Challenges of Keyword-Based Location Disclosure

Chris Riederer, Augustin Chaintreau
Columbia University
New York, NY, USA
{mani,augustin}@cs.columbia.edu

Jacob Cahan
Brown University
Providence, RI
Jacob_Cahan@brown.edu

Vijay Erramilli
Telefónica Research
Barcelona, Spain
vijay@tid.es

ABSTRACT

A practical solution to location privacy should be incrementally deployable. We claim it should hence reconcile the *economic* value of location to aggregators, usually ignored by prior works, with a user's *control* over her information. Location information indeed is being collected and used by many mobile services to improve revenues, and this gives rise to a heated debate: Privacy advocates ask for stricter regulation on information collection, while companies argue that it would jeopardize the thriving economy of the mobile web.

We describe a system that gives users control over their information and does not degrade the data given to aggregators. Recognizing that the first challenge is to express locations in a way that is meaningful for advertisers and users, we propose a *keyword* based design. Keywords characterize locations, let the users inform the system about their sensitivity to disclosure, and build information directly usable by an advertiser's targeting campaign. Our work makes two main contributions: we design a market of location information based on keywords and we analyze its robustness to attacks using data from ad-networks, geo-located services, and cell networks.

Categories and Subject Descriptors

Security and Privacy [**Human and societal aspects of security and privacy**]: Usability in security and privacy

General Terms

Design, Economics, Human Factors

Keywords

Privacy, Location information, Mobile advertising

WPES'13, November 4, 2013, Berlin, Germany.
Copyright 2013 ACM 978-1-4503-2485-4/13/11 ...$15.00.
http://dx.doi.org/10.1145/2517840.2517862 .

1. INTRODUCTION

The rapid adoption of smart phones and tablets has led to innovative applications and services that exploit location information. Location information is increasingly used to drive advertising – location-based targeting generates four times as much revenue per impression compared to ads without location data[1]. Even brick-and-mortar stores use location data, with retailers using cell phones' WiFi signals to learn where customers spend time in their stores[2].

There are many privacy concerns surrounding the use of this data. For example, many applications access location information even when such information is not needed, and may share it with multiple third parties, leading to privacy concerns [6, 20] and attracting the attention of regulators [7, 1]. This work focuses on location information generated in real-time by users with mobile devices.

Many privacy concerns around location information are rooted in the mobile application ecosystem. Most mobile services and applications are free and operate by collecting personal information (browsing activity, location, etc.) and monetizing this information through targeted ads [15]. Because it affects their profits, companies that are a part of the mobile application ecosystem oppose any regulation that may restrict access to location data and claim that the "cost" of a privacy bill threatens the web's general economy and ultimately hurts customers. In fact, one may argue that users today exchange their data for services. An ideal privacy solution therefore should provide adequate privacy protection to the user while simultaneously enabling service providers to collect and monetize data. Our objective is to lay the groundwork for a comprehensive and deployable solution to location privacy.

In contrast to previous work, we aim to reconcile the users' control over their location information with its commercial value. This approach raises three challenges: (1) The solution should be *incrementally deployable.* It must easily integrate with current devices and practices while giving all parties an incentive to participate. (2) The solution should be *robust* against threats from its participants. Advertisers should not be able to access data without compensating users or access more than the users specify. Users should not be able to benefit from seeking unfair compensation. (3) The solution should be *easy to use.* The system should be easily understood by both users and advertisers.

Our solution is based on selective disclosure; users decide what location information they want to disclose. At

[1] `http://bit.ly/vXWdsw`
[2] `http://nyti.ms/15vLRva`

the heart of our solution is a *keyword-based* method where keywords are associated with locations, and the decision to release locations is based on keywords. We observe that keywords are naturally associated with the elements that define this problem, but also offer a strong abstraction to handle location data. In order to drive the adoption of the solution, we propose providing economic compensation to the users for the location information they disclose. Application and web service providers bid to gain *access* to users at these specific locations in real-time.

Our main contributions are: (1) The design of a keyword-based system that integrates well into today's location collection and monetization. Our solution requires no change on users' devices, a minimum level of indirection, and addresses goals like usability, deployability and scaling (Sec. 2). (2) A test of our solution's usability and relevance with a small scale trial on real users. While this experiment is too small to form statistically significant conclusions, it allowed us to test the feasibility of our design (Sec. 3). (3) An analysis of how such a system can offer different levels of protection against various threats, including freeriding, inference attacks using auxiliary information, and user misconduct (Sec. 4).

2. OVERVIEW

This section presents the motivation, design and advantages of a location disclosure system based on keywords.

2.1 A keyword-based solution

Our requirements calls for a solution to share information about location monetized by ad-networks and 3rd party aggregators through *selective disclosure*. For the user to retain control, our privacy solution should address *how* the information is released, under *which conditions* the information is released and to *whom*, as seen in previous ones, e.g. Koi [9],

To specify *how* and *under which conditions* location information is released, we choose to use keywords. While the information that is released is a latitude longitude pair (lat-long), the decision to disclose is based on associated keywords. Users who are comfortable disclosing location under certain circumstances [12] opt-in to reveal lat-long associated with keywords of their choices. An example would be a street that has many restaurants serving different cuisines, it would have keywords like "restaurant, Thai, French, Indian" associated each with the lat-long of each particular venue. The use of keywords brings important advantages: (i) Keywords let us deal with the problem of location privacy at a higher abstraction than coordinates or even location descriptors as in Koi [9]. (ii) Keywords are user friendly: instead of having to decide the sensitivity of every location, users decide on a much smaller set of keywords that they are comfortable releasing or not. (iii) Today's ad-networks function primarily around keywords, thereby a solution around keywords can make it easier for ad-networks to adopt and use. (iv) As there can be a finite set of keywords associated with any location, and the association of a keyword with a location typically remains for long periods of times, modifying keywords associated with a location is easy, making the solution scalable.

Our solution compensates users *economically* for information they release to aggregators and ad-networks. Economic incentives can nudge more users towards adoption, as concerns about privacy alone are rarely sufficient. Concrete incentives also sometimes reduce users' cognitive biases when it comes to perceiving their privacy [3]. Specifying to *whom* the information is released is implicitly done by a market. In principle, any parties that can pay for it is legitimate. In practice, this agreement should be facilitated by a trusted third party who vet the parties and send information about the user *only* for locations she agreed on, upon payment.

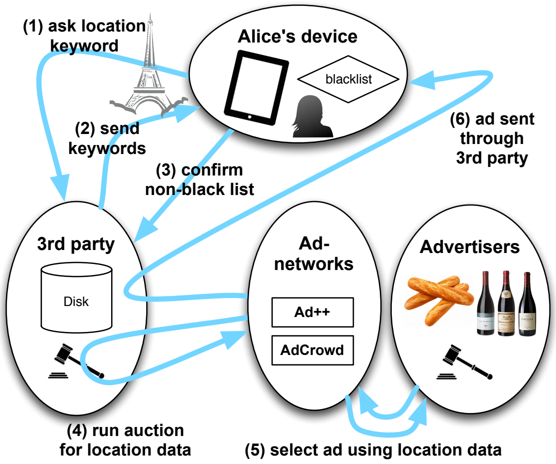

(1) ask location keyword
Alice's device
blacklist
(2) send keywords
(3) confirm non-black list
(6) ad sent through 3rd party
3rd party
Disk
Ad-networks
Ad++
AdCrowd
Advertisers
(4) run auction for location data
(5) select ad using location data

Figure 1: Solution overview

The design we next describe is meant to operate under the following set of **assumptions**. Given the amount of press on privacy related issues, we believe that the PR backlash in the case of a serious privacy violation will make such violations undesirable. As a consequence, we provision against an *honest-but-curious* advertiser. It means the adversary complies with the system but it can exploit the information that is gathered for its own interest. We provide safeguards against inference and linkage attacks. We also assume that the mobile OS used complies with user's privacy, hence not sharing location information with any application once the user stated that request. Note that the architecture presented next is oblivious to a background service model (passive, potentially continuous tracking) or a check-in model.

2.2 Design and Example

The architecture consists of the following components: (i) a keyword server which maps physical locations to keywords (ii) a location blacklist module which contains a list of sensitive keywords, communicates with the keyword server, and reveals non-sensitive locations (iii) a blocking module in the network that blocks access to various parties, (iv) a market that puts up for sale information about locations visited by the user that are not in the blacklist, and (v) a module that grants *access* to the user for parties that pay, after purchasing access on the market. With the exception of (ii), which can be a simple smartphone app, all modules are stored in the network; *no* changes are required on the device.

A high-level diagram is shown in Fig. 1. We describe the process with a simple example. Alice is willing to share certain locations and would like to hide her presence at other locations, a typical occurrence [12]. Alice wants to buy bread, shop for wine, and go to the Libertarian party headquarters. She would like to conceal her political leanings. Alice would therefore put 'Libertarian, Politics' as keywords in her *blacklist module*. We describe in Sec. 3.1 how the blacklist formation can be simplified through nested menus and re-ordering. We assume the third party is trusted and leave lowering this requirement to future work.

As Alice arrives at the bakery, her network activity goes through the *blocking module* that runs a mix-network to conceal her real network address, and provides privacy protection like dropping cookies to third parties, overwriting `referer` headers etc. [14] (see Sec. 3.1 for more on implementation). At every location, Alice's device contacts the *keyword server* which translates locations to keywords. A check is then made against the blacklist to verify if Alice is comfortable releasing this information. If a location has multiple keywords and *any* of them are on the blacklist, it is considered private. Once a location passes the check, it is put on the *market* for sale with a unique user-id and the keywords. This user-id is generated independently and can be periodically changed. The information then is (UID_{Alice}, (lat_1, long_1), Bakery). As she arrives at the wine shop, the information on the market will be (UID_{Alice}, (lat_2, long_2), Wine Shop), as the wine shop also passes the blacklist test. Ad-networks can pay to *access* Alice based on these two locations released. The payment will be credited to Alice, with a small fraction taken by the third party. The third party then fixes a network address to reach Alice at the wine shop and conveys it to the ad-networks. Alice can receive a targeted ad (via an app or via SMS) for a particular wine.

As soon as Alice moves out of the wine shop, her network address changes and her location again is not known to anyone but the trusted third party. When she is close to the Libertarian party headquarters, the check against the blacklist returns a positive result, and this location is not revealed to anyone.

2.3 Summary of Advantages

Now that we've described the system, we discuss the benefits of the system for various parties.

Users obtain monetary payment for their data and privacy through choice. The architecture operates in the network and hence, users do not need to make changes to their devices. If information is leaked or shared between colluding ad-networks, these parties would have to gain access to the user to monetize this information – and unless these parties have paid, they are prevented from gaining access to the user. Hence, we protect against adversaries aiming to extract economic gain. We deal with adversaries who try to infer the identity of users or blacklisted keywords in Sec. 4.

The keyword system also benefits the user. If a user is visiting a place they are unfamiliar with, they may not be accustomed to what areas are privacy sensitive. Because keyword mappings work in any location, a user's privacy is protected even in unfamiliar areas. Additionally, a user may simply not realize the privacy sensitive nature of a location they are in. Because all traffic is directed through our system, if a user starts using a location-based service at a location they don't realize is privacy sensitive, our system can catch it and warn the user before they complete the action.

Ad-networks and aggregators can obtain non obfuscated data in a legal way, minimizing data breaches. As the data is 'bought', the ad-networks can micro-target. Ad-networks and advertisers can easily make sense of the location data, as keywords are already used for context in current online advertising systems. Rather than having advertisers need to bid specifically for each location, ad-networks can simply run auctions for ad impressions in locations associated with specific keywords.

Application developers do not need to alter their code as we operate directly in the network. Applications serve as a conduit to show ads to the users, much as they do today.

Finally, mapping locations to keywords helps our system evaluation. Ad-networks constantly run many auctions of impressions to a customer searching for a specific term. Cost-per-click (CPC) data from ad-networks hence reflects the overall advertising demand on this topic. We show how CPC data may be collected and used to understand the economic value of locations.

3. DEPLOYMENT AND USER STUDY

We now describe in detail how such a system could be implemented. We additionally discuss a small-scale deployment and user study we ran in order to demonstrate the system's feasibility.

3.1 Implementation

An implementation consists of the five components described in section 2.2: a keyword server, a location blacklist module, a network blocking module, an information market, and an access module.

Our **keyword server** used Yelp's API. Each time a device uploaded a lat-long to the server, we queried Yelp to find the categories of each location within 50 meters. This is a possible area for improvement; in future work, the radius of a query could change depending on an estimate of the device's current accuracy or a user's privacy preferences. The categories were then sent to the device.

Future implementations could likewise map locations to keywords by reusing online services such as Yelp, Google Places, and Foursquare. A "folksonomy" approach can be used where users label a map over time, possibly receiving incentive. To encourage tagging of privacy-sensitive locations, the system can allow anonymous tagging.

The **location blacklist** module was written as an Android application, using the phone's GPS. The app, available on Google Play[3], was designed to give users a way to edit a blacklist and monitor which locations (and corresponding keywords) were being recorded. We used Yelp's 885 categories as our keywords during the study, meaning users had a large number of potential keywords to blacklist. To make adding keywords to the blacklist manageable, all possible keywords were placed in a nested menu by category. Thus, a user could select and de-select whole categories of keywords with a single button press, but could also expand categories to select specific words. We placed categories previously defined to be sensitive [2] near the top of this list, and alphabetized all potentially less sensitive categories. The blacklist was stored locally on the phone. *At no point did the authors have access to a study participant's blacklist.* Each half hour, the app would passively check the keywords in the current location and upload the location and keywords to the server only if no keywords were on the blacklist.

For the purposes of our small scale user study, we did not create a **blocking module**. In a full implementation, it would be necessary to block any third-party advertisers who did not participate in the system. The connections to ad-networks and aggregators (AdMob, Flurry Analytics

[3] Link to app: `http://bit.ly/13qOMqC`

etc.) can be blocked by a proxy and spoofing the MAC address. All necessary proxies already exist: Privoxy comes with advanced filtering capabilities and handles rewrites of the HTTP headers like the 'referrer' header to prevent leakages of any form, and mitmproxy can handle SSL[4]. In addition, users could upload their SSH certificates to enable the module in the middle to masquerade as the user. From an application's perspective, no logic is broken. Even for location based services like Foursquare or maps, an unintentional checkin or a search at a private location can be prevented by checking against the blacklist – an added benefit.

As this deployment was meant for exploratory purposes, we did not connect the system to any ad exchanges. Instead of implementing a **market** or **access module**, we simulated the incentives and costs a user might experience while using our system. All participants received a $10 for participating and were entered into a lottery. Each user was instructed that releasing more 'valuable' information would give them a higher chance of the lottery. We did not disclose the exact method of valuing information, mimicking the opaque way in which information would be priced in a real implementation of the system. The intention was that this would incentivize users to release more information. To simulate the costs of disclosing information, we publicly displayed a user's non-blacklisted locations on a web interface, viewable at `keyword.cs.columbia.edu`. In a real system, a user would risk that her information is used improperly or released to those who might use it in a damaging way. We believed that publicly displaying a user's information simulated this risk. To increase the publicity of their information, we instructed users to post the link on a social media site, such as Facebook or Twitter, and email us a screenshot.

To protect users' safety, users could contact us at any point if they were concerned about an unintentional location release. Additionally, any time a data point was recorded, we delayed making it public by 24 hours. Users could see their data points in real-time via a password-secured link.

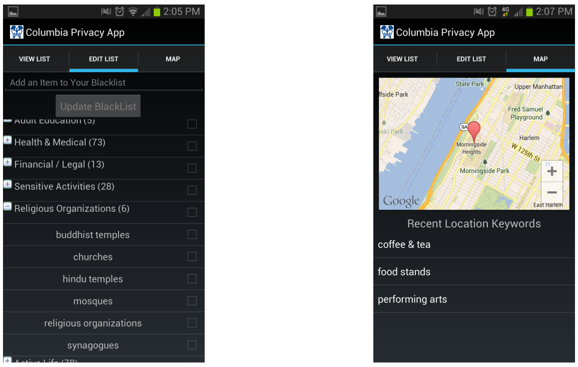


Figure 2: User Interface: (left) managing keywords black list, (right) visualizing locations released.

3.2 Deployment and Observations

We deployed our implementation with six users for two weeks. Users were geographically diverse, located in multiple cities throughout the United States. Study participants were recruited through advertising on social networks and were primarily adults in their mid-twenties.

After the study, we asked users to complete a survey. Our study was too small to make general conclusions, but we present results here to inform future work. Users easily understood both the keyword system and the interface. Users were divided on how well they felt the system secured their privacy, with some users concerned that our mapping of keywords to locations was not precise enough. Our users expressed a range of privacy sensitivities. Some did not use the blacklist and others used the blacklist to hide sites they associated with social stigma or that they thought would send negative signals to employers, insurers or the police.

4. MITIGATING ATTACKS

Having introduced the design of the system, we now turn our focus to one of our key goals: protecting the privacy and value of system participants.

4.1 Attacks on the Value of User Data

Our system prevents an adversary from economically benefiting by using information about a user without properly compensating her.

Ad-networks may try to build up interest profiles of users over time in order to better target ads later *without* compensating the user. Even if a user's anonymous ID is changed regularly, human mobility patterns are periodic and somewhat predictable, making it easy to link a current anonymous ID to an older one[5]. Our system does not prevent such profiling, and it even makes it easier as the market announces which data is for sale. However, we ensure that this strategy has no economic benefit, for the following reason: all traffic flows go through a proxy, and an ad network who does not pay will receive the identity and location of a user, but a random temporary ID. Then the ad-network, although it has a rich profile of user u, is not able to recognize u as the recipient of an ad. For the same reason, ad-networks do not gain by colluding or reselling the information. Unless a payment is made, the identity and location of u is unknown, and the profile alone does not aid targeting.

A related issue is trajectory-based profiling. If an ad-network learns the habits of a particular user over time, the ad-network can show ads based on where a user *is likely to be* rather than paying for an exact location. Again, ad-networks must always pay to be able to access a user's identity. Care must be taken, however, to make sure that a user does not unwittingly display information about a visited blacklisted location based on her trajectory: *e.g.* location B is sensitive and locations A and C are not, and the only way to get to C from A is via B). If Alice checks in at point A and then at point C, ad-networks may infer that she visited B. Such attacks are not likely, and can be dealt with by ensuring that after visiting a blacklisted location a minimum amount of time has passed before disclosing a location.

One concern is if an app works to circumvent the proxies and leak information about either the location or the identity of the user. Against location leakage, one solution is to substitute a fake location to the app if it does not disrupt service [11]. An adversarial app could monitor the location market and try to associate an anonymous user profile with a particular device. Combined with a profiling attack, it can then send targeted advertisements without compensation by recognizing this device from now on. This is a costly attack

[4] `www.privoxy.org`, `www.mitmproxy.org`

[5] Note this profiling works on *non*-blacklisted locations only.

and can be prevented if OSes separate their advertising services from applications [15] or if the users does not need a permanent ID for this application. Note also that, since UIDs are changed periodically, the profile cannot be updated without paying and hence loses some value over time.

4.2 Attacks on User Privacy

We study the robustness of our solution against a form of attack based on *inference*. We consider a malicious adversary whose goal is to predict the visits to blacklisted locations of a specific user with some accuracy. This may seem a priori impossible since whenever a user visits a blacklisted location, no information about this visit is sent or shared anywhere.

However, because mobility patterns tend to be periodic and similar people may have similar mobility patterns, an adversary may be able to discover something about a specific user's blacklist by comparing their publicly available location information with the full (including blacklisted) location information of 'compromised users'. This auxiliary location information could be obtained via hacking or a malicious or buggy application. Inspired by de-anonymization techniques based on auxiliary information [16], we now pose the following question: "Can an adversary with the full knowledge of the location information of a significant fraction of users predict the blacklisted locations of other users with high accuracy?" We test this on a large dataset of Foursquare checkins. Intuitively, the sparsity of locations and checkins in this dataset allows for strong attacks of this kind.

As in the de-anonymization technique, we consider a similarity score $\mathrm{Sim}(u, v)$ between two users based on common visits. Let L_u denotes the places that are visited at least 1 time by u. We define similarity as:

$$\mathrm{Sim}(u,v) = \sum_{l \in \mathcal{L}} \frac{1}{\mathrm{span}(l)} \mathbb{I}_{l \in L_u \cap L_v} \text{ , for } \mathrm{span}(l) = \sum_{u \in \mathcal{U}} \mathbb{I}_{\{l \in L_u\}} \,.$$

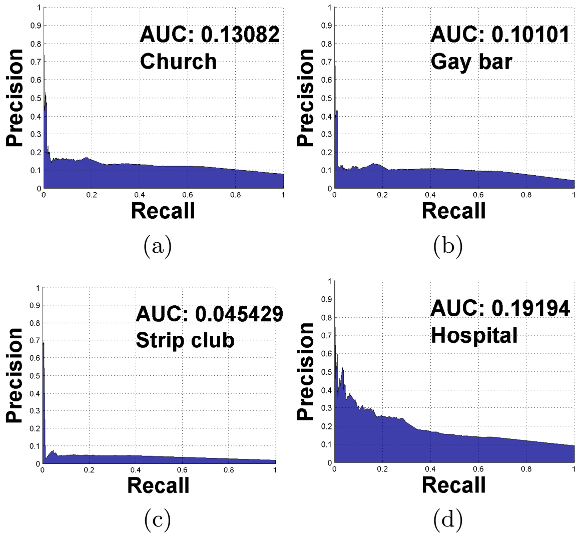
Note that by doing so we weight more the co-occurrence of a rare location as a sign of similarity between two nodes.

The attack then proceeds as follows. For a given keyword k, the attacker looks at all accounts that visited a location tagged with k. For simplicity we will say that such a user visits keyword k. These are the probes used to find similar users who are more likely to behave like them. For a given user u, the adversary first locates the $n = 10$ closest users that are compromised in terms of similarity $v_1, \cdots, v_n$. The attacker then computes the following weighted sum:

$$P(u) = \frac{1}{\sum_{i=1}^{n} \mathrm{Sim}(u, v_i)} \sum_{i=1}^{n} \mathrm{Sim}(u, v_i) \mathbb{I}_{\{v \text{ visits keyword } k\}} \,.$$

It then predicts that u visits locations associated with keyword k if and only if $P(u) \geq \theta$ where $\theta \in [0; 1]$ is a parameter that allows a trade off between accuracy and aggressiveness of the reconstruction technique.

We empirically study the effectiveness of this attack using 1.3 million checkins at 460,663 locations from 40,578 Foursquare users, obtained through crawling publicly available tweets of checkins between March and August 2011. Each Foursquare location is marked with a category, which we assigned to be that location's keyword. In this attack, we consider a severe case where the adversary has compromised 20% of all accounts. We vary the value of θ from 0 to 1 and plot the precision-recall of this attack for various keywords in Fig. 3. As one can see, this attack is rarely effective, even in such extreme case where many user accounts have been compromised. The area under the curve is almost always very small. This turns out to be true even for locations that are sparse, as it is much more difficult to guess right when only a handful of users are visiting a rare location.



(a)

(b)

(c)

(d)

Figure 3: Precision-Recall curves for four sensitive keywords: (a) Church (b) Gay Bar (c) Strip Club (d) Hospital

This points to an interesting difference between inference in our scheme and de-anonymization attacks. While de-anonymization attacks always benefit from sparsity since the data are present in a sanitized form, in our context, the attack does not always benefit from sparsity. This is because a minimum critical mass of typical behavior is needed in order to run inference. This shows that a proper choice of blacklist could potentially protect many locations, even as several accounts are compromised in the system.

4.3 Attacks on Advertiser Revenue

We now consider if advertisers can unfairly lose money to unscrupulous users of the system. Because users are paid when they are accessed by advertisers, they have an incentive to view or click on many ads, even when they are not interested in the displayed products, to artificially boost their profile's value to derive more money from each click. We label these activities "user fraud."

User fraud is a special case of invalid traffic in online advertising. According to Google's Ad Traffic Quality Resource Center, "invalid traffic includes both clicks and impressions ... [that are] not the result of genuine user interest. This covers intentionally fraudulent traffic as well as accidental clicks and other mechanically generated traffic."[6] A request for an ad within our system is just like a request for an ad in the current ad ecosystem, but with some privacy-protecting filtering and potential additional location information. Thus, previous techniques used to identify invalid traffic can be used to identify user fraud. There is a lot of recent research on this topic. Dave et al propose methods to fingerprint click spam [4]. Haddadi uses

[6] `www.google.com/ads/adtrafficquality/index.html`

"bluff ads", ads designed to not appeal to humans and thus only be clicked by bots, to defeat click fraud [10]. Information on the structure of Google's click fraud detection system is available [13]. Beyond academia, multiple startups exist that estimate the rates of click fraud, such as Adometry, Visual IQ, and ClearSaleing (`www.adometry.com`, `www.visualiq.com`, `www.clearsaleing.com`).

Additionally, it is easier to detect user fraud than traditional invalid traffic because location information is more constrained than web-browsing. Users are physically constrained in how far they can travel in a certain period of time and typically display periodic mobility patterns, returning to their homes at night and spending week days at work locations. A more extreme use of physical constraints would be to use location tags; fingerprints extracted from ambient signals at a specific location at a specific time [17]. These constraints can be used to filter out automated attacks on a system. For example, if a user appears to be traveling faster than is physically possible, we can remove them from the system or verify their accounts with a Captcha or phone call. Because of these physical constraints, and because click fraud prevention techniques can easily be applied to our system, we believe that our system is no more vulnerable to gaming than current online advertising. The ongoing viability of online advertising shows that our solution should likewise not be derailed by invalid traffic.

Beyond automated attacks, users might "physically" attack the system by simply going to a high value location in order to appear more valuable to an advertiser than they actually are. Again, techniques to combat click fraud can be employed here. Click fraud techniques must deal with situations in which users actually click links to unfairly gain money, a nice analogy to this form of attack. Beyond this, traveling to a location takes significant time and effort and will likely be too costly to be a viable way of making money.

5. RELATED WORK

Our work is part of a growing body of work that deals with privacy solutions that aim to reconcile the privacy concerns of users with the economic needs of 'free' online web services and mobile applications [8, 9, 18, 19]. Privad [8] and Adnostic [19] are browser based systems that enable behavioral targeting while ensuring users' PII is not leaked to ad-networks performing the targeting. Our focus in this paper is different – we are concerned with location information on mobile devices. Koi [9] is a system developed to address location privacy by way of location matching – applications and service providers pre-declare which locations they would be interested in and the device releases this information at those specified locations. Our solution is different, in that we have an economic component where application developers need to pay to access the user at the specified location. In addition, neither the device nor applications have to be modified to use our solution. Our work is closely related to transaction privacy [18]. The difference is that we focus on location information for mobile devices and develop a keyword-based disclosure scheme.

6. CONCLUSION

The collection and monetization of location information has become a large concern. The main contribution of this paper is the design and analysis of a solution for location privacy using economics. Our solution is simple – opt-in users decide which locations to reveal and only these locations are sold on an information market. Buyers pay to gain access to users at specified locations. Locations are specified in keywords, a notion intuitive to both end users and advertisers. Our solution relies on a privacy protection component that ensures that the location information the user chooses not to release will not be leaked, and also minimizes the linkage of the user's identity with the released information. Future research directions on keyword-based disclosure may include reducing the role of the trusted third party, larger implementations, and a stronger economic analysis of the solution. A few locations, at a cell level, have been shown to provide poor anonymity [5]. An interesting open question is if keywords provide better k-anonymity.

7. REFERENCES

[1] United States v. Jones. *S. Ct.*, 132(10-1259):945, 2012.

[2] J. Bing. Classification of personal information with respect to the sensitivity aspect. *Databanks and Society*, pages 98–150, 1972.

[3] L. Brandimarte et al. Misplaced confidences: Privacy and the control paradox. *WEIS*, 2010.

[4] V. Dave et al. Measuring and fingerprinting click-spam in ad networks. In *ACM SIGCOMM*, 2012.

[5] Y.-A. de Montjoye et al. Unique in the crowd: The privacy bounds of human mobility. *Sci. Rep.*, 3, 2013.

[6] W. Enck et al. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In *USENIX OSDI*, 2010.

[7] A. Franken. Location privacy protection act, 2011. `www.govtrack.us/congress/bills/112/s1223`.

[8] S. Guha et al. Privad: practical privacy in online advertising. In *USENIX NSDI*, 2011.

[9] S. Guha et al. Koi: A Location-Privacy Platform for Smartphone Apps. In *USENIX NSDI*, 2012.

[10] H. Haddadi. Fighting online click-fraud using bluff ads. In *ACM CCR*, pages 22–25, 2010.

[11] P. Hornyack et al. These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. *ACM CCS*, 2011.

[12] P. G. Kelley et al. When are users comfortable sharing locations with advertisers? In *ACM CHI*, 2011.

[13] C. Kintana et al. The goals and challenges of click fraud penetration testing systems. In *ISSRE*, 2009.

[14] B. Krishnamurthy et al. Measuring privacy loss and the impact of privacy protection in web browsing. In *SOUPS*, 2007.

[15] I. Leontiadis et al. Don't kill my ads!: balancing privacy in an ad-supported mobile application market. In *ACM HotMobile*, 2012.

[16] A. Narayanan et al. Robust de-anonymization of large sparse datasets. *IEEE S&P*, 2008.

[17] A. Narayanan et al. Location privacy via private proximity testing. In *NDSS*, 2011.

[18] C. Riederer et al. For sale : Your Data By : You. In *ACM HotNets-X*, 2011.

[19] V. Toubiana et al. Adnostic: Privacy preserving targeted advertising. *NDSS*, 2010.

[20] Wall Street Journal. Apple, google collect user data, 2011. `http://on.wsj.com/gDfmEV`.