Incognitus: Privacy-Preserving User Interests in Online Social Networks

Alexandros Kornilakis, Panagiotis Papadopoulos, Evangelos Markatos

FORTH-ICS, Greece
{kornilak, panpap, markatos}@ics.forth.gr

Abstract. Online Social Networks have changed the way we reach news and information. An increasing number of people use social networks not only for communicating with friends and colleagues but also for their daily information needs. Apart from providing the users with personalized information in a timely manner, this functionality may also raise significant privacy concerns. The service provider is able to observe both the Pages a user is subscribed to and her interactions with them. The collected data can form a detailed user profile, which can later be used for several purposes; usually beyond the control of the user. To address these privacy concerns, we propose Incognitus: an approach to allow users browse Pages of OSNs without disclosing their interests or activity to the service provider. Our approach provides (i) a incognito mode of operation when browsing privacy-sensitive content. In this isolated, offline mode no tracking mechanisms can monitor the users behavior and no information can be leaked to the provider. At the same time, (ii) by using an obfuscation-based mechanism, Incognitus reduces the accuracy of the service provider when monitoring the interests of a user. Early results show that Incognitus has minimal bandwidth requirements and imposes reasonable latency to the users browsing experience.

1 Introduction

Online Social Networks such as Facebook, Twitter, Tumblr, Google+, Weibo, etc. do not constitute any more, a platform solely used by the users for communication and social interaction. The so-called model of ”social broadcast” enables users, instead of searching and consuming news through traditional media or news websites, to have more personalized news delivered directly to them in a timely manner. According to a recent study [30], an increasing 67% of adults in the US get their news from social media; it was also found that social media now outperforms television as the major information source [34]. By providing a handy user interface OSNs enables users, through a publish-subscribe model, to subscribe/follow information providers who maintain *channels (or Pages)* and receive updates regarding the content they publish.

Of course, all this handy and timely superabundance of information does not come for free. In exchange for their monetarily free services OSN providers deliver targeted advertisements to their users. To provide better-matched advertisements and suggestions a wide spectrum of user social interconnections [25], activity and interactions [22] has to be monitored. The OSN by monitoring the user’s interactions with the content of such Pages - *links clicked, photos viewed, videos watched* - it is able to create a very detailed profile for every user containing information related to interests and preferences reconstructing thus parts of her actual personality [36]. While recently there is an increased awareness about privacy on social networks and a desire for data protection regularization [1], there are incidents [32] indicating that the user might have to take additional steps to protect her privacy.

In this paper, we are interested in protecting content considered as privacy-sensitive. For example, content related to politics, sexual orientation, religion or health issues. If a user follows a Page of a particular politician, the service may infer the user's political beliefs. We think that it is of the user's the main interest to protect such sensitive information. Thus the objective of this paper is to help users that although interested in receiving updates about privacy-sensitive issues, may not be willing to disclose their personal preferences and interests. Even if these users avoid subscribing to particular privacy-sensitive Pages thus choosing to manually fetch them every now and then, the service provider still can identify not only the fact that they requested for these Pages but also their interactions with the content.

The alternatives such a user has is to hide her actual identity: by creating a second disposable account to subscribe to the privacy-sensitive Pages she is interested in. Unfortunately, this approach would be subject to contamination since information from different web browsing sessions, such as browser or device fingerprints [2, 7, 24] or persistent and synced cookies [16, 26] are able to correlate anonymous and eponymous browsing sessions revealing thus the true identity of the user [15, 27]. To make matters worse, in popular online social networks disposable accounts is not an option [6] and having multiple fake accounts is not easy nowadays (some OSN's require a mobile phone number in order to register). To remedy this problem one might use a second account and a VM per browsing session over an anonymization network (such as Tor [12]) to hide the IP address aiming to finally limit cross-contamination. Of course the applicability of such approach (i.e. a combination of VM and Tor) in tablets and smartphones alongside the inconvenience it may cause to ordinary users, may repel most users. The user can always stop using OSNs to protect her privacy; however, there are a lot of organizations, associations, groups or communities that leverage the ease of message broadcasting and audience gathering in social networking platforms and do not maintain websites of their own outside the OSN. In this work, we propose a method to allow OSN users to protect their interests when anonymity is not feasible.

There are two distinct cases able to disclose a user's interests: (i) the actual fact that the user follows, likes or subscribes to a specific Page (ii) her interactions with this Page (navigating through past posts, watching videos or photos, etc.). In this paper, we propose *Incognitus*, a system to preserve the privacy of the users' interests against both cases. Our approach is twofold: we provide an obfuscation mechanism and an incognito mode of operation. In our system, the user follows/subscribes to an additional number of Pages which are used as noise. These additional Pages will reduce the accuracy of the service provider while trying to distinguish the noisy Pages from the real ones. After concealing the actual subscription our approach provides the user with a protected offline mode in which she can switch to whenever she wants access to a Page she considers as privacy-sensitive. In this isolated, offline mode of operation no requests or tracking mechanisms of the service provider can leak information about her activity and behavior.

To summarize, in this paper, we make the following contributions:

1. We propose a methodology to preserve the privacy of the sensitive interests of a user while browsing an OSN. Our approach provides an isolated protected mode of operation where no user monitoring can be applied by the curious service provider.
2. To assess the feasibility and effectiveness of our approach, we implemented our system as an extension for the Firefox browser using Facebook as our case study.
3. We experimentally evaluate our prototype and we show that it has minimal bandwidth requirements and adds reasonable latency to the user's browsing experience.

2 Motivation and Threat Model

In this paper, we assume the existence of an online social networking service, where there are several Pages maintained by organizations, authorities, corporations, groups or individuals (e.g. journalists, politicians, activists, doctors, etc.) to publish content and inform their audience about a particular subject. Such published content may include photos, articles, posts, links to external web pages, videos, etc. Users subscribe themselves to the Page in order to access this content and have updates delivered to their newsfeed in a timely manner.

Furthermore, we assume an honest online social networking service provider, which may try to identify the user's preferences in order to display personalized recommendations and advertisements matching the interests of the user. This service provider is capable of passively recording the user's activity at any time by observing her online interactions with a Page's content. This way, this provider knows: *When the user watches what video and for how long? When she browses what photo? On which photos she spends more time? What articles she reads?, What posts she reads and if she unfolds its related comments.* We assume that the service provider will not try to "cheat" by actively interfering with the process users are employing to protect their privacy, or try to gain more information than what a user is willing, or required, to give. We think that this assumption is valid since social networking service providers do not want to be hostile against their own users thus jeopardizing their reputation.

All this collected information about users' interests, is a property of the OSN and can be considered as an asset in case of future acquisition by another company [19,31]. In addition, this information could be later sold to advertisers [4,29] and used beyond the control of the user. We consider this tracking capability as a potential concern for the users' privacy and therefore in this paper, we aim to preserve the privacy of users who use OSNs to fulfill their daily information needs, users who use OSNs to get timely personalized news and not for communicating purposes. As a consequence, in this paper, we assume users that act as consumers of information, without being interested in posting, sharing or reacting [18] at any information related to the sensitive Page they are interested in. Such actions could allow a service provider by deploying term extraction and ML algorithms to identify the actual interests of the user.

3 Design Overview

The motivation behind our work is to build an offline incognito mode, to which the user can switch every time she feels like browsing a privacy-sensitive Facebook Page. In this isolated mode, no trackers would be able to monitor her behavior and interactions. In order for the user to browse this content, her browser needs to fetch the entire Page leaking this way to the service provider the fact that she is interested in navigating to this Page. As a consequence, the service provider can associate her interests with the related subject of the Page. To remedy this privacy infringement and reduce the accuracy of such association, we use an obfuscation-based approach, where we introduce noise, by encouraging the user to retrieve k more Pages used as noise.

3.1 Concealing the user's subscriptions

Assume a user interested in a Page P, which deals with a privacy-sensitive issue: for example, the electoral campaign of a candidate. As soon as the user subscribes herself to this Page, the social network can consider with high probability that she is interested in the associated political party and hence infer her political preferences. In *Incognitus*, to reduce this certainty of the service provider when predicting the interests of its users, we encourage the user to subscribe, along with the Page P, to an additional number of $k-1$ Pages, which act as noise. By obfuscating this way her choices, the service provider will get subscribe requests for k Pages without being able to accurately identify the Page she is actually interested in.

Parameter k practically defines the amount of noise introduced in the user's profile. By fine tuning this obfuscation number, users are able to achieve the level of privacy they are comfortable with. Very small values of k may increase the confidence of a service provider and disclosure the actual interests of the user. On the other hand, very high values of k may skyrocket the overhead and the bytes downloaded in her browser, since *Incognitus* will download more noise Pages and content.

Selection of noise

All such noise Pages are randomly chosen from a publicly accessible list SP of "privacy-sensitive" Pages, which is shared among all users. In *Incognitus*, these $k-1$ noise Pages are chosen randomly with uniform probability from SP. As a consequence, when a user subscribes to k Pages (i.e. P+noise) the probability for the service provider to identify the one the user is actually interested in, is $1/n$. Note at this point that apart from the P, the $k-1$ noise Pages are not fake and also privacy-sensitive. It is apparent, that users enable *Incognitus* to specifically conceal Pages that consider sensitive, the use of non sensitive Pages as noise would help the service provider easily filter out those Pages. By using the same SP list for all users, *Incognitus* allows for every Page a user may use as noise, to appear some other users that are indeed interested in it and hence subscribed to it.

Of course, not all Pages enjoy the same popularity. In case of a user interested in a very popular Page (e.g. 30% of the total users are already subscribed to it), a service provider may pinpoint the very popular Page among the $k-1$ others and assume that most probably this Page is the one the user is interested in. To mitigate such cases, instead of uniform selection of noise Pages, one can use a proportional selection, already

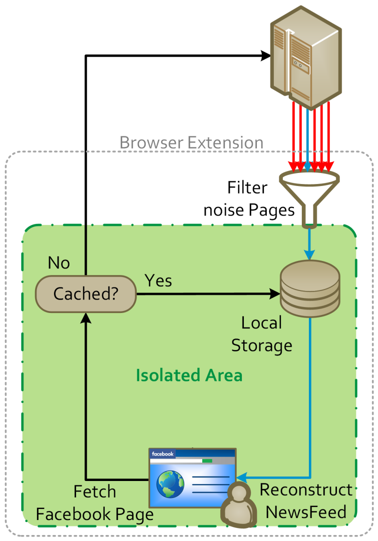


Fig. 1: High level overview of Incognitus. Incognitus provides the same browsing experience, while the user is able to browse the privacy-sensitive content incognito.

proposed in related studies [28], based on the size of the Pages' audience. This would lead to a selection of noise Pages with similar popularity to the Page the user is actually interested in.

3.2 Concealing the user's interactions

Apart from the subscriptions of a user to Pages, her interests can be also leaked by monitoring her behavior, while browsing a Page. Specifically, even after obfuscating her subscriptions, the service provider may track her interactions (e.g. clicks, mouse hovers etc.) and distinguish which Pages she fetches to use as noise and which Pages she fetches because she is actually interested in.

To remedy this leak, *Incognitus* stores the entire content of a privacy-sensitive Page locally (e.g. posts, videos, photos, etc. along with their metadata) and whenever the user browses this Page, it intercepts the HTTP requests to the web server and instead retrieves the content from the disk. As a consequence, an isolated offline mode is created allowing the user to interact with the content (e.g. view a photo) without allowing the service provider to learn about it. To achieve the same user experience, *Incognitus* reconstructs the user's NewsFeed by transcoding on-the-fly incoming components. This way, in case of privacy-sensitive content (e.g. posts) originated from privacy-sensitive Pages the links to the associated components will point to the local stored files.

Of course, whenever *Incognitus* downloads the content of a Page P to store it locally, and in order for the content request of Page P to remain obfuscated, it also downloads the content of the rest $k-1$ Pages as well (although it does not need to store it). This way, the service provider will see a bunch of identical HTTP requests for content, being unable to identify the noise requests. Such *update* operation of *Incognitus* is performed asynchronously in the background, so it cannot degrade the user's browsing experience by imposing additional latency.

The delta approach

Its easy to anticipate that downloading bulk content from multiple pages, every time a user browses Page *P*, may put an unbearable load to the browsing experience, causing a significant increase not only to the rendering latency but also to the required bandwidth. To avoid such overhead, *Incognitus*, on the background and periodically, downloads any possible recent updates of Page *P* and stores them locally. This way, whenever a user needs to fetch *P*, the browser extension will use the pre-fetched content from the disk and thus render the content in zero time. In addition, by using the delta approach *Incognitus* downloads less bytes from the network, avoiding re-fetching content every time the user browses *P*. Note again that like above *Incognitus* does not download only the deltas of Page *P* but the deltas of all *k* privacy-sensitive Pages the user has subscribed to.

In summary, we can see in figure 1 a high level overview of the internal design of *Incognitus*. It periodically (i) retrieves the deltas from all *k* Pages, (ii) filters them to discard the noise (i.e. posts from noise Pages) providing the exact same browsing experience as before and (iii) stores locally the content of *P* (i.e. html, images, photos, videos, json files etc.). When the user opens her NewsFeed, the served html gets transcoded and the links of the components, originated from *P*, are rewritten to point to the local stored content.

4 Sensitive Pages

In this section, we describe how the list *SP* of sensitive Pages is formed in *Incognitus*. We first present the web-based indexing service, responsible for distributing the list *SP*. Then we describe how this list gets its elements updated and how the users, by sending anonymous requests, can add Pages they consider sensitive to the next version of this list. Finally, we discuss how the Page indexing service may control the size of *SP* to keep it smaller than a desired threshold and what *Incognitus* does in case of disappearing sensitive Pages.

5 A web-based indexing service

To be effective, our system needs to make sure that all of its users share the exact same list *SP* of sensitive Pages. The rationale behind it is that users' random selections of "noise" Pages from *SP*, contribute to hide the Pages other users are interested in, which belong to *SP*, too. Therefore, upon installation, but also periodically, *Incognitus* downloads the final version of this list from a publicly accessible web index.

We envision that this web index along with the list *SP* and the project in general would be maintained by the broader community of users. Such privacy-concerned communities already support similar volunteer-based projects, such as Tor [5]. The list *SP* can be seeded by an initial set of sensitive Pages and further improved through human intervention and participation of the community.

5.1 Updating the list of sensitive Pages

Initially, *SP* can contain a set of commonly used sensitive Pages from several different categories. Nevertheless, new sensitive Pages may arise, or users may want to subscribe to a sensitive Page that is not included in *SP*. Thus, there is a need to update the list of sensitive Pages dynamically based on the users needs. To accommodate this necessity, we propose a publicly accessible web-based indexing service that hosts the latest version of the list *SP* with the sensitive Pages. *Incognitus*, which runs transparently at the client-side as a browser extension, periodically communicates with this service to receive the updated list of sensitive Pages. This way, each user is equipped with the latest version of *SP*, which includes any newly added sensitive Page.

Apart from that, the indexing service must be able to accept user requests for new Pages to be included in *SP*. These requests must be performed, through *Incognitus* transparently and anonymously. To achieve this, whenever a user wants to subscribe herself to a sensitive Page, *Incognitus* intercepts the “subscribe” request and checks whether this Page already exists in *SP*. If so, *Incognitus* subscribes the user to the noise selection described in Section 3. If the Page does not exist in *SP*, the user can ask *Incognitus* to consider this Page as sensitive from now on and thus include it in the next version of *SP*. After that, *Incognitus* communicates anonymously with the indexing service counterpart, sending a request with the new Page to be added.

It is important that this communication, between the user and the indexing service to be anonymous without revealing either the identity or the IP address of the user. Hence, the user would need to trust no indexing service, and consequently the indexing service would not ever learn which user asked for what Page to be added. To achieve such anonymous communication, *Incognitus* uses a network anonymizer (i.e. Tor [5]) when sending requests for the sensitive Page index.

In addition, to avoid any timing correlations, *Incognitus*, upon sending a request to include a new Page in *SP*, waits for a reasonable period of time before actually subscribing the user to this Page (and noise Pages). This way, it ensures that there will be a number of other users that will have downloaded the updated version of *SP*, which includes the newly added Page.

5.2 Controlling the size of sensitive Page list

Eventually, the new additions of sensitive Pages will end up increasing significantly the size of the list. As a result, each sensitive Page in *SP* will have less probability to get selected as noise in case of uniform noise selection. This, as a consequence, would decrease the efficiency of *Incognitus* since for a Page *P* a user is interested in, there must be a number of other users using it as noise, in order for the service provider not to be able to distinguish who uses it as noise and who does not. To address this issue, the indexing service is responsible to keep *SP* in the desirable size. Specifically, when *SP* exceeds a predefined size threshold, it automatically removes from *SP* the less important Pages, i.e., less popular Pages, less active Pages, or the older ones based on the date they were inserted in *SP*. This way, the most important Pages, which are more popular, active, or were recently added to *SP*, will remain in the list of sensitive Pages. Note that if a user wants to subscribe to a Page that has been previously removed from *SP*, she can re-insert it any time through the procedure we describe above.

5.3 Disappearing Pages

While using *Incognitus*, $k-1$ noise channels are selected and used as noise for every sensitive Page a user is actually interested in and subscribes to. These batch of noise Pages remain constant over time. However, it is theoretically possible that some of the Pages included in this batch k will disappear at some point in time. Indeed, people may choose to delete their accounts, corporations may go out of business, or organizations may change their focus. In such cases, their Page (say D) in the online social network will be deleted or it will become inactive. This will create two kinds of problems:

First, people who used to be subscribed to D cannot just stop being subscribed to it. Moreover, they cannot stop being subscribed to the noise Pages, which were selected along with D, too. If the users unsubscribe from the above Pages, the service provider will immediately correlate D's disappearance with the users' change of subscribing patterns, and figure out that the users were actually interested in Page D. Fortunately, this not particularly worrisome. Indeed, the users should continue being subscribed to D along with their selected noise Pages. They will just pay a small overhead in downloading content from Pages they do not need anymore.

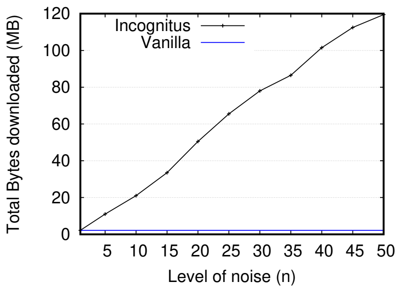
The second problem is that people who included D among their noise Pages may be more exposed to the service provider. Indeed, if their noise Pages start disappearing one after the other, it seems that the service provider will be in a better position to find what is the Page they are really interested in. An obvious approach would be to add other noise Pages in the place of the disappeared ones. Unfortunately, this is not correct, as the service provider will easily figure out that both the disappeared ones as well as the recently subscribed ones are noise Pages. Surprisingly, the best approach for the users is to *do nothing*: they should just keep being subscribed to the Pages even if they disappear one after another all the way to the last one. The key observation here is that both users who are interested in D as well as users who are not interested in D but have just included D as noise, should do the same thing: *nothing*. In this way, the service provider will not be able to differentiate, which users are interested in D and which are not.

6 Implementation

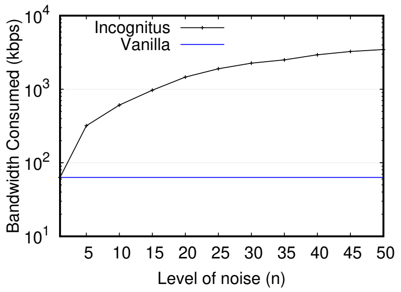
In order to evaluate the feasibility and efficiency of *Incognitus*, we have implemented an extension for the popular browser of Mozilla Firefox. As a case study, we used the online social network of Facebook. In this social network, we assume users interested in subscribing to Facebook Pages.

Facebook Pages [9] are for businesses, brands and organizations to share content and can be customized by publishing stories, hosting events, adding apps and more. A user who subscribe to a Page can get timely updates directly delivered in her account's NewsFeed . On Facebook, a NewsFeed is a list of updates on user's Facebook home page. NewsFeed includes status updates, photos, videos, links, app activity and Likes from people, Pages and groups that the user is subscribed to on Facebook.

In *Incognitus* browser extension, the user can subscribe to a sensitive Page, by using the standard Facebook Like button (either from inside the Facebook platform or from



(a) Total Bytes downloaded as a factor of the different level of noise



(b) Average bandwidth consumption as a factor of the different level of noise

Fig. 2: The overhead of noise in terms of additional downloaded bytes , when a user browses a sensitive Page for the first time with no previously stored content

a website by using the appropriate Facebook widget). To achieve the same user experience, in *Incognitus* we intercept all Like requests and check whether they correspond to a Page considered as sensitive (i.e. if the corresponding Page is included in the *SP*). If so, the extension based on the k parameter set by the user, on the background, randomly selects the additional $k-1$ Pages from the list *SP* of sensitive Pages, which will be used as noise. Finally it sends a Like request to each one of the k (real and noise) sensitive Pages. The browser extension, internally, keeps the list of the noise Pages it has used and the real Page with which they are associated.

Our prototype implementation is built using the Firefox Add-on SDK [23], due to the convenience it provides regarding the access of the browser extensions to the local storage. In addition, it is developed using Javascript along with JQuery and Mutation Summary libraries [35] according to the Facebook documentation [11] and policies [10].

7 Performance Evaluation

For *Incognitus*'s performance evaluation, we used a PC equipped with an AMD FX-6300 processor (3.5 GHz, 8MB L3 Cache) and 8GB RAM. We populated the set S with 60 Pages of categories we personally consider as privacy-sensitive: medical diseases, political parties, sexual preferences, etc.

Bandwidth consumption: It is easy to anticipate that the introduction of noise leads to an increased bandwidth consumption in order to obfuscate the sensitive transmitted information. The traffic volume that is generated depends on the noise level the user has chosen; specifically, on the value of parameter k. This parameter practically denotes that, with our system, the user roughly downloads k times more bytes.

First, we monitor the traffic generated from the user's browser when visiting a sensitive Page, for several values of k. In this experiment, we measure the consumed bandwidth in the worst case: when the user visits the Page for the first time and there is no pre-stored content on her disk. At Figure 3, we observe the total bytes downloaded

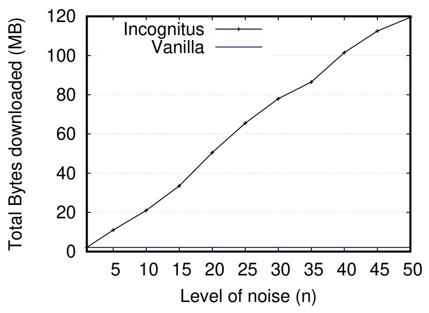


Fig. 3: Total Bytes downloaded as a factor of the different level of noise

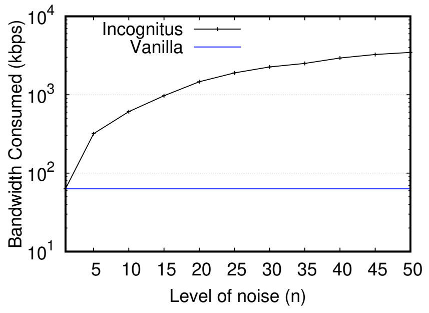


Fig. 4: Average bandwidth consumption as a factor of the different level of noise

and at Figure 4 the traffic load generated, for different levels of noise, both for a user subscribed to a sensitive Page. Moreover, we include the vanilla case $k = 1$, when *Incognitus* is disabled. We noticed that the bandwidth consumed is reasonably low compared to the vanilla case, adding less than an order of magnitude overhead in case of $k = 10$.

It is worth recalling at this point, that these results regard the first fetch of a Page where no data are stored on disk. After the first fetch, the delta approach is used as we describe at Section 3. Then, *Incognitus* periodically downloads the updates from all sensitive Pages, both noise, and real ones. This way, we are able to reduce both the latency imposed and the bandwidth consumed, since we retrieve the content from the disk instead of fetching data from the Facebook server. In the following experiment we measure (i) the total bytes downloaded when the user browses P without having any content pre-fetched on disk and (ii) the average bytes downloaded periodically with the delta approach.

To quantify the bandwidth consumption over time, we perform measurements with our prototype for 30 minutes. Specifically, we subscribe to a sensitive Page and fetch it for the first time, without having any content previously stored on the disk. In Figure 5, we plot the results of this experiment using 3 different values of k, the vanilla case (k=1), a low one (k=5), and a higher one (k=25). As we see, there is an increased spike at the beginning, when the browser downloads the content for the first time, after that the bandwidth consumed is close to zero. In this figure, we also see the periodic delta mechanism at 10th and 20th-minute where *Incognitus* checks for updates.

Browsing Latency: The most important aspect that can degrade the browsing experience of a user is latency: the time needed to render and show to the user the content she is waiting for. In the following experiment, we set out to explore the delay imposed by *Incognitus* when the user visits a privacy-sensitive Page. In this experiment we inspect the worst case scenario: when the user fetches the Page as soon as she subscribes to it and therefore no data were previously locally stored. In this scenario, *Incognitus* before reconstructing the web-page the user requested to see, has to download the set of k noise Pages and filter the noise content. As a consequence, we fetch a sensitive Page P with our browser extension enabled and we measure the time it takes to load the Page's full content. We run the experiment 100 times for different noise levels and

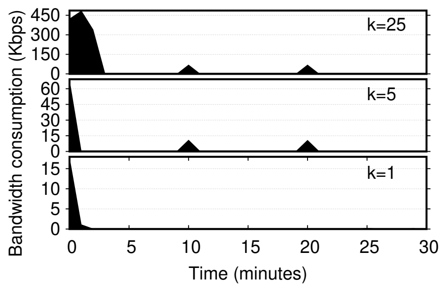


Fig. 5: Bandwidth consumed as a factor of time for the initial fetch and the delta updating mechanism.

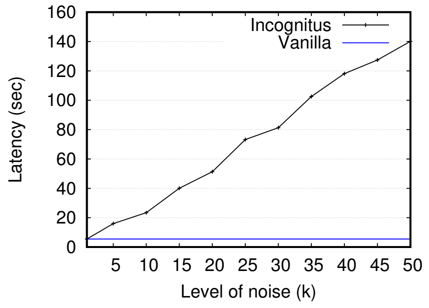


Fig. 6: Average rendering latency as a function of different level of noise, the first time a user browses a sensitive Page.

in Figure 6 we plot the results. As expected, the higher the amount of noise we add, the larger the imposed latency is. We see that for *k*=10 the additional latency is about 22 seconds, which we believe is unable to harm the users browsing experience; at the same time, the privacy of her interests is preserved. We have to note that, this latency does not correspond to an optimal scenario. The actual Page is always at the last position in the sensitive set, while at the same time, the background fetch of sensitive pages is performed sequentially; thus, latency could be further reduced by leveraging the multiprocess architecture and the HTML5 Web Workers API of modern browsers.

8 Related Work

Our work's foundations lie in the concept of k-anonymity [33]. K-anonymity provide privacy guarantees that the individuals who are the subjects of a released dataset cannot be re-identified. With K-anonymity an individual cannot be distinguished from at least k-1 individuals whose information also appear in the data. L-diversity [21] and [20] extend k-anonymity by handling some of the weaknesses.

TrackMeNot [13] is a Firefox add-on designed to achieve privacy in web search by obfuscating user's queries within a stream of programmatically-generated decoys. For each real query submitted to the search engine, TrackMeNot also submits additional queries to confuse the search engine and introduce doubt for the user's real queries. TrackMeNot utilizes the same idea (i.e. obfuscation) to protect user's privacy when using a search engine, although it has one fundamental difference: the set of possible queries is not limited to a finite set as Facebook Pages do. So there is a good possibility that a user submits a rare query, which would enable the search engine to accurately find her interests. Moreover, an adversary may be able to find a user's interests by studying successive sequences of queries [3] and make use of clustering approaches [8].

In [28] authors propose k-subscription to conceal the user's interests in microblogging services. Their approach relies on introducing noise in order to decrease the disclosure probability of the service provider. They propose and evaluate two algorithms to compute the necessary amount of noise that needs to be used based on the popularity of the channel the user is interested in following. Although k-subscription is related to our work, it assumes that users subscribe to a channel using the standard mechanism (e.g., follow) that the microblogging service offers. In *Incognitus*, we assume users that may avoid subscriptions thus browsing content from Pages manually and we assume service providers that deploy more aggressive behavioral tracking techniques, monitoring any action the user may perform with a Page's content.

Single sign-on (SSO) services (like Facebook Login [14]) often improve the user's browsing experience by offering a convenient way to register to third party websites and adding a social dimension to the user's experience. But their existence and popularity also raise significant privacy concerns, since third-party Web sites are granted access to personal information. In addition, the social network can observe the Web sites that the user visits as well. To mitigate that problem, SudoWeb [17] proposes a system that enables users to surf the Web using downgraded sessions with the single sign-on platform, i.e., stripped from excessive or personal information, and with a limited set of privileged actions. Although both SudoWeb and *Incognitus* belong to the domain of privacy-preserving browsing, SudoWeb focus on enhancing the user's privacy when visiting third-party Web sites and not when browsing an OSN.

9 Conclusion

Online social networks have changed the way we fulfill our daily information needs. However, using OSN's for information consumption is a double-edged sword. The low cost and timely access to information come with a cost to user's privacy: the online social network provider is able to collect information regarding its users' interests, which sometimes may regard privacy-sensitive topics.

To cope with this situation, we propose *Incognitus*: a tool to (i) allow users to browse Pages with privacy-sensitive content in an offline mode, without disclosing their interactions to the service provider. And at the same time, (ii) by using an obfuscation-based mechanism conceal the privacy-sensitive topics a user is interested in, thus reducing the accuracy with which the service provider can determine the user's interests. We implemented our approach as a browser extension using Facebook as a case study. Our experimental evaluation shows that *Incognitus* has minimal bandwidth requirements and imposes reasonable latency for a normal amount of noise, when at the same time is able to adequately preserve the privacy of the actual interests of the user.

Acknowledgement: The research leading to these results has received funding from European Unions Marie Sklodowska-Curie grant agreement No 690972. The paper reflects only the authors view and the Agency and the Commission are not responsible for any use that may be made of the information it contains.

References

1. What does the general data protection regulation (gdpr) govern? https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-does-general-data-protection-regulation-gdpr-govern_en, Mar 2016.
2. G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on CCS*.
3. E. Balsa, C. Troncoso, and C. Diaz. Ob-pws: obfuscation-based private web search. In *IEEE Symposium on S&P'12*.
4. M. S. Brown. When and where to buy consumer data (and 12 companies who sell it). http://www.forbes.com/sites/metabrown/2015/09/30/when-and-where-to-buy-consumer-data-and-12-companies-who-sell-it, 2015.
5. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association.
6. N. DRAKE. Help, i'm trapped in facebook's absurd pseudonym purgatory. http://www.wired.com/2015/06/facebook-real-name-policy-problems, 2015.
7. P. Eckersley. How unique is your web browser? In *Proceedings of International Symposium on PETS'10*.
8. Y. Elovici, B. Shapira, and A. Meshiach. Cluster-analysis attack against a private web solution (praw). *Online Information Review*, 30(6):624–643, 2006.
9. Facebook Business. Facebook Pages. https://www.facebook.com/business/products/pages, 2017.
10. Facebook for Developers. Facebook platform policy. https://developers.facebook.com/policy/, 2016.
11. Facebook for Developers. Facebook developer documentation. https://developers.facebook.com/docs/, 2017.
12. A. Greenberg. Why facebook just launched its own "dark web" site. https://www.wired.com/2014/10/facebook-tor-dark-site/, 2014.
13. D. C. Howe and H. Nissenbaum. Trackmenot: Resisting surveillance in web search. *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, 23:417–436, 2009.
14. F. Inc. Facebook login. https://developers.facebook.com/docs/facebook-login/, 2018.
15. F. Johansson, L. Kaati, and A. Shrestha. Timeprints for identifying social media users with multiple aliases. *Security Informatics*, 4(1):1, 2015.
16. S. Kamkar. Evercookie - virtually irrevocable persistent cookies. http://samy.pl/evercookie/, 2010.
17. G. Kontaxis, M. Polychronakis, and E. P. Markatos. Minimizing information disclosure to third parties in social login platforms. *Int. J. Inf. Secur.*, 11(5):321–332, Oct. 2012.
18. S. Krug. Reactions now available globally. https://newsroom.fb.com/news/2016/02/reactions-now-available-globally/, 2016.
19. M. Kumar. Why facebook is buying whatsapp for $19 billion? http://thehackernews.com/2014/02/why-facebook-is-buying-whatsapp-for-19.html, 2014.
20. N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.
21. A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.
22. J. Mander. Why facebook is tracking time spent on newsfeeds. http://www.globalwebindex.net/blog/why-facebook-is-tracking-time-spent-on-newsfeeds, 2015.
23. Mozilla Developer Network and individual contributors. Add-on SDK. https://developer.mozilla.org/en-US/Add-ons/SDK, 2016.
24. E. P. Papadopoulos, M. Diamantaris, P. Papadopoulos, T. Petsas, S. Ioannidis, and E. P. Markatos. The long-standing privacy debate: Mobile websites vs mobile apps. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 153–162, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.
25. P. Papadopoulos, A. A. Chariton, E. Athanasopoulos, and E. P. Markatos. Where's wally?: How to privately discover your friends on the internet. In *Proceedings of the ASIACCS '18*.
26. P. Papadopoulos, N. Kourtellis, and E. P. Markatos. Cookie synchronization: Everything you always wanted to know but were afraid to ask. *arXiv preprint arXiv:1805.10505*, 2018.
27. P. Papadopoulos, N. Kourtellis, and E. P. Markatos. Exclusive: How the (synced) cookie monster breached my encrypted vpn session. In *Proceedings of the 11th European Workshop on Systems Security*, EuroSec'18, 2018.
28. P. Papadopoulos, A. Papadogiannakis, M. Polychronakis, A. Zarras, T. Holz, and E. P. Markatos. K-subscription: Privacy-preserving microblogging browsing through obfuscation. In *Proceedings of the ACSAC '13*.
29. RT. Privacy betrayed: Twitter sells multi-billion tweet archive. https://www.rt.com/news/twitter-sells-tweet-archive-529/, 2012.
30. E. SHEARER and J. GOTTFRIED. News use across social media platforms. http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/, 2017.
31. N. SINGER and J. B. MERRILL. When a company is put up for sale, in many cases, your personal data is, too. http://www.nytimes.com/2015/06/29/technology/when-a-company-goes-up-for-sale-in-many-cases-so-does-your-personal-data.html, 2015.
32. O. Solon. Facebook says cambridge analytica may have gained 37m more users' data. https://www.theguardian.com/technology/2018/apr/04/facebook-cambridge-analytica-user-data-latest-more-than-thought, 2018.

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

34. J. Wakefield. Social media 'outstrips tv' as news source for young people. https://www.bbc.com/news/uk-36528256, Jun 2016.

35. R. Weinstein. Mutation summary. https://github.com/rafaelw/mutation-summary, 2015.

36. W. Youyou, M. Kosinski, and D. Stillwell. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences*, 112(4):1036–1040, 2015.