, if collected, can draw an accurate profile of the user interests.

Chapter 7

A Closer Look at Third-Party OSN Applications: Are They Leaking Your Personal Information?

Contents

7.1 Introduction

Previous chapter showed that tracking techniques used by OSN operators allow them to silently and persistently trace the user navigation. However, the leakage of personal information from the OSN itself to third parties remains an open question. This chapter answers this questions by examining the interaction between third-party OSN applications, OSN platform and external entities (fourth parties) and exposing the information leakage.

Third-party OSN applications are tremendously popular with some apps being actively used by more than 100 million users in Facebook. Besides, with apps potentially having access to users' personal information, through access permissions, they introduce an

alternative avenue for privacy leakage. With the users' personal information being exposed outside of the OSN sphere, the privacy risk becomes even higher.

We examine third-party OSN applications for two major OSNs: Facebook and RenRen. These third-party applications typically gather, from the OSN, user personal information, such as user ID, user name, gender, list of friends, email address, and so on. Third-party applications also typically interact with "fourth parties," such as ad networks, data brokers, and analytics services. According to Facebook's Terms of Service, third-party applications are prohibited from sharing users' personal information, collected from Facebook, with such fourth parties. We develop a measurement platform to study the interaction between OSN applications and fourth parties.

We use this platform to analyze the behavior of 997 Facebook applications and 377 applications in RenRen. We observe that 98% of the Facebook applications gather users' basic information including full name, hometown and friend list, and that 75% of apps collect the users' email addresses. We also find that the Facebook and RenRen applications interact with hundreds of different fourth-party tracking entities. More worrisome, 22% of the Facebook applications and 69% of the RenRen applications provide users' personal information to one or more fourth-party tracking entities.

7.2 Background

This section introduces the general concepts behind third-party applications for both Facebook and RenRen networks. For concreteness, we discuss these concepts in the context of Facebook, and point out how RenRen differs at the end of this section.

As shown in Figure 7.1, while logged into the OSN, the user selects an app, which brings the user to a web page that includes a "canvas" served by Facebook, the application (in an iframe) served by the publisher's server, and possibly some advertisements served by ad networks. If it is the user's first visit, Facebook displays a dialog frame which asks the user for permissions to access information in the user's profile (step 1). This dialog frame indicates the particular set of permissions the application is requesting. The application can, for example, request permission for "basic info" [1], which includes user name, profile picture, gender, user ID (account number), and user networks. Applications also have access to friend lists and any other information users choose to make public (e.g., interests and notes). In order to access additional attributes, or to publish content to Facebook on behalf of the user, the application needs additional permissions.

The user's browser then contacts Facebook seeking an *access token*, which is used to query Facebook servers to fetch the user's information (steps 2 and 3). The token is transmitted to the publisher's server (step 4), which queries Facebook for user information (steps 5 and 6). Once the server obtains the user information, it may load all or some of that information in the HTML (for example, using JavaScript) provided to the user's browser.

OSN applications typically further interact with "fourth parties" such as ad networks, data brokers, and analytics services. Different techniques can be used to contact these external entities, among which include using an iframe (e.g., loading an ad) and Javascript (e.g., sending data to an analytics service). Observe that when these entities are contacted,

[1] `https://developers.facebook.com/docs/graph-api/reference/user/`

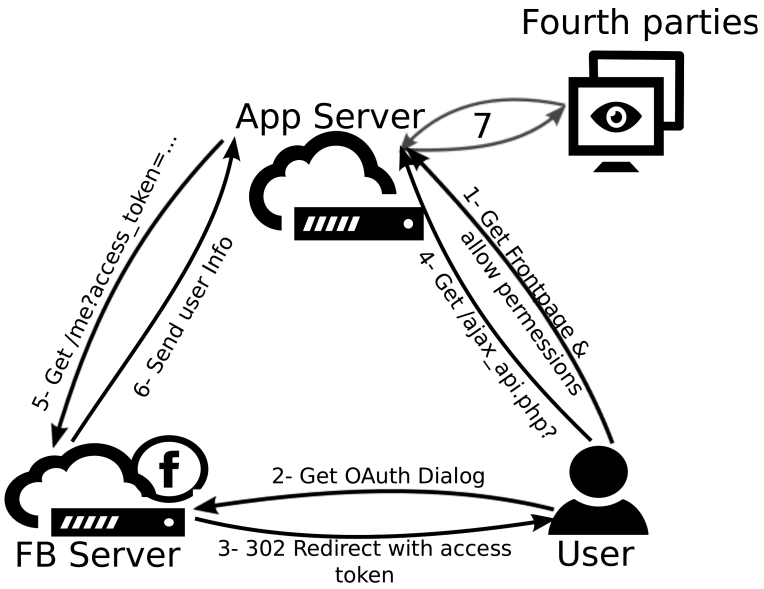


FIGURE 7.1: An overview of the Facebook application architecture

the referrer field is automatically filled with the current page (i.e., application main page) URI. Our focus in this chapter is on these external entities and whether the personal information obtained by the user's browser is transferred to the external entities.

From an architectural point of view, RenRen has the same conceptual features and operation as Facebook with a few minor exceptions. In particular, RenRen has only three permissions: (i) access personal information and friend relations, (ii) access timeline information (e.g., posts, shared content) and (iii) allowing the app to post on behalf of the user. The first permission is granted by default to all applications.

Privacy issues: Third party applications naturally give rise to several privacy issues. First, the application code is hosted on the publisher's own servers and are out of Facebook's control. This inherently prevents Facebook from monitoring and/or controlling the application's behavior, and impedes any proactive measures to block malicious activities. Second, as user information is transferred out of Facebook servers, user information usage and dissemination is out of the user's control. Finally, privacy control for third-party apps are very limited due to the coarse-grain granularity of permissions, and as such it is debatable whether this is in accordance with the "principle of minimal privilege" which states that only minimum privileges should be granted to fulfill the task.

7.3 Methodology

In December 2012, we investigated each of the 997 working applications listed on the official Facebook App center.[2] To be referenced by the Facebook App center, the

[2] `https://www.facebook.com/appcenter/`

application needs to be reviewed and sanctioned by the Facebook staff.[3] As a result, most of the applications considered in our study are very popular, as we discuss below. For each of these applications, we first obtain the application name, ID, installation URL, popularity (in terms of number of users), category (e.g., game, Health & Fitness, Finance, etc.), publisher (which was not available for a few applications) and a summary description. We then automate the process of application installation based on the Selenium WebDriver.[4] In particular, using several different Facebook accounts with distinctly different user profiles, we install and accept the requested permissions for each of the 997 applications. To monitor the application behavior, we use a modified version of a Firefox plug-in [4], allowing us to record all the HTTP and HTTPS traffic. Similarly, we investigated each of the 377 working applications listed on the RenRen App center.

7.3.1 Limitations of the methodology

In our experimental methodology, we aim to measure and characterize third-party applications in a semi-controlled environment. We note, however, that our tested applications are all gathered from the official App center and as such do not represent the totality of the OSN third-party application ecosystem, since there are many other applications that do not belong to the App Center. For the privacy leakage analysis, our methodology only examines traffic originating from the user browser; any information leakage that might happen outside this channel (e.g., communication between the application servers and external entities) are not identified. Therefore, the extent of privacy leakage quantified in this work serves as a *lower bound*.

7.3.2 Basic characteristics of applications

Our main interest centers on the applications' interactions with external fourth-party servers and resulting privacy leakages. To this end, it is useful to first understand the basic characteristics of the Facebook and RenRen applications under investigation. Specifically, in this subsection, we examine the popularity of the applications, the applications' publishing companies, and the permissions the applications request.

Application popularity: Figure 7.2(a) shows the cumulative distribution for the popularity of our tested Facebook and RenRen applications. We observe that both distributions exhibit a similar shape with 60% of the Facebook and RenRen applications having more than 100 thousands users and 10 thousands users, respectively. Importantly, the most popular applications have more than 100 million users in Facebook and more than 10 million in RenRen. This shows the potential of third-party applications to collect large volumes of user data.

Application companies: We were able to collect the publisher's company name for 845 applications. Table 7.1 shows the top seven publishers among the applications considered. These top seven companies only cover 10% of the applications; furthermore, there are 536 different publishers for the 997 tested applications. From a data retention perspective, this suggests that user data is more likely to be scattered among multiple entities, further reducing the user's control over its data.

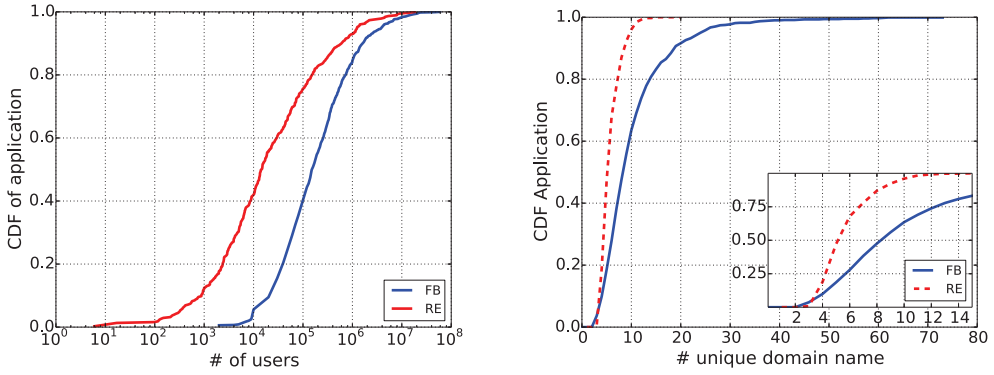
[3] `https://developers.facebook.com/docs/appcenter/guidelines/`
[4] `http://seleniumhq.org/`

6waves	2.35%
Zynga	1.66%
Playdom	1.37%
Peak Games	1.17%
Kingnet	1.07%
Electronic Arts	1.07%
MindJolt	0.98%

TABLE 7.1: Most frequent app companies for Facebook (997 apps)

User basic information	98%
Personal email address	74.5%
Publish Action	59.6%
Access user's birthday	33.6%
Publish stream	20.5%
Access user's likes	12.25%
Access user's location	9.8%

TABLE 7.2: Most frequently requested permissions for Facebook (997 apps)



(a) (b)

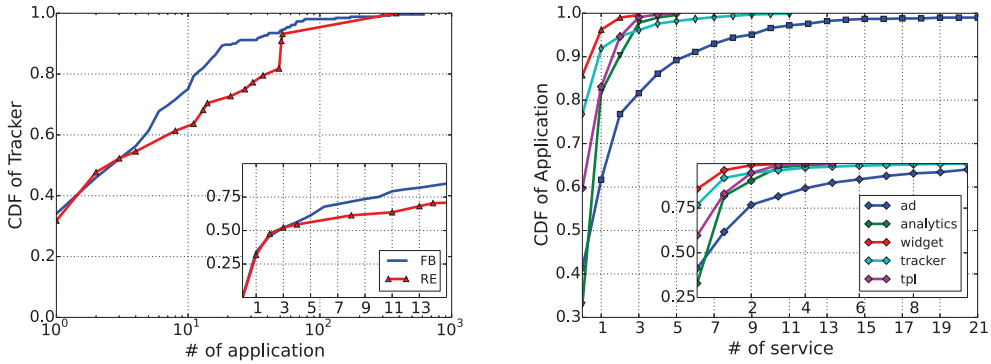
FIGURE 7.2: (a) Application popularity, (b) Number of contacted servers for each application

Permissions: Table 7.2 shows the most frequently requested permissions. As expected a large fraction of applications request permission to obtain the basic information, which as mentioned previously, encompasses not only the user name but also all information the user makes public in the public profile. Requests for access to email information is also very frequent (75%). Sensitive information such as user' birthday and hometown seems to be less requested with 33% and 10% respectively.

7.4 Interaction with external entities

Now we turn our attention to the interaction between the third-party application running in the browser and external entities. Since most, if not all, of the functionalities are very similar in Facebook and RenRen networks, we mainly discuss features of the Facebook network. However, the figures show our results for both OSNs.

HTTP Connections For an application to function properly in Facebook, the user browser has to contact three main domains: the Facebook login page at `facebook.com` to exchange credentials, the Facebook content server at `fbcdn.net` to extract user data



(a) (b)

FIGURE 7.3: (a) Tracker distribution for third-party apps (b) Distribution of tracker categories

(e.g., the user's photo) and the application's main server. For each of our tested applications, we capture the traffic exchanged between the browser and the external entities, and extract the external domains with which the application communicates. Figure 7.2(b) shows the CDF of the number of unique contacted domains per application for both Facebook and RenRen OSNs. Surprisingly, more than 75% of the Facebook applications exchange traffic with at least six different domains, and for almost 10% of the tested applications the number of unique domains exceeds 20.

The RenRen network exhibits a slightly different behavior with 70% of the tested applications contacting less than 6 servers. The maximum number of domains contacted by RenRen applications is four times smaller than in Facebook. This suggests that the tracking eco-system in RenRen is less complex and includes a smaller number of entities.

Tracker distribution Many of these external entities are "trackers," including ad networks and analytics services, which are contacted when the user visits the application webpage. To identify the tracker domains, we use lists provided by Ghostery, AdBlock and the Microsoft Tracking Protection List (TPL) to compile a set of 10,292 tracking domains. The total number of trackers identified within our set of Facebook and RenRen applications is 410 and 126, respectively. Their distributions are shown in Figure 7.3(a). These results show that for Facebook (respectively, RenRen), 39% (respectively, 37%) of the tracking domains are employed by a single application. The tail of the CDF also shows that a few trackers are employed by a large number of applications, with less than 5% of the trackers in both Facebook and RenRen tracking more than 100 different applications. The top 3 of the observed trackers is composed of Google Analytics (with 613 tracked applications), smartadserver (416 applications) and Turn.com (344 apps).

Tracker Categories We now further classify the set of identified trackers into five categories: ad networks (e.g., Google Adsense) referred to as Ad; analytical services (e.g., Google analytics) referred to as analytics; online service plug-ins (e.g., Twitter connect) referred to as widgets; ad-network tracking services as special tracking features (e.g.,

DoubleClick Floodlight), which are referred to as trackers. Finally, we also consider the trackers not belonging to these classes but included in the Microsoft Tracking Protection List (Scorecard Research) and refer to these as tpl.

Figure 7.3(b) shows the cumulative distribution of the trackers according to their different categories (only for Facebook). As expected, we observe that more than 70% of the applications use analytics services. Notably, Google analytics is employed by 60% of the applications, far ahead of all other analytics services. Note that 84% of the applications use a single analytics service, and only 2% of use more than 3 different analytics services.

More than 60% of the tested applications did not use a known "ad network", which puts in question the revenue model of these applications. There are numerous ways for a Facebook application to generate revenue: inserting ads from a particular ad-network (which is the case for 40% of our tested applications); by monetizing the "pay more, play more" scheme which allows the users to buy virtual credits; by selling private advertising space (e.g., through the Facebook exchange protocol FBX); or selling user data, although this is officially not compliant with the application development agreement. We highlight that the high proportion of applications not relying on ad-network revenue is surprising, which merits further investigation.

The sharp slope of the ad CDF curve shows that a large fraction of the applications that use ad-networks tend to include a variety of different networks; in particular, 10% of the applications embed at least 5 different ad-networks. Other types of trackers are less popular and most of them are employed by a single application.

```
GET /api/.../?s=USERID&g=male&lc=US&f=1...
Host: api.geo.kontagent.net
```

```
GET /__utm.gif?..&utmhn=iframe.
onlinesoccermnager.nl&utmul=en-us
&...& utmhid=110829611
&utmp=userName, ProfilePicture,email
,Network First/LastName, USERID
Host: www.google-analytics.com
```

Domain	Leaking	Total
kontagent.net	60	66
ajax.googleapis.com	38	480
google-analytics.com	36	624
6waves.com	18	30
socialpointgames.com	13	16
mindjolt.com	9	10
disney.com	8	9
adobe.com	6	183

FIGURE 7.4 & TABLE 7.2: Left: (top) Information leakage to Kontagent, (down) Information leakage to Google Analytics – Right: Number of leaking Facebook apps vs. total number apps contacting this domain

7.5 Personal Information Leakage

In this section we present a methodology to detect potential privacy leakage from Facebook and RenRen apps to fourth parties. We then employ this methodology to quantify the amount of privacy leakage.

7.5.1 Methodology

Our methodology is as follows. First, we create multiple user accounts with distinctly different profiles (i.e., attribute values). For each of these accounts, we then automatically install and run the apps and record the network traffic. We then examine, for each app and user pair, whether the HTTP requests are transferring user information to fourth parties. For instance, to assess whether a user's gender is leaked, we check all requests that transfer the string "male" for a male user and "female" for a female user. While this approach allows us to automatically search for personal data leakages, encrypted or encoded data are not detected as we only use string matching. We further checked the API documentation of known services (e.g., kontagent and Google analytics) to assess the meaning of parameters observed in the traffic.[5]

7.5.2 Data leakage classification

The process of leaking information to external entities can be categorized into two types: *intentional* and *unintentional*.

Intentional information leakage In this scenario, the app developer intentionally transmits user information to external entities (usually analytic services) by embedding user data into the HTTP request. The total number of Facebook apps that are leaking user info intentionally is 183. In the following, we study two representative examples:
Kontagent This company presents its business as helping customers "derive insights from app data in ways beyond traditional analytics." Kontagent provides detailed statistics about app usage. To achieve this, the app sends a set of user attributes to Kontagent; the API specification[6] provides a set of functions for transferring user data, among which are year of birth, country of origin, or friend count. Note also that the API allows the transfer of any other type of data as an associative array. Figure 4 shows how user ID, gender and location are transferred to Kontagent.
Google Analytics As with Kontagent, some developers are using Google Analytics to generate statistics about app usage. To do so, they embed user data inside the request to Google Analytics. This data can then be used (in Google dashboard) to derive statistics. Figure 4 shows how data is transferred.

Unintentional data leakage A website may unintentionally leak personal information to a third party in a Request URI or referrer. Krishnamurthy *et al.* [16] examined this problem for 120 popular websites and found that 48% leaked a user identifier. We consider user information to be leaked unintentionally if it is transferred through the referrer. In fact, the referrer is automatically filled by the browser; thus data leakage through it is generally the result of poor data sanitisation. The total number of applications leaking info through the referrer field is 79.

scorecardresearch.com	170	377
sinaapp.com	61	64
google-analytics.com	38	51
doubleclick.net	36	51
baidu.com	23	69
linezing.com	12	13
friendoc.net	10	10

TABLE 7.3: Number of leaking RenRen apps vs. total Number apps contacting this domain

Info	# App
user ID	181
Name	17
Gender	72
Country	2
City	2
Age	10

TABLE 7.4: Information leaked by Facebook apps

# leaked attribute	# Apps
One or more	220
2 or more	48
3 or more	14
More than 3	0

TABLE 7.5: Number of attributes leaked per application

7.5.3 Statistics

Table 7.4 shows the number of applications that leak various user attributes. More than 18% of apps transmit user ID to an external entity. While this information seems harmless, in fact querying Facebook Graph API[7] with the User ID allows the external entity to gather all public information about the user (i.e., username, full name, link to Facebook profile, hometown, gender, and so on). Moreover, as the user ID is unique, it can be used to track a user across different apps. Finally, there is substantial evidence that user ID (and username) can be used to (re)identify a user [123]. We observe that 1% of apps are transmitting age to an external entity; this attribute is considered highly sensitive and only few users disclose it publicly [180]. Finally, the low value for country and city (only two apps are leaking this info) can be explained by two facts: First, some apps are using IP-geo location to identify the user location.[8] Second, Facebook provides a more coarse grained attribute that determines the user language (e.g., fr_FR). In a second step, we analyzed how many attributes are leaked per application. Table 7.5 shows that 220 applications (22%) leak at least one attribute, 48 leak at least 2 attributes and 14 more than 2.

The question remains: To whom is this data being transferred? Table 2 answers this question. From a domain perspective, three main categories are sharing data gathering in the top 10 domains: analytics services (e.g., Kontagent), social app companies (e.g., 6waves) and entertainment companies (e.g., disney).

Table 2 shows that analytics services are way ahead of the others for data gathering. However, there is a significant distinction between them. Kontagent's main goal is to draw statistics from social apps and as such is inherently dependent on the user data that the app is leaking. This can clearly be seen by the large proportion of apps that are using Kontagent and are leaking user information (60 apps out of 66). On the other hand, the Google service is not expressly designed to derive statistics about social apps but is instead adapted to this task. Not surprisingly, a relatively smaller percentage of applications using

[5]For instance, Kontagent is using a parameter g=m for transmitting the gender (male)

[6]`https://github.com/whydna/Kontagent-API---ActionScript3-Wrapper`

[7]`http://goo.gl/KlOL8`

[8]Facebook is using IP-Geo location in its ad platform to determine user location

a Google service are leaking user information.

Social app companies are ranked second (6waves, socialpointgames and mindjolt). This can be explained by the app publishing process. For instance, 6waves is the company behind the Astro Garden app. However, this app is not hosted under the 6waves domain but rather under `redspell.ru`. As such, 6waves is considered an external entity as it is not the app main page. To centralize data gathering, this company sends back user data to the main corporate server (e.g., 6waves.com) which explains the data leakage. Note that using this process, companies like 6waves can track users across multiple applications. Finally, entertainment companies such as Disney and Adobe are ranked third.

Disney is gathering data in a systematic way which is shown by the high number of apps that are leaking data (8 out 9). As such, Disney is collecting data from different (affiliated) apps and collecting the data in a centralized way. Adobe, on the other hand, is receiving the user information unintentionally. This claim is confirmed by the small number of apps that are leaking data (6 out of 183). In most cases, the information is transmitted to Adobe in the referrer when loading the Flash player.

7.5.4 RenRen leakage

At a first glance, RenRen apps appear to be privacy preserving as no user data is transferred to fourth parties. However, a deeper look shows that the situation is much worse than for Facebook. Recall from Section 7.2 that the app receives an *access token* from the OSN operator, and this token is then used to query the OSN for the user data. Our measurements reveal that 69% of RenRen tested apps are transmitting this token to external entities. This behavior represents a major privacy breach as external entities "inherit" the app privileges and can therefore query RenRen on behalf of the user. Table 7.3 shows the top external domains receiving the access token. In contrast with Facebook, the leaked information is sent to both Chinese and US tracking companies.

7.6 Discussion and Conclusion

Several third party applications are leaking user information to "fourth" party entities such as trackers and advertisers. This behavior affects both Facebook and RenRen with varying severity. 22% of tested Facebook applications are transmitting at least one attribute to an external entity with user ID being the most prominent (18%). While in 183 applications the user information is intentionally transmitted to fourth parties (e.g., through an API call), some leakages are the result of a poor data sanitization and hence can be considered unintentional. On the other hand, RenRensuffers from a major privacy breach caused by the leakage of the *access token* in 69% of the tested apps. These tokens can be used by trackers and advertisers to impersonate the app and query RenRen on behalf of the user.

While user information is transmitted to several entities, some major players might represent a bigger risk. For instance, Google is able to track 60% of Facebook applications and receives some user information from 8% of them. In RenRen, the situation is even worse, as 45% of tested apps transmit the full user profile to a single tracker (`scorecardresearch.com`). Hence, a single social networking app might lead to

users being tracked across multiple websites with their real identity. Web tracking in combination with personal information from social networks represents a serious privacy violation that shifts the tracking from a virtual tracking (i.e., the user is virtual) to a real "physical" tracking (i.e., based on user personal information).

Chapter 8

Conclusion & Perspectives

In spite of the numerous benefits of Online Social Networks, they inflict terrible privacy costs. This thesis aimed at quantifying these risks by analyzing several privacy breaches. Our work is composed of two main parts that shed light on privacy issues from both sides: user and OSN operators.

In the first part, we studied how the data supplied to the OSN can be exploited by a malicious attacker to invade user privacy and even break its online security.

In our first work, we demonstrated that an attacker who has only access to public interests can exploit this knowledge to infer – with high accuracy– other sensitive information about the user [6]. Specifically, we presented a semantic-driven inference technique that exploits music interests —which are often disclosed by users — to infer user attributes. To do so, we proceeded in three steps: First, we augmented the user music interests with information obtained from Wikipedia. Second, we extracted Interests Feature Vectors (IFV) using the Latent Dirichlet Allocation (LDA) generative model and the augmented interests. Finally, we computed the similarity between users according to the IFV. The inference step makes the assumption that similar users are likely to share the same attributes. Hence, we predict a user attribute by computing the most probable attribute of the n closest users to him in the IFV space. Through extensive experiments and using both a public and a private dataset, we showed that our approach predicts hidden attributes with a high accuracy.

These inferred attributes can be exploited in several other attacks such as spear phishing [10, 146] and Sybil nodes creation [181]. Our second contribution [37] shows that this personal data can also be used to speed up the process of password guessing.

Our work has two main contributions. From a security perspective, we proposed a novel password cracker based on Markov models that outputs passwords in decreasing order of probability. Through extensive experiments, we showed that it is able to crack more than 80% of passwords, more than all probabilistic password crackers we compared against. From a privacy perspective, we formally investigated the impact of *additional personal information* on the cracking success ratio. We provided an extension to our algorithm that exploits such information and assess its performance. Our results show that the speed gain

can go up to 30% for passwords that are actually based on personal attributes. Our findings suggest that privacy breaches can have a disastrous security consequence as they might allow an attacker to break into the user account.

These studies called for a better measure of privacy exposure. To answer this question, we developed a practical, yet formally proved, method to estimate the uniqueness of each profile by measuring the amount of information carried by public profile attributes [39]. To this end, we elaborated a methodology to compute the Information Surprisal (IS) of each public profile —independently from the used profile dataset— using the Ads audience estimation platform of a major OSN (Facebook). Our measurements, based on an unbiased sample of more than 400K Facebook public profiles, show that the combination of *gender*, *current city* and *age* can identify close to 55% of users to within a group of 20 and uniquely identify around 18% of them. We believe that our approach enhances privacy in at least two scenarios: (i) it provides a *quantitative privacy measure*: by hiding or revealing attributes a user can measure the amount of information he is disclosing (ii) data providers can quantify the risk of re-identification (i.e., identifying a user in a dataset) when disclosing a set of attributes.

In the second part of this thesis, we analyzed the privacy issues related to the OSN's ecosystem. We analyzed the privacy issues that arise from the interactions between the OSN's platform and several other third and fourth parties.

We started by presenting tracking techniques that are deployed by three major OSN providers to trace user navigation pattern. First, we studied the mechanisms by which the three most popular OSNs track visited sites and showed that even without an OSN account a user is still vulnerable to tracking. Second, we evaluated the extent of user tracking by the three major OSNs based on the Alexa's top 10K sites and demonstrated that it is widespread. Finally, based on real traffic traces, we demonstrated the real potential of large scale user's profiling and history reconstruction by showing that OSNs are able to construct up to 77% of user's web profile.

The second part dealt with information leakage from OSN to third and fourth parties. By examining the application ecosystem of both Facebook and RenRen, we demonstrated that these complex interactions lead to information leakages. Specifically, we showed that several third party applications are leaking user information to "fourth" party entities such as trackers and advertisers. This behavior affects both Facebook and RenRen with varying severity. 22% of tested Facebook applications were transmitting at least one attribute to an external entity with the user ID being the most prominent (18%). On the other hand, we showed that RenRen suffers from a major privacy breach caused by the leakage of the *access token* in 69% of the tested apps. These tokens can be used by trackers and advertisers to impersonate the app and query RenRen on behalf of the user. This represents a serious privacy risk as the tracking is shifted from the virtual world (i.e., tracking an Internet user) to the physical one (i.e., the identity of the tracked user is known).

We conclude this dissertation by highlighting some open problems and items for future work:

Attributes Inference. First, we only tested our approach on profiles that provide music interests. Even with this limitation, our results show the effectiveness of our technique in inferring undisclosed attributes. In addition, we only based our method on user interests and did not combine it with any other available attributes (e.g., gender or relationship) to improve inference accuracy. One open question remains whether or not and to what extent considering other interests (e.g., movies, books, etc.) and/or combining with different available attributes can improve the inference accuracy.

Second, we demonstrated our approach using an English-version of Wikipedia. However, our approach is not restricted to English, since Wikipedia is also available in other languages. Finally, we encountered a few examples that denote a non-interest (or"dislike"). In particular, we observed interests that semantically express a dislike for a group, or an ideology. For instance, an interest can be created with the title "I hate Hollande". Our semantics-driven classification will falsely identify the users having this interest as "Holland" supporter. One might integrate Applied Linguistics Techniques to handle such peculiar cases and filter out dislikes.

Exploiting Public Information to enhance password cracking One of the main limitations of our approach is the data availability. In fact, we had only access to a very limited amount of information which might have an impact on the results (e.g., we had only three users publicly revealing their birthday). One vital extension is to collect more information and assess further the results we obtained. In near future, we are planning to deploy a social application that allows the user to measure his password strength by exploiting data collected from his public profile.

Privacy Measure. Although the proposed methodology provides a general framework to compute information surprisal when access to private attribute statistics is available (e.g., the Ads audience platform), our probability estimation depends on the OSN default privacy settings. However, we note that one of the desirable properties of our approach is that it only requires an unbiased sample of public profiles. Therefore, a massive privacy update (e.g., Facebook changing the default policy) would only require an up-to-date dataset to reflect these changes. In this work we did not investigate in depth the impact of attribute values on the uniqueness of attributes. We have observed the impact of some attribute values (e.g., US in *current country*) on the IS values computed for a single attribute. Such analysis can and should be extended to other attributes and attribute combinations. The goal of such extension would be to study the distinction between the impact of *rare combinations of attributes* and *the rarity of the attribute values*. Hence, this work represents only a first step towards the computation of an OSN "fingerprint" resulting from the combinations of attributes regardless of their values.

Information Leakage to OSN. We have shown that OSN operators are collecting a tremendous amount of data about their users. However, two questions remain unanswered:

(i) Is this data being used? (ii) If yes how?. We plan to further study these questions by collecting advertisements in OSNs and checking any discrepancies that might be the results of user tracking.

Part III

Appendices

Password creation policies

One surprising fact highlighted by the data in Chapter 4 (Section 4.5) is that usernames and passwords are very similar in a small, yet significant, fraction of the cases. Common sense mandates that a password should not be too similar to the corresponding username, because the username is almost always available to the attacker in a guessing attack.

This fact prompted us to study this specific aspect of password policies in more detail. We conducted a brief test[1] across 48 popular international sites (from the Alexa Top 500 list), to see how they handle similarities between the username and the password (See the table below).

These sites have different demands for security, ranging from relatively low security demands (Facebook, Twitter), to high security demands (Ebay, PayPal). We did not rely on the stated password policies, but manually created an account on each site. We created a random but plausible username that was not yet used with the service, and tried to register an account. We initially tried to use the username as the password, and if that failed we tried subsequent modifications until we succeeded. The results are worrisome, but not too surprising. Out of 48 sites tested, 27 allowed *identical* usernames and passwords, including major sites such as Google, Facebook, and Amazon, and only 4 sites required more than one character difference between the two. This could lead to highly effective guessing.

[1] This survey is neither representative nor complete, but the results are clear enough to show that the problem exists on a large scale.

Account	Username	Password	Same accepted?	Min. Diff. (# Chars)	Comments
Google	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
Facebook	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
Twitter	berkusrnfe02	berkusrnfe03	no	1	
Baidu	berkusrnfe02	berkusrnfe03	no	1	
Ebay	berkusrnfe03	BerkUsrnfe14	no	2	Username cannot be same as email; requires capitals.
Amazon	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
Paypal	berkusrnfe02@gmail.com	berkusrnfe03	no	1	
Yahoo	berkusrnfe02@yahoo.com	berkusrnfe03	no	1	
Wikipedia	berkusrnfe02	berkusrnfe03	no	1	
Windows Live	berkusrnfe02@hotmail.com	berkusrnfe03	no	1	
QQ.com	berkusrnfe02	berkusrnfe02	yes	0	
LinkedIn	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
Taobao	berkusrnfe02	berkusrnfe02	yes	0	
Sina.cn.com	berkusrnfe02@yahoo.com	berkusrnfe02	yes	0	
MSN	berkusrnfe02@gmail.com	berkusrnfe03	no	1	
WordPress	berkusrnfe02	berkusrnfe02	yes	0	
Yandex	berkusrnfe02	berkusrnfe03	no	1	
163.com	berkusrnfe02@163.com	berkusrnfe03	no	1	
Mail.ru	berkusrnfe02@Mail.ru	berkusrnfe03	no	1	
Weibo	berkusrnfe02@gmail.com	berkusrenfe02	no	0	
Tumblr	berkusrnfe02	berkusrnfe02	yes	0	
Apple	berkusrnfe02	BerkUsrnfe02	no	0	Password at least 1 capital, 1 number, no 3 consecutive identical characters, not same as account, at least 8 char.
IMDB	berkusrnfe02	berkusrnfe02	yes	0	
Craigslist	berkusrnfe02@gmail.com	berkusrnfe03	no	1	
Sohu	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
FC2	berkusrnfe02	berksurnfe03	no	3	The password cannot contain any five (5) consecutive characters of your e-mail address.
Tudou	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
Ask	berkusrnfe02	berkusrnfe02	yes	0	
iFeng	berkusrnfe02	berkusrnfe03	no	1	
Youku	berkusrnfe02	berkusrnfe02	yes	0	
Tmall	berkusrnfe02	berksurnfe03	no	3	
Imgur	berkusrnfe02	berkusrnfe02	yes	0	
Mediafire	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
CNN	berkusrnfe02	berkusrnfe02	yes	0	
Adobe	berkusrnfe02	berkusrnfe02	yes	0	
Conduit	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
odnoklassniki.ru/	berksrnfe02	berkusrnfe03	no	1	
AOL	berkusrnfe02	beruksrnef03	no	5	
The Pirate Bay	berkusrn	berkusrn	yes	0	Username length limit.
ESPN	berkusrnfe02	berkusrnfe02	yes	0	
Alibaba	berkusrnfe02	berkusrnfe02	yes	0	
Dailymotion	berkusrnfe02	berkusrnfe02	yes	0	
Chinaz	berkusrnfe02	berkusrnfe02	yes	0	
AVG	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	
Ameblo	berkusrnfe02	berkusrnfe03	no	1	
GoDaddy	berkusrnfe02	berkusrnfe02	yes	0	
StackOverflow	berkusrnfe02	BerkUsrnfe03	no	1	Needs capitals or special characters.
4shared	berkusrnfe02@gmail.com	berkusrnfe02	yes	0	

TABLE 8.1: Detailed results from a small survey on 48 large sites concerning their password policies.

Bibliography

Bibliography

[1] Lawrence Lessig. *Code 2.0*. CreateSpace, Paramount, CA, 2nd edition, 2009. ISBN 1441437649, 9781441437648.

[2] CCTV camera abuses. `http://www.notbored.org/camera-abuses.html`, 2014.

[3] Balachander Krishnamurthy and Craig Wills. Privacy diffusion on the web: a longitudinal perspective. In *Proceedings of the International Conference on World Wide Web*, 2009.

[4] Jonathan R Mayer and John C Mitchell. Third-party web tracking: Policy and technology. In *IEEE Symposium on Security and Privacy*, 2012.

[5] Claude Castelluccia and Arvind Narayanan. Privacy considerations of online behavioural tracking. *European Network and Information Security Agency (ENISA)*, 2012.

[6] Abdelberi Chaabane, Gergely Acs, and Mohamed Ali Kaafar. You Are What You Like! Information Leakage Through Users' Interests. In *Annual Network & Distributed System Security Symposium*, 2012.

[7] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You Are Who You Know: Inferring User Profiles in Online Social Networks. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, 2010.

[8] Elena Zheleva and Lise Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the International Conference on World Wide Web*, 2009.

[9] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the Annual Computer Security Applications Conference*, 2011.

[10] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the ACM Annual Computer Security Applications Conference*, 2010.

[11] Abdelberi Chaabane, Yuan Ding, Ratan Dey, Mohamed Ali Kaafar, and Keith W. Ross. A closer look at third-party osn applications: Are they leaking your personal information? *Passive and Active Measurement Conference*, 2014.

[12] Mario Frank, Ben Dong, Adrienne Porter Felt, and Dawn Song. Mining Permission Request Patterns from Android and Facebook Applications. In *IEEE International Conference on Data Mining*, 2012.

[13] Pern Hui Chia, Yusuke Yamamoto, and N Asokan. Is this app safe?: A large scale study on application permissions and risk signals. In *Proceedings of the international conference on World Wide Web*, 2012.

[14] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[15] Abdelberi Chaabane, Mohamed Ali Kaafar, and Roksana Boreli. Big friend is watching you: analyzing online social networks tracking capabilities. In *Proceedings of the 2012 ACM Workshop on Online Social Networks*, 2012.

[16] Balachander Krishnamurthy and Craig E Wills. On the leakage of personally identifiable information via online social networks. In *Proceedings of the ACM workshop on Online social networks*, 2009.

[17] Global Internet User Survey 2012 . `https://www.internetsociety.org/internet/global-internet-user-survey-2012`.

[18] Aleecia M. McDonald and Lorrie F. Cranor. The Cost of Reading Privacy Policies. *I/S: A Journal of Law and Policy for the Information Society*, 4:540–565, 2008. URL `http://www.is-journal.org/files/2012/02/Cranor_Formatted_Final.pdf`.

[19] Thorben Burghardt, Klemens Böhm, Erik Buchmann, Jürgen Kühling, and Anastasios Sivridis. A study on the lack of enforcement of data protection acts. In *e-Democracy*. Springer, 2009.

[20] Leucio Antonio Cutillo, Refik Molva, and Thorsten Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, 47(12):94–101, 2009.

[21] Sonja Buchegger, Doris Schiöberg, Le-Hung Vu, and Anwitaman Datta. Peerson: P2P social networking: early experiences and insights. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, 2009.

[22] A. Acquisti. Nudging privacy: The behavioral economics of personal information. *IEEE Security Privacy*, 7(6):82–85, 2009.

[23] Oshrat Ayalon and Eran Toch. Retrospective privacy: Managing longitudinal privacy in online social networks. In *Proceedings of the Symposium on Usable Privacy and Security*, 2013.

[24] Craig E. Wills and Mihajlo Zeljkovic. A personalized approach to web privacy: awareness, attitudes and actions. *Information Management & Computer Security*, 19, 2011.

[25] Aleecia M. Mcdonald. Individual privacy decisions in online contexts. 2010. URL `http://repository.cmu.edu/dissertations/7/`.

[26] Janice Y. Tsai, Serge Egelman, Lorrie Cranor, and Alessandro Acquisti. The effect of online privacy information on purchasing behavior: An experimental study. *Info. Sys. Research*, pages 254–268.

[27] Eran Toch, Yang Wang, and LorrieFaith Cranor. Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. *User Modeling and User-Adapted Interaction*, 2012.

[28] Janice Tsai, Patrick Kelley, Lorrie Cranor, and Norman Sadeh. Location-sharing technologies: Privacy risks and controls. TPRC, 2009.

[29] Eran Toch, Justin Cranshaw, Paul Hankes Drielsma, Janice Y Tsai, Patrick Gage Kelley, James Springfield, Lorrie Cranor, Jason Hong, and Norman Sadeh. Empirical models of privacy in location sharing. In *Proceedings of the ACM international conference on Ubiquitous computing*, 2010.

[30] Madhu Prabaker, Jinghai Rao, Ian Fette, Patrick Kelley, Lorrie Cranor, Jason Hong, and N Sadeh. Understanding and capturing people's privacy policies in a people finder application. In *Proceedings of the Workshop Ubicomp Privacy*, 2007.

[31] Norman Sadeh, Jason Hong, Lorrie Cranor, Ian Fette, Patrick Kelley, Madhu Prabaker, and Jinghai Rao. Understanding and capturing people's privacy policies in a mobile social networking application. *Personal and Ubiquitous Computing*, 13(6):401–412, 2009.

[32] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W Reeder. A nutrition label for privacy. In *Proceedings of the ACM Symposium on Usable Privacy and Security*, 2009.

[33] Yabing Liu, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Analyzing Facebook privacy settings: User expectations vs. reality. In *Proceedings of the ACM SIGCOMM Internet measurement conference*, 2011.

[34] Maritza Johnson, Serge Egelman, and Steven M Bellovin. Facebook and privacy: it's complicated. In *Proceedings of the Symposium on Usable Privacy and Security*, 2012.

[35] Richard H Thaler and Cass R Sunstein. Libertarian paternalism. *The American Economic Review*, 93(2):175–179, 2003.

[36] Cass R Sunstein and Richard H Thaler. Libertarian paternalism is not an oxymoron. *The University of Chicago Law Review*, pages 1159–1202, 2003.

[37] Claude Castelluccia, Abdelberi Chaabane, Markus Dürmuth, and Daniele Perito. When privacy meets security: Leveraging personal information for password cracking. *arXiv preprint arXiv:1304.6584*, 2013.

[38] Arvind Narayanan and Vitaly Shmatikov. Fast dictionary attacks on passwords using time-space tradeoff. In *Proceedings of the ACM conference on Computer and communications security*, 2005.

[39] Terence Chen, Abdelberi Chaabane, Pierre Ugo Tournoux, Mohamed-Ali Kaafar, and Roksana Boreli. How Much Is Too Much? Leveraging Ads Audience Estimation to Evaluate Public Profile Uniqueness. In *Privacy Enhancing Technologies*. 2013.

[40] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM SIGKDD Explorations Newsletter*, 10(2):12–22, 2008.

[41] Elena Zheleva and Lise Getoor. Privacy in social networks: A survey. In *Social Network Data Analytics*, pages 277–306. Springer US, 2011.

[42] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the International Conference on Information and Knowledge Management*, 2003.

[43] Michael Beye, Arjan Jeckmans, Zekeriya Erkin, Pieter Hartel, Reginald Lagendijk, and Qiang Tang. Literature overview-privacy in online social networks. 2010.

[44] George Pallis, Demetrios Zeinalipour-Yazti, and MariosD. Dikaiakos. Online social networks: Status and trends. In *New Directions in Web Data Management 1*, volume 331. 2011.

[45] Chi Zhang, Jinyuan Sun, Xiaoyan Zhu, and Yuguang Fang. Privacy and security for online social networks: challenges and opportunities. *Network, IEEE*, 24, 2010.

[46] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, 2009.

[47] Peter S Bearman, James Moody, and Katherine Stovel. Chains of affection: The structure of adolescent romantic and sexual networks1. *American Journal of Sociology*, 110(1):44–91, 2004.

[48] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the International Conference on World Wide Web*, 2007.

[49] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, 2008.

[50] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham D Flaxman. Sybilguard: Defending against sybil attacks via social networks. *IEEE/ACM Transactions on Networking*, 16(3):576–589, 2008.

[51] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[52] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the ACM SIGMOD international conference on Management of data*, 2008.

[53] Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. Anonymizing social networks. *Computer Science Department Faculty Publication Series*, page 180, 2007.

[54] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *IEEE International Conference on Data Engineering*, 2008.

[55] Lei Zou, Lei Chen, and M Tamer Özsu. K-automorphism: A general framework for privacy preserving network publication. *Proceedings of the VLDB Endowment*, 2 (1):946–957, 2009.

[56] Bin Zhou and Jian Pei. The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowledge and Information Systems*, 28(1):47–77, 2011.

[57] Xiaowei Ying and Xintao Wu. Randomizing social networks: a spectrum preserving approach. In *SDM*, volume 8, pages 2008–739, 2008.

[58] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. Anonymizing bipartite graph data using safe groupings. *Proceedings of the VLDB Endowment*, 2008.

[59] Smriti Bhagat, Graham Cormode, Balachander Krishnamurthy, and Divesh Srivastava. Class-based graph anonymization for social network data. *Proceedings of the VLDB Endowment*, 2(1):766–777, 2009.

[60] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, 2010.

[61] Arvind Narayanan, Elaine Shi, and Benjamin IP Rubinstein. Link prediction by de-anonymization: How we won the kaggle social network challenge. In *The 2011 International Joint Conference on Neural Networks*, 2011.

[62] László Babai. Moderately exponential bound for graph isomorphism. In *Fundamentals of Computation Theory*, pages 34–50. Springer, 1981.

[63] Cynthia Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.

[64] Vibhor Rastogi, Michael Hay, Gerome Miklau, and Dan Suciu. Relationship privacy: Output perturbation for queries with joins. In *Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2009.

[65] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *Proceedings of the IEEE International Conference on Data Mining*, 2009.

[66] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y. Zhao. Sharing graphs using differentially private graph models. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, 2011.

[67] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2007.

[68] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. In *VLDB*, 2011.

[69] Davide Proserpio, Sharon Goldberg, and Frank McSherry. A workflow for differentially-private graph synthesis. In *Proceedings of the ACM Workshop on Online Social Networks*, 2012.

[70] Lise Getoor. Link mining: A new data mining challenge. *SIGKDD Explorations*, pages 1–6, 2003.

[71] Joshua O'Madadhain, Jon Hutchins, and Padhraic Smyth. Prediction and ranking algorithms for event-based network data. *SIGKDD Explor. Newsl.*, 7(2):23–30, 2005.

[72] Matthew J. Rattigan and David Jensen. The case for anomalous link discovery. *SIGKDD Explor. Newsl.*, 7(2):41–47, December 2005. ISSN 1931-0145.

[73] Rin Popescul and Lyle H. Ungar. Statistical relational learning for link prediction. In *Proceedings of the Workshop on Learning Statistical Models from Relational Data at IJCAI-2003*, 2003.

[74] Yonghong Tian, Qiang Yang, Tiejun Huang, Charles X. Ling, and Wen Gao. Learning contextual dependency network models for link-based classification. *IEEE Transactions on Knowledge and Data Engineering*, 2006.

[75] Zhijun Yin, Manish Gupta, Tim Weninger, and Jiawei Han. LINKREC: A Unified Framework for Link Recommendation with User Attributes and Graph Structure. In *Proceedings of the International Conference on World Wide Web*, 2010.

[76] Mathias Humbert, Théophile Studer, Matthias Grossglauser, and Jean-Pierre Hubaux. In *European Symposium on Research in Computer Security*, 2013.

[77] Elena Zheleva and Lise Getoor. Preserving the privacy of sensitive relationships in graph data. In *Proceedings of the ACM SIGKDD International Conference on Privacy, Security, and Trust*, 2008.

[78] Lada Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2001.

[79] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Structure and evolution of blogspace. *Communications of the ACM*, 47(12):35–39, 2004.

[80] Myunghwan Kim and Jure Leskovec. Modeling social networks with node attributes using the multiplicative attribute graph model. *CoRR*, 2011.

[81] Jianming He, Wesley W. Chu, and Zhenyu (victor Liu. Inferring privacy information from social networks. In *IEEE International Conference on Intelligence and Security Informatics*, 2006.

[82] Jack Lindamood and Murat Kantarcioglu. Inferring Private Information Using Social Network Data. Technical report, University of Texas at Dallas, 2008. URL `http://www.utdallas.edu/~mxk055100/publications/techreport-sn-privacy.pdf`.

[83] Justin Becker and Hao Chen. In *Proceedings of the Web 2.0 Security and Privacy*, 2009.

[84] Delip Rao, Michael J Paul, Clayton Fink, David Yarowsky, Timothy Oates, and Glen Coppersmith. Hierarchical bayesian models for latent attribute detection in social media. In *ICWSM*, 2011.

[85] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. Classifying latent user attributes in twitter. In *Proceedings of the international workshop on Search and mining user-generated contents*, 2010.

[86] John D. Burger, John Henderson, George Kim, and Guido Zarrella. Discriminating gender on twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011.

[87] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, republicans and starbucks afficionados: User classification in twitter. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.

[88] R. Dey, Cong Tang, K. Ross, and N. Saxena. Estimating age privacy leakage in online social networks. In *Proceedings of INFOCOM*, 2012.

[89] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social Structure of Facebook Networks. 2011. URL `http://arxiv.org/abs/1102.2166`.

[90] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[91] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 2002.

[92] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, 2004.

[93] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*, volume 4, pages 412–418, 2004.

[94] Mike Thelwall. Heart and soul: Sentiment strength detection in the social web with sentistrength1. URL `http://sentistrength.wlv.ac.uk`.

[95] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, page 71, 2001.

[96] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.

[97] Pollyanna Gonçalves, Fabrício Benevenuto, and Meeyoung Cha. Panas-t: A pychometric scale for measuring sentiments on twitter. *arXiv preprint arXiv:1308.1857*, 2013.

[98] Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceeding of the International AAAI Conference on Weblogs and Social Media*, 2011.

[99] Pollyanna Gonçalves, Matheus Araújo, Fabrício Benevenuto, and Meeyoung Cha. Comparing and combining sentiment analysis methods. In *Proceedings of the ACM Conference on Online Social Networks*, 2013.

[100] Aniko Hannak, Eric Anderson, Lisa Feldman Barrett, Sune Lehmann, Alan Mislove, and Mirek Riedewald. Tweetin'in the rain: Exploring societal-scale effects of weather on mood. In *Proceeding of the International AAAI Conference on Weblogs and Social Media*, 2012.

[101] Ratan Dey, Zubin Jelveh, and Keith W. Ross. Facebook users have become much more private: A large-scale study. In *PerCom Workshops*, 2012.

[102] Amin Tootoonchian, Geoffrey Salmon, and Ahmad Ziad Hatahet. Fine grained access control in online social networks. *Technical report, University of Toronto*, 2007.

[103] Amin Tootoonchian, Kiran Kumar Gollu, Stefan Saroiu, Yashar Ganjali, and Alec Wolman. Lockr: Social access control for web 2.0. In *Proceedings of the Workshop on Online Social Networks*, 2008.

[104] Amin Tootoonchian, Stefan Saroiu, Yashar Ganjali, and Alec Wolman. Lockr: Better privacy for social networks. In *Proceedings of the International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, 2009.

[105] Wanying Luo, Qi Xie, and U. Hengartner. Facecloak: An architecture for user privacy on social networking sites. In *International Conference on Computational Science and Engineering*, 2009.

[106] Matthew M. Lucas and Nikita Borisov. Flybynight: Mitigating the privacy risks of social networking. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, 2008.

[107] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology—EUROCRYPT*. 1998.

[108] Saikat Guha, Kevin Tang, and Paul Francis. Noyb: Privacy in online social networks. In *Proceedings of the Workshop on Online Social Networks*, 2008.

[109] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2006.

[110] Randy Baden, Adam Bender, Neil Spring, Bobby Bhattacharjee, and Daniel Starin. Persona: An online social network with user-defined privacy. In *Proceedings of the ACM SIGCOMM Conference on Data Communication*, 2009.

[111] Filipe Beato, Markulf Kohlweiss, and Karel Wouters. Scramble! your social network data. In *Proceedings of the International Conference on Privacy Enhancing Technologies*, 2011.

[112] Ning Xia, Han Hee Song, Yong Liao, Marios Iliofotou, Antonio Nucci, Zhi-Li Zhang, and Aleksandar Kuzmanovic. Mosaic: quantifying privacy leakage in mobile networks. In *Proceedings of the ACM SIGCOMM conference*, 2013.

[113] M. Egele, A. Moser, C. Kruegel, and E. Kirda. Pox: Protecting users from malicious facebook applications. *Computer Communications*, 35(12):1507–1515, 2012.

[114] M. Shehab, A. Squicciarini, and G. Ahn. Beyond user-to-user access control for online social networks. In *Information and Communications Security*. 2008.

[115] Adrienne Felt and David Evans. Privacy protection for social networking apis. *Web 2.0 Security and Privacy*, 2008.

[116] Facebook Sets Historic IPO. `http://online.wsj.com/article/SB10001424052970204879004577110780078310366.html`.

[117] Arnold Roosendaal. Facebook Tracks and Traces Everyone: Like This! *SSRN eLibrary*, 2010.

[118] Ting-Fang Yen, Yinglian Xie, Fang Yu, Roger P. Yu, and Martin Abadi. Host fingerprinting and tracking on the web: Privacy and security implications. In *Proceedings of the Annual Network & Distributed System Security Symposium*, 2012.

[119] Alerts users about the web bugs, ad networks and widgets on visited web pages. `http://www.ghostery.com`.

[120] Latanya Sweeney. Uniqueness of Simple Demographics in the U.S. Population. *LIDAP-WP4 Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA*, 2000.

[121] Philippe Golle. Revisiting the Uniqueness of Simple Demographics in the US Population. In *Proceedings of the ACM Workshop on Privacy in Electronic Society*, 2006.

[122] Peter Eckersley. How Unique is Your Web Browser? In *Proceedings of the International Conference on Privacy Enhancing Technologies*, 2010.

[123] Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, and Pere Manils. How Unique and Traceable Are Usernames? In *Proceedings of the International Conference on Privacy Enhancing Technologies*, 2011.

[124] Balachander Krishnamurthy and Craig E. Wills. Privacy Leakage in Mobile Online Social Networks. In *Proceedings of the Workshop on Online social networks*, 2010.

[125] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[126] Thomas Hofmann. Probabilistic Latent Semantic Analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, 1999.

[127] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[128] Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y. Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proceedings of the International conference on World wide web*, 2009.

[129] E. Zheleva, J. Guiver, E. M. Rodrigues, and N. Milic-Frayling. Statistical models of music-listening sessions in social media. In *Proceedings of the International conference on World Wide Web*, 2010.

[130] Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May 1998. URL `http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=8106`.

[131] OpenCyc. `http://www.opencyc.org/`, 2006.

[132] David Milne. An open-source toolkit for mining wikipedia. In *Proceeding of New Zealand Computer Science Research Student Conf*, 2009.

[133] Zhiyuan Liu, Yuzhou Zhang, Edward Y. Chang, and Maosong Sun. Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing. *ACM Transactions on Intelligent Systems and Technology, special issue on Large Scale Machine Learning*, 2011. Software available at `http://code.google.com/p/plda`.

[134] J. Puzicha, T. Hofmann, and J.M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[135] Maciej Kurant, Minas Gjoka, Carter T. Butts, and Athina Markopoulou. Walking on a Graph with a Magnifying Glass. In *Proceedings of ACM SIGMETRICS*, 2011.

[136] Qt port of webkit: an open source web browser engine. `http://trac.webkit.org/wiki/QtWebKit`.

[137] Robots Exclusion Protocol. `http://www.robotstxt.org/`, 1996.

[138] Facebook Statistics. `http://gold.insidenetwork.com/facebook/facebook-stats/`, 2011.

[139] T. K. Landauer and S. T. Dumais. Solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 1997.

[140] Thomas L. Griffiths, Joshua B. Tenenbaum, and Mark Steyvers. Topics in semantic representation. *Psychological Review*, 114:2007, 2007.

[141] Andrew I. Schein, Alexandrin Popescul, Lyle H., Rin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the Annual International ACM Conference on Research and Development in Information Retrieval*, 2002.

[142] Paolo Cremonesi and Roberto Turrin. Analysis of cold-start recommendations in IPTV systems. In *Proceedings of the ACM conference on Recommender systems*, New York, NY, USA, 2009.

[143] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proceedings of the International Conference on Machine Learning*, 1998.

[144] S. Lai, L. Xiang, R. Diao, Y. Liu, H. Gu, L. Xu, H. Li, D. Wang, K. Liu, J. Zhao, and C. Pan. Hybrid recommendation models for binary user preference prediction problem. In *KDD Cup*, 2011.

[145] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the ACM International conference on World Wide Web*, 2012.

[146] Tom N Jagatic, Nathaniel A Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[147] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proceedings of the Symposium on Security and Privacy*, 2012.

[148] Gershon Kedem and Yuriko Ishihara. Brute force attack on unix passwords with simd computer. In *Proceedings of the Conference on USENIX Security Symposium*, 1999.

[149] HashCat. OCL HashCat-Plus, 2012. `http://hashcat.net/oclhashcat-plus/`.

[150] Matt Weir, Sudhir Aggarwal, Breno de Medeiros, and Bill Glodek. Password cracking using probabilistic context-free grammars. In *IEEE Symposium on Security and Privacy*, 2009.

[151] Niels Provos and David Mazières. A future-adaptive password scheme. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 32–32, 1999.

[152] OpenWall. John the Ripper, 2012. `http://www.openwall.com/john`.

[153] Matteo Dell'Amico, Michiardi Pietro, and Yves Roudier. Password strength: An empirical analysis. In *Proceedings of IEEE Conference on Computer Communications*, 2010.

[154] Claude Castelluccia, Markus Dürmuth, and Daniele Perito. Adaptive password-strength meters from markov models. In *Proceedings of the Network and Distributed Systems Security Symposium*. The Internet Society, 2012.

[155] R. Morris and K. Thompson. Password security: a case history. *Communications. ACM*, 22(11):594 – 597, 1979.

[156] E. H. Spafford. Observing reusable password choices. In *Proceedings of the USENIX Security Symposium*, 1992.

[157] D. V. Klein. Foiling the cracker: A survey of, and improvements to, password security. In *Proceedings of the USENIX UNIX Security Workshop*, 1990.

[158] M. Bishop and D. V. Klein. Improving system security via proactive password checking. *Computers & Security*, 14(3):233–249, 1995.

[159] The password meter. Online at http://www.passwordmeter.com/.

[160] William E. Burr, Donna F. Dodson, and W. Timothy Polk. Electronic authentication guideline: NIST special publication 800-63, 2006.

[161] Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *Proceedings of the ACM conference on Computer and communications security*, 2010.

[162] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. Of passwords and people: Measuring the effect of password-composition policies. In *Conference on Human Factors in Computing Systems*, 2011.

[163] Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Tim Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2012.

[164] Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2012.

[165] Stuart Schechter, Cormac Herley, and Michael Mitzenmacher. Popularity is everything: a new approach to protecting passwords from statistical-guessing attacks. In *Proceedings of the USENIX conference on Hot Topics in Security*, 2010.

[166] Serge Egelman, Joseph Bonneau, Sonia Chiasson, David Dittrich, and Stuart Schechter. It's not stealing if you need it: A panel on the ethics of performing research using public data of illicit origin. In *Financial Cryptography and Data Security*, 2012.

[167] PCFG. Matt Weir, 2012. https://sites.google.com/site/reusablesec/Home/password-cracking-tools/probablistic_cracker.

[168] Word list Collection, 2012. http://www.outpost9.com/files/WordLists.html.

[169] Alan S. Brown, Elisabeth Bracken, Sandy Zoccoli, and King Douglas. Generating and remembering passwords. *Applied Cognitive Psychology*, 2004.

[170] Arvind Narayanan and Vitaly Shmatikov. Robust De-anonymization of Large Sparse Datasets. *IEEE Symposium on Security and Privacy*, 2008.

[171] How microsoft and yahoo are selling politicians access to you. https://www.propublica.org/article/how-microsoft-and-yahoo-are-selling-politicians-access-to-you.

[172] Facebook's Name Policy. http://www.facebook.com/help/?page=258984010787183.

[173] Google geocoding api. https://developers.google.com/maps/documentation/geocoding/.

[174] Facebook ads - optimization. http://ads.ak.facebook.com/ads/FacebookAds/ad_optimization_final.pdf.

[175] William A. Gale and Geoffrey Sampson. Good-turning Frequency Estimation without Tears. *Journal of Quantitative Linguistics*, 2:217–237, 1995.

[176] Adriano Azevedo-filho. Laplace's Method Approximations for Probabilistic Inference in Belief Networks with Continuous Variables. In *De Mantaras*, pages 28–36, 1994.

[177] TrustedSource - Customer URL Ticketing System. www.trustedsource.org/en/feedback/url.

[178] Method and system for web user profiling and selective content delivery. http://www.google.com/patents/US8108245.

[179] Sung Min Bae, Sung Ho Ha, and Sang Chan Park. Fuzzy web ad selector based on web usage mining. *IEEE Intelligent Systems*, 18:62–69, 2003. ISSN 1541-1672. doi: http://doi.ieeecomputersociety.org/10.1109/MIS.2003.1249171.

[180] R. Dey, Z. Jelveh, and K. Ross. Facebook users have become much more private: A large-scale study. In *PERCOM Workshops*, 2012.

[181] Georgios Kontaxis, Iasonas Polakis, Sotiris Ioannidis, and Evangelos P Markatos. Detecting social network profile cloning. In *IEEE International Conference on Pervasive Computing and Communications*, 2011.