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

Abdelberi Chaabane, Yuan Ding, Ratan Dey, Mohamed Ali Kaafar, Keith Ross

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

Abdelberi Chaabane[1], Yuan Ding[2], Ratan Dey[2], Mohamed Ali Kaafar[1,3], Keith W. Ross[2]

[1]INRIA, France [2] Polytechnic Institute of NYU, USA [3] NICTA, Australia

Abstract. We examine third-party Online Social Network (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 use this platform to study the behavior of 997 Facebook applications and 377 RenRen applications. We find that the Facebook and RenRen applications interact with hundreds of different fourth-party tracking entities. More worrisome, 22% of Facebook applications and 69% of RenRen applications provide users' personal information to one or more fourth-party tracking entities.

1 Introduction

OSN user profiles represent a rich source of personal information, including demographic information, users' interests and their social relations. Privacy threats resulting from this direct exposure of personal information have been widely publicized and researched. 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.

1.1 Related research

Krishnamurthy and Wills examined privacy leakage that can occur from OSNs directly to external entities [5, 6]. Chaabane et al. evaluated the tracking capabilities of the OSNs in [1]. However, to our knowledge, this is the first paper to explore indirect privacy leakage to external entities via third-party applications. Another line of related research has analyzed the permission systems in third-party applications. Chia et al. showed that community ratings are not reliable and that most applications request more permissions than needed [2]. Frank et al. extended this work, showing that Facebook permissions follow a predefined pattern and that malicious applications deviate from it [4]. Finally, Xia et al. proposed *Tessellation* [9] 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.

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 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

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

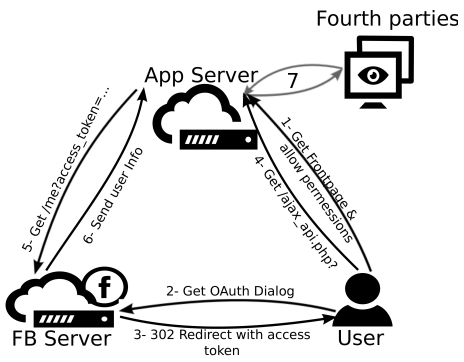


Fig. 1: An overview of the Facebook application architecture

an ad) and Javascript (e.g., sending data to an analytics service). Observe that when these entities are contacted, the referrer field is automatically filled with the current page (i.e. application main page) URI. Our focus in this paper 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 fulfil the task.

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 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

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

[3] `https://developers.facebook.com/docs/appcenter/guidelines/`

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 [7], 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.

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 un this paper serves as a *lower bound*.

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 2a 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.

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

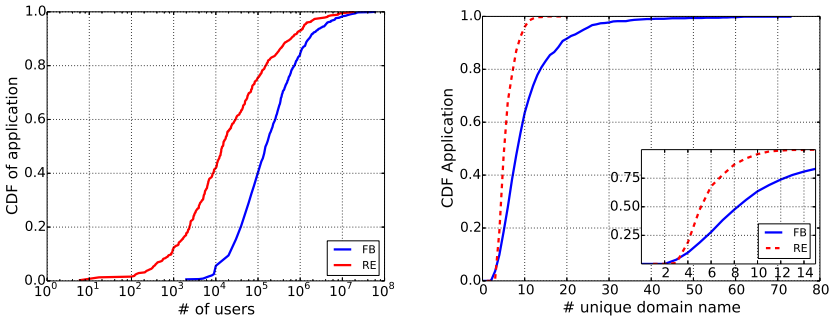
Table 1: Most frequent app companies for Facebook (997 apps)

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 2: Most frequently requested permissions for Facebook (997 apps)

Application companies: We were able to collect the publisher's company name for 845 applications. Table 1 shows the top seven publishers among the applications

[4] `http://seleniumhq.org/`



(a) (b)

Fig. 2: (a) Application popularity, (b) Number of contacted servers for each application

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.

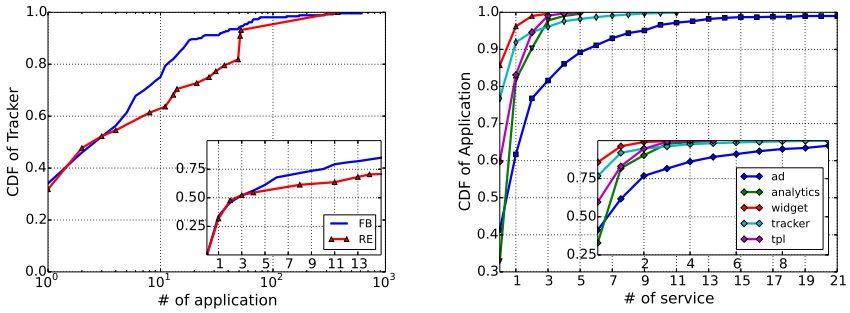
Permissions: Table 2 shows the most frequently requested permissions. As expected a large fraction of applications request permission to obtain the basic information, which as mentioned previously, encompasses not only the user name but also all information the user makes public in the public profile. 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.

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 (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 2b 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.



(a) (b)

Fig. 3: (a) Tracker distribution for third-party apps (b) Distribution of tracker categories

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 3a. 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).

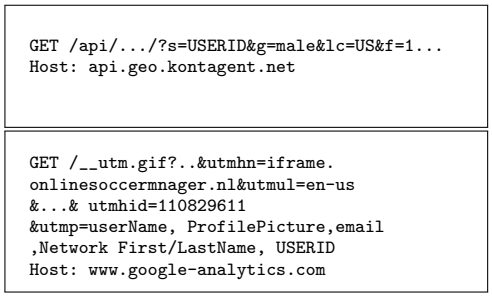
Tracker Categories We now further classify the set of identified trackers into five categories: ad networks (e.g., Google Adsense) referred to as Ad; analytical services (e.g., Google analytics) referred to as analytics; online service plug-ins (e.g., Twitter connect) referred to as widgets; ad-network 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 3b 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 sell-

ing 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.



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 4 & Table 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

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.

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]

5.2 Data leakage classification

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

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

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. [6] 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.

5.3 Statistics

Table 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 [8]. 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 [3]. 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 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

[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

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 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.

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 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 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.

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

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 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 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 5: Number of attributes leaked per application

data sanitization and hence can be considered unintentional. In 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).

Acknowledgements Thanks to Alan Mislove for shepherding this manuscript and the anonymous reviewers for their valuable feedback. This research was funded by French ANR project PFlower.

References

1. A. Chaabane, M. A. Kaafar, and R. Boreli. Big friend is watching you: Analyzing online social networks tracking capabilities. In *WOSN*, 2012.
2. P. H. Chia, Y. Yamamoto, and N. Asokan. Is this app safe?: A large scale study on application permissions and risk signals. In *WWW*, 2012.
3. R. Dey, Z. Jelveh, and K. Ross. Facebook users have become much more private: A large-scale study. In *PERCOM Workshops*, 2012.
4. M. Frank, B. Dong, A. Porter Felt, and D. Song. Mining permission request patterns from Android and Facebook applications. In *ICDM*, 2012.
5. B. Krishnamurthy and C. E. Wills. Characterizing privacy in online social networks. In *WOSN*, 2008.
6. B. Krishnamurthy and C. E. Wills. On the leakage of personally identifiable information via online social networks. In *WOSN*, 2009.
7. J. R. Mayer and J. C. Mitchell. Third-party web tracking: Policy and technology. In *S&P*, 2012.
8. D. Perito, C. Castelluccia, M. Kaafar, and P. Manils. How unique and traceable are usernames? In *PETS*, 2011.
9. N. Xia, H. Song, Y. Liao, M. Iliofotou, A. Nucci, Z. Zhang, and A. Kuzmanovic. Mosaic: Quantifying privacy leakage in mobile networks. In *SIGCOMM*, 2013.