Exploiting Innocuous Activity for Correlating Users Across Sites

Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, Renata Teixeira

Exploiting Innocuous Activity for Correlating Users Across Sites

Oana Goga
UPMC Sorbonne Universités
23 Avenue d'Italie
Paris, France
oana.goga@lip6.fr

Howard Lei
ICSI
1947 Center St., Suite 600
Berkeley, USA
hlei@icsi.berkeley.edu

Sree Hari Krishnan Parthasarathi
ICSI
1947 Center St., Suite 600
Berkeley, USA
sparta@icsi.berkeley.edu

Gerald Friedland
ICSI and UC Berkeley
1947 Center St., Suite 600
Berkeley, USA
fractor@icsi.berkeley.edu

Robin Sommer
ICSI and LBNL
1947 Center St., Suite 600
Berkeley, USA
robin@icir.org

Renata Teixeira
CNRS and UPMC Sorbonne Universités
23 Avenue d'Italie
Paris, France
renata.teixeira@lip6.fr

ABSTRACT

We study how potential attackers can identify accounts on different social network sites that all belong to the same user, exploiting only innocuous activity that inherently comes with posted content. We examine three specific features on Yelp, Flickr, and Twitter: the geo-location attached to a user's posts, the timestamp of posts, and the user's writing style as captured by language models. We show that among these three features the location of posts is the most powerful feature to identify accounts that belong to the same user in different sites. When we combine all three features, the accuracy of identifying Twitter accounts that belong to a set of Flickr users is comparable to that of existing attacks that exploit usernames. Our attack can identify 37% more accounts than using usernames when we instead correlate Yelp and Twitter. Our results have significant privacy implications as they present a novel class of attacks that exploit users' tendency to assume that, if they maintain different personas with different names, the accounts cannot be linked together; whereas we show that the posts themselves can provide enough information to correlate the accounts.

Categories and Subject Descriptors

K.4.1 [**Public Policy Issues**]: Privacy; H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: H.3.5 Online Information Services, H.3.4 Systems and Software; G.3 [**PROBABILITY AND STATISTICS**]; D.4.6 [**Security and Protection**]

Keywords

Privacy; Online Social Networks; User Profiles; Location; Language; Geotags; Account Correlation

1. INTRODUCTION

Users of online social network sites increasingly scrutinize privacy protections as they realize the risks that sharing personal content entails. Typically, however, much of the attention focuses on properties pertaining to *individual* sites, such as specific sharing settings Facebook offers or Google's terms of service. What users tend to miss, though, is a broader threat of attackers correlating personal information *across site boundaries*. While on a per-site basis, a user may deem fine what she posts to her Facebook, Twitter, and LinkedIn accounts, she might be revealing much more than she realizes when considering them in *aggregate*. As one example, a social engineering attack could first identify employees of a victim organization on LinkedIn, and then examine their Facebook accounts for personal background to exploit while also following their tweets to understand travel patterns. Indeed, we already see legitimate business models based on such correlation techniques, such as services offering "social media screening" to weed out job applicants (e.g., [1]). Also, modern sales portals combine crowdsourced phone information with social networking posts to present a customer profile to sales representatives and telephone agents in assisting hotline callers more effectively [2].

In this work we set out to advance our understanding of such correlation attacks. In general, it is much harder to defend against cross-site inference than to protect personal information on individual sites where privacy settings directly control what becomes public. As combined data sets can often reveal non-obvious relationships —as prior work on de-anonymization [3] convincingly demonstrates— it remains challenging to assess the correlation threat even for sophisticated users. More fundamentally, as a research community we lack insight into what precisely enables correlation attacks to succeed, along with counter-measures one can take for protection.

To further our understanding, we examine the initial step of any correlation attack: *identifying* accounts on different sites that belong to the same user. In contrast to past work [4], we focus on exploiting implicit features derived from a user's *activity*, rather than leveraging information explicitly provided—and hence more easily controlled—such as name or date of birth. Specifically, we explore matching accounts based on *where*, *when* and *what* a user is posting. As it turns out, combining these three types of features provides attackers with a powerful tool to correlate accounts.

In this paper we examine correlating accounts across Twitter, Flickr, and Yelp; we demonstrate that they provide sufficient public information to link user accounts. We deliberately choose sites where account correlation is unlikely to cause much concern. However, similar techniques apply to more sensitive targets as well, in

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

particular to sites where users expect to remain anonymous such as on dating services, job portals, medical advice forums, and other special-interest sites.

We devise a possible set of attack heuristics, yet we emphasize that our choices are far from exhaustive. We also emphasize that it is unrealistic to expect such attacks to work reliably in a fully automated fashion. Given the vast amount of information online, even small false positive rates would quickly render any fully automated approach infeasible. In that setting, identifying a small candidate set of accounts on other networks is sufficient to allow for manually sifting through for the correct match.

We profile users with three implicit features of their activity: the geo-location attached to a user's posts; the timestamps of a user's posts; and the user's writing style modeled with a probabilistic approach. After discussing our methodology in §2, we first evaluate the potential of each of these three features individually to match user accounts across sites (in §3, §4, and §5, respectively). Then, we evaluate the improvements in accuracy that result from combining all three features (§6). Our results show that, when available, location and timing are powerful for correlating accounts across sites while a user's language model is not as effective. We find that the combination can identify almost as many correlated accounts between Flickr and Twitter as existing attacks that exploit usernames (a much more obvious feature to key on). Moreover, the three features together can identify 37% more correlated accounts between Yelp and Twitter than usernames.

Our work demonstrates a novel *class* of attacks that we believe our community has not yet paid sufficient attention to. The novelty concerns showing that innocuous features of a user's posts can help link accounts across sites. Indeed, it remains the very fact that users want to post content that makes them vulnerable.

2. METHODOLOGY

Our overall goal concerns understanding how user activity on one site can implicitly reveal their identity on other sites. In §2.1, we discuss *features* that we derive from user activity to build activity profiles. In §2.2 we then introduce our basic threat model: an attacker with moderate resources targeting a specific individual or a group of individuals. We discuss the data sets we use for evaluation in §2.3, and the metrics for measuring attack performance in §2.4.

2.1 Features

For our case study, we choose three types of features for building activity profiles that are present on many social network sites: location, timing, and language characteristics.

Location: Many sites provide location information directly in the form of geotags attached to user content, potentially with high accuracy if generated by GPS-enabled devices like mobile phones. However, even without geotags, one can often derive locations implicitly from posted content (e.g., when users review a place on Yelp, that gives us an address). Furthermore, a number of online services map images and textual descriptions to locations or geographic regions (e.g., by identifying landmarks) [5, 6, 7, 8, 9]. For our study, we use the *location profile* of a user, i.e., the list of all locations associated with her posts on a specific social network. The intuition behind that choice is that the combination of locations a user posts from may sufficiently fingerprint an individual across sites.

Timing: Many mobile services and applications such as `Gowalla`, `Foursquare`, and `Instagram` allow users to automatically send content to multiple sites simultaneously. The resulting posts then have almost identical timestamps, which we can exploit to link the corresponding accounts.

Language: The natural language community has demonstrated that users tend to have characteristic writing styles that identify them with high confidence [10]. While these methods typically work best with longer texts, such as blog posts or articles, it is unknown how they perform for short texts such as tweets and how they can contribute to correlation attacks.

2.2 Attacker Model

As our basic threat model, we assume a *targeted individual*: the attacker knows the identity of his victim on one social network, and she wants to find further accounts elsewhere that belong to the same individual. More precisely, for two social network sites SN_1 and SN_2, we assume having one account $a \in SN_1$ and aim to identify account $b \in SN_2$ so that $user(a) = user(b)$.

We assume an attacker with moderate resources–e.g., with access to a small number of computers and the ability to rent further cloud services for a limited period of time. For such an attacker, it is not practical to compare the known account a with *all* accounts of SN_2 as that would require exhaustively crawling the target network. Hence, we assume an attack that proceeds in three steps. First, the attacker pre-filters by selecting a *subset* of accounts for $\widetilde{SN}_2 \subset SN_2$ that will plausibly include b. She then measures the similarity between a and all the $b_i \in \widetilde{SN}_2$ using an appropriate metric $(a, b) \in SN_1 \times \widetilde{SN}_2 \longrightarrow s(a, b) \in \mathbb{R}$. Finally, she selects one account $\hat{b} \in \widetilde{SN}_2$ or a small list of accounts $\hat{b}_j \subset \widetilde{SN}_2$ that are the most similar with a:

$$\hat{b} := \underset{b_i \in \widetilde{SN}_2}{\text{argmax}} \quad s(a, b_i).$$

The attack is successful if $\hat{b}$ equals b or $b \in \hat{b}_j$.

Besides defining an appropriate metric (which we discuss in the following sections), a successful attack requires selecting a candidate set $\widetilde{SN}_2$ so that $b \in \widetilde{SN}_2$ while keeping its size sufficiently small to allow for collecting features from all of the included accounts. The key to that is selecting the accounts in $\widetilde{SN}_2$ based on the features considered. For example, if the attacker aims to link accounts by their location, she may assume that users who post regularly from within a certain region will most likely live there, and thus their posts on other sites will originate there as well. She can then build $\widetilde{SN}_2$ by extracting all users from SN_2 who have posted from that region. Likewise, if she strives to link accounts based on timing, she may select $\widetilde{SN}_2$ as those accounts for which she finds a temporal overlap with posts from $a \in SN_1$. Furthermore, she can also select accounts based on any other information she knows in advance about the user.

This basic threat model assumes that the attacker knows that her victim indeed has a profile on the target network, which may not always be realistic. However, in a variation of the model, an attacker might also target a *group* of people (e.g., a company's employees or users visiting a particular site). Even if some members of the group do not maintain a presence there, others likely will. Thus the attacker will be able to successfully find a part of all the targeted accounts.

2.3 Data Sets

For our case study, we analyze correlation attacks with data collected from three sites: Flickr, Twitter, and Yelp. We choose these sites because of their popularity and because they represent different types of sites: photo sharing, micro-blogging, and service reviewing. We note that many Flickr, Twitter, and Yelp users may *not* necessarily consider account linking across these networks as a compromise of their privacy. In fact, 40% of the Flickr users in

	GT	GT in SF†	SD†	NY†	C†	LA†
Twitter-Flickr	13,629	474	152	427	236	284
Twitter-Yelp	1,889	160	45	106	50	117
Flickr-Yelp	1,199	120	46	81	42	82
Twitter-Flickr-Yelp	559	33	9	25	11	23

Table 1: Number of users in the ground-truth dataset GT (total, and divided into 5 selected areas). † Users with more posts inside a given area than outside it.

	$\widetilde{SN_2}$	$\widetilde{SN_2}$ in SF†	SD†	NY†	C†	LA†
Twitter	232,924	75,747	35,068	89,219	54,774	77,402
Flickr	22,169	6,916	2,305	5,730	4,122	4,113
Yelp	28,976	16,463	4,064	6,239	3,629	9,556

Table 2: Number of users in the $\widetilde{SN_2}$ dataset (total, and divided into 5 selected areas). † Users with at least one post inside a given area; users may belong to multiple areas.

our dataset have an identical username on Twitter. We use them to demonstrate a technique that would also apply to users with different usernames as well as to more sensitive sites, for which the users may care if they were aware of the threat. In the following, we describe the sets of users used for our evaluation and the information we collected about them.

To assess the performance of our attacks, we collect a ground truth set of users for whom we know their accounts on the three sites. We obtain this set by exploiting the "Friend Finder" mechanism present on many social networking sites, including the three we examine. As the Finders often return pages that embed HTML in extensive Javascript, we use browser automation tools (Watir and Selenuim) to extract the results. We give the Friend Finders an existing list of 10 million e-mails[1] and check if the emails correspond to accounts on any of the other sites. Table 1 shows the number of common accounts identified between each pair of sites; this is shown in the GT column. We filtered out all accounts that have no posts or no locations attached (considering addresses for Yelp, and geotags otherwise).

Given the ground truth set, we could evaluate correlation attacks by directly following the attacker model discussed in §2.2: for each ground truth user, we collect corresponding sets $\widetilde{SN_2}$ from a target social network; and our attack would then identify an account $\hat{b} \in \widetilde{SN_2}$ as a likely match. However, this would require us to collect *separate* sets $\widetilde{SN_2}$ for each ground truth user, which is not feasible. Instead, we limit our evaluation to users living in five urban areas in the US (San Francisco, San Diego, New York, Chicago, and Los Angeles), which allows us to use a *single* set $\widetilde{SN_2}$ for all the GT users in each of these areas. We define the subset $GT^{\text{area}} \subset GT$ of users in an area as those who have more posts inside the respective area than outside it. Table 1 shows the number of such users we get in each area.

We also limit the language and timestamp analyses to these subsets so that we can evaluate the combination of multiple features on the same set of users. We note that such a geographical pre-filtering is consistent with the initial stage of our attack model (see §2.2): inferring the region where a victim lives tends to be straight-forward and hence location gives an obvious hint to reducing the size of the candidate set for language- and timing-based matching as well.

[1]This list comes from an earlier study by colleagues analyzing email spam. The local IRB approved collection and usage.

We obtain the corresponding sets $\widetilde{SN_2}^{\text{area}}$ by crawling the three social networks for users from each of the five areas. For Twitter, we use the Streaming API[2] to collect in real-time all the tweets tagged with a location in one of the five areas between August and November of 2012. We then extract all users that have at least one tweet in this collection. We find that 75% of the Twitter GT^{SF} users are included in the set of $\widetilde{SN_2}^{SF}$ users we collected using this approach (a set of users taken from the San Francisco for one year achieves 95% coverage). This confirms our assumption that pre-filtering by location is a realistic approach in narrowing down the set of SN_2 users. Table 2 presents the number of users we collected for each area. In this paper we focus on correlating Yelp or Flickr to Twitter accounts, thus for Flickr and Yelp $\widetilde{SN_2}^{\text{area}}$ datasets we do not strive to get such complete lists of users in each area as we do for Twitter. The Flickr API allows for search of all photos with geotags in a certain region (defined as a latitude/longitude bounding box), which we use to obtain lists of users who posted photos taken in one of the five areas. For Yelp, we retrieve a list of all restaurants in each of the areas and then consider users that reviewed one of them.

For all GT and $\widetilde{SN_2}$ accounts, we download the publicly available profile information from the corresponding social network, including text, timestamps, and location of each posting. For Twitter, we use its API to get all tweets and their attached metadata. Flickr's API likewise provides us the metadata attached to the photos. For Yelp, we again manually crawl and parse the profile pages.

2.4 Evaluation and Performance Metrics

We assess the performance of our attack by checking the power of finding matching accounts using each feature individually §3, §4, §5, followed by the combination of features §6. This allows us to assess the effectiveness of location, timing, and language by themselves, followed by their effectiveness in combination.

For each area, we compute similarity scores $s(a_i, b_i)$, for all $a_i \in GT_1^{\text{area}}$ and $b_i \in \widetilde{SN_2}^{\text{area}}$. To evaluate the effectiveness of a feature in determining whether $user(a_i) = user(b_i)$, we threshold the similarity scores. We measure the *true positive rate* as percentage of accounts above the threshold where $user(a_i) = user(b_i)$, and the *false positive rate* as the percentage of accounts above threshold where $user(a_i) \neq user(b_i)$. As usual, there is a tradeoff between the true positive rate and the false positive rate as lowering the threshold will increase the number of correctly matched accounts but also increase the number of errors. We use receiver operating characteristics (ROC) curves to understand this tradeoff. We focus on thresholds that give results with low false positive rates. Given the large number of users in our data sets (and online in general), even a small false positive rate could render an attack infeasible by returning a large number of false matched accounts. Hence, we typically tune the threshold so that it reports false positive rates of 1%.

We also examine a second performance metric in addition to true/false positive rate. Recall from §2.2 that an attacker chooses one account $\hat{b}$ (or a small list of accounts $\hat{b}_j$) from all $b_i \in \widetilde{SN_2}$ so that it maximizes similarity with $a \in SN_1$. If successful, $\hat{b} = b$ (or $b \in \hat{b}_j$). Our second metric determines the number of accounts in $\widetilde{SN_2}^{\text{area}}$ with similarity scores higher than or equal to $s(a, b)$ (the score of the true matching pair), which we term a user's *rank* for a given attack:

[2]While the Streaming API generally returns only a sample of tweets, limiting a query to a region the size of, e.g., the San Francisco area seems to indeed return the complete set.

$$rank(a,b) := \#\{b_i \in \widetilde{SN}_2 : s(a,b_i) \geq s(a,b)\}$$

$rank(a,b) = 1$ means the matching is perfect and the attacker will pick the right account $\hat{b} = b$ directly. Since a perfect matching is hard to obtain, we typically check if $rank(a,b) \leq m$, i.e., the correct user is amongst the top m matches. For small m, an attacker can inspect that set manually.

3. LOCATION PROFILES

We first examine location information in more detail. Our goal is to understand the degree to which locations attached to user content are sufficiently unique to identify an individual. Matching locations involves two parts, which we discuss in turn: *(i)* representing a user's location profile in the form of a fingerprint suitable for comparison; and *(ii)* defining a similarity measure between two such profiles. For evaluation, we focus on matching accounts from the Yelp and Flickr GT^{area} sets to the Twitter $\widetilde{SN}_2^{\text{area}}$ sets. Based on the results, we also investigate what properties enable correlating users successfully by their location profiles.

3.1 Building Profiles

To motivate the use of locations, we start by examining the degree to which location profiles represented as zip code sets uniquely identify a user. Out of all Twitter accounts from the combined sets $\widetilde{SN}_2^{\text{area}}$ from all the five areas, 91% exhibit *unique* zip code combinations (i.e., no other user posts from the same set of zip codes). Of the remaining 9%, almost all post from only a very small number of zip codes: 74% only post from one, 21% from two, 5% from more than two, and 5 accounts post from more than ten locations. Manually inspecting the latter, we find that three of them appear to belong to a single person maintaining separate personas on Twitter—which, incidentally, means we have just linked related accounts by their location information. For Flickr, 96% have unique zip code sets; out of the remaining 4%, 97% post from only one zip code and 3% from two. For Yelp, 77% have unique zip code sets; out of the 23% non-unique ones, 89% post from one zip code, 8% from two, and 3% from more than two zip codes. These results encourage us to use locations to fingerprint users.

We define a user's *location profile* as a histogram that records how often we observe each location in her posts. The histogram's bins represent "location units", such as zip code, city, coordinates of a longitude/latitude cluster or region[3]. To eliminate the bias of users posting more often on one site than another, we normalize each histogram by the total number of location units in the histogram such that they represent probability distributions.

As location units, we test three different types of choices:

Grids: We map each latitude/longitude geo-coordinate to the cell within a spatial grid that has its center closest to the coordinate. Considering cell sizes ranging from 1x1 km^2 to 12x12 km^2, 10x10 km^2 proves most effective in our experiments.

Administrative Regions: We map each latitude/longitude geo-coordinate to an address using the Bing Maps API [11]. Trying alternative address granularities (streets, zip codes, cities, counties, states), we find zip codes yield the best results.

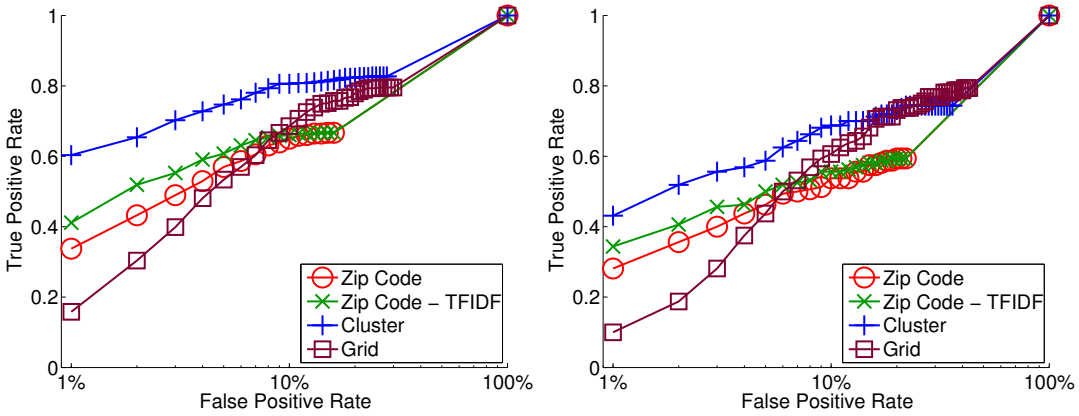
One problem with representing location profiles as normalized histograms of zip codes is that all zip codes contribute the same to the similarity between two accounts. That however is undesirable as some zip codes are much more popular than the others (especially on Yelp, where people go out). Profiles containing those zip codes are therefore likely to have a high similarity even if they do not correspond to matching accounts. To adjust for that effect, we borrow the *term frequency - inverse document frequency (TF-IDF) [12]* weighting scheme from the information retrieval field to weight zip codes proportionally to their popularity. We apply TF-IDF as follows: for each zip code in an account's location profile, TF represents the frequency of the zip code in the location profile, and IDF represents the number of times the zip code appears in other location profiles in $\widetilde{SN}_2^{\text{area}}$. Then the weight of the zip code is $TF/log(IDF)$. With TF-IDF, zip codes that are less common across all profiles but more representative of specific location profiles have higher weights.

Clusters: We use a clustering approach as a more dynamic scheme to group geo-coordinates into regions. Using the k-means algorithm with an Euclidean distance, we group latitude/longitude geo-coordinates from all users in each $\widetilde{SN}_2^{\text{area}}$ into corresponding longitude/latitude clusters. A small cluster represents a popular small area (e.g., blocks of downtown San Francisco), while larger clusters represent bigger, less populous regions (e.g., a park or forest). Our experiments show that using 10,000 total clusters per area produces the best results. We then associate each geo-coordinate with its N closest clusters. We assign weights to each of the N clusters based on a Gaussian distribution, with mean equal to the location, and variance set to 400 (the optimal value according to our experiments). In this approach, the cluster with the centroid closest to the location is assigned the largest weight of 1. The remaining clusters are assigned decreasing weights equal to values along the tail of the Gaussian distribution, according to the distances of their centroids to the geo-coordinate. Using $N > 1$ is better because associating more clusters to each location represents a "soft" assignment. This soft assignment is advantageous in cases where locations of a user's posts in one site is close to, but not exactly the same as, the locations of the same user on a different site. In our experiments N=20 produces the best results. We obtain the final cluster-based location profile histogram for an account by first adding the weights of all clusters associated with all locations of the account, and then normalizing the weight of each location by this total sum of the weights.

Figures 1a and 1b compare the accuracy of using histograms at grid level, zip code level, zip code level weighted with TF-IDF, and cluster level at their best configurations. We use the Cosine distance to measure the similarity between histograms (in the next section, we explore alternative choices). Figure 1a shows the ROC curve for matching Flickr to Twitter for users in San Francisco (the conclusions were similar for other cities), and Figure 1b for Yelp to Twitter. We obtain each ROC curve by varying the similarity score threshold from highest to lowest similarity score values, and computing the true positive rate (TPR) and false positive rate (FPR) when only considering as a match accounts with similarity score above the threshold. These plots take into account all pairs $s(a_i, b_j)$, where $a_i \in GT_1^{SF}$ and $b_j \in \widetilde{SN}_2^{SF}$. The best case would be a vertical line at 0% FPR followed by a horizontal line at 100% TPR; a random classifier would be a diagonal line from 0% TPR and FPR, to 100% TPR and FPR. Note that the plots are in log scale to focus on low false positive rates.

Grids have the lowest TPR in both cases. For a FPR of 1%, grids never achieve TPR higher than 20%. Users from dense populated areas have a greater chance of being confused with one another, when using grids, because the places from where they post tend to be closer to each other, which makes users post from different grids

[3]We also experimented with other fingerprint representations, such as a binary vector just indicating whether a location is present and non-histogram approaches such as matching directly on geo-coordinates, but the histogram approach provided the best results.



(a) Flickr to Twitter

(b) Yelp to Twitter

Figure 1: ROC curves for different location representations for matching Flickr and Yelp users (GT_1^{SF}) to Twitter users ($\widetilde{SN}_2^{SF}$).

less often. In addition, in less populated areas (e.g national parks), grids split places that should be considered the same in different locations, which makes correlated accounts that post from different grids in the same place look less similar.

Zip codes achieve higher TPR than grids, in particular when combined with TF-IDF, because zip codes take into account population density. Clusters achieve the highest TPR for all values of FPR. Their accuracy is significantly better for small FPR, which is the operational point we are interested in. For example, when the FPR is 1%, the TPR for identifying Flickr users in Twitter is 60%. Clusters have higher accuracy because they capture population densities. Furthermore, the soft cluster assignment finds similarities in cases when a user posts from two close by zip codes in her Flickr and her Twitter account. We analyze all ground truth users for which the location profile of their account in Flickr and Twitter had no zip code in common (i.e., they had similarity score equal to zero when using zip codes). Half of these users were indeed posting from neighboring zip codes, and hence had higher similarity scores when using clusters.

While the above considers the complete data sets, we also examine building location profiles individually per time interval: one month, one year, two years, three years, and all available data. Our results show that by aggregating at smaller time intervals, we end up removing too many data points from the profiles, making them less precise. While doing so helps to better identify a few prolific users, it impacts most users negatively.

The clusters made of complete datasets of posts achieve higher TPR than grids and zip codes, in particular for low FPR, thus we use them for the rest of the paper.

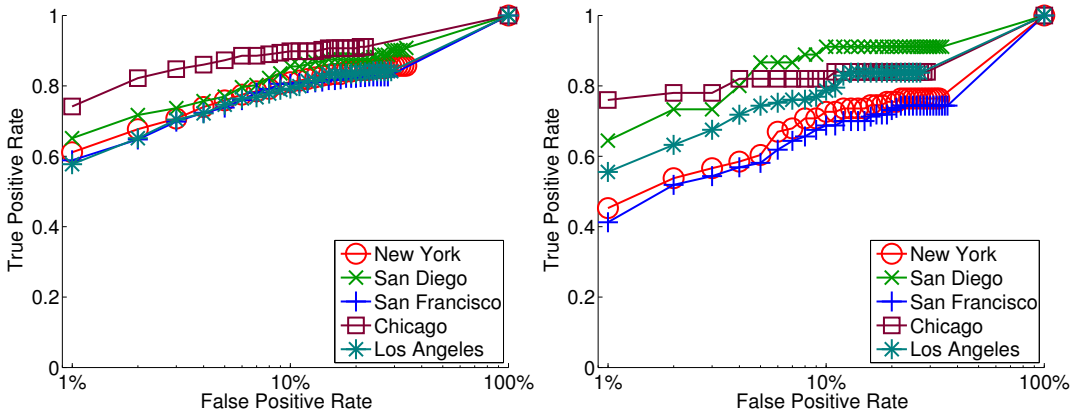
3.2 Similarity Metrics

So far we have used the Cosine distance to compare histogram-based location profiles. Another possibility is to train a classifier and obtain a data-driven function to perform this match; however, the feature space for the classifier is too large and sparse, as we have more than 300,000 features (i.e. clusters). Furthermore it has been shown that if you train a neural network to match two discrete probability distributions using the squared error criterion, it learns to approximate the cosine distance [13]. We now proceed and examine other distance functions to compare the histograms. The statistics literature offers a variety of metrics for measuring similarity between two probability density functions P and Q [14]. We test a series of candidates, including Cosine and Jaccard from the Inner Product family; Euclidean and Manhattan from the Minkowski family; Hellinger from the Squared-chord family and Kullback-Leibler (KL) divergence from the Shannon Entropy family. We skip the details here for brevity but our analysis finds that except for the Euclidean distance others show comparable accuracy (which agrees with the previous mentioned result [13]). The Euclidean distance yields a much lower accuracy because it is sensitive to the absolute difference between two bins, in particular if it is large. In contrast, similarity metrics such as Cosine are sensitive to bins with non-zero values in both profiles, which better suit the matching of location profiles. Since the Cosine, Jaccard, and Hellinger distances have similar TPR in our experiments, we use Cosine for the remainder of our discussion.

3.3 Accuracy

The previous sections show that representing the location of posts with clusters and identifying similar location profiles with the Cosine distance achieve the best tradeoff between TPR and FPR for identifying correlated accounts in Flickr-Twitter and Yelp-Twitter. In this section, we discuss the overall accuracy of using location to identify correlated accounts across sites.

Figure 2a presents the accuracy of matching Flickr to Twitter accounts for each of the five regions we study, whereas Figure 2b presents the same results for Yelp to Twitter (the San Francisco results are the same as the Cluster curves in Figures 1a and 1b). For San Francisco, at 1% FPR, we have 60% TPR to match Flickr and Twitter accounts and 42% TPR for Yelp to Twitter. As a toy example, consider how these numbers apply to a small company of 10 employees, where all of them have Flickr accounts. Assume an attacker aiming to find their respective accounts on Twitter starting from a pre-filtered list of 100 candidate accounts. Among the total of 1,000 (Flickr, Twitter) account pairs, 10 are true matches and 990 are not. With 60% TPR and 1% FPR, our location-based attack will return a set of about 16 (Flickr, Twitter) account pairs that are possibly correlated: 6 true matches (60% of the 10 users) and 10 false matches (1% of the 990 pairs). An attacker will need to sift through these 16 account pairs manually to identify the 6 true matches. Consider now the scenario used in our experiments, where an attacker wishes to identify the Twitter account of one given Flickr user in the San Francisco area using only location information. In this scenario, she has 60% chance of finding the Twitter account



(a) Flickr to Twitter (b) Yelp to Twitter

Figure 2: ROC curves for different urban areas for matching Flickr and Yelp users to Twitter users using clusters.

associated with the Flickr account by manually investigating the 750 most similar Twitter accounts, instead of searching all of the 75,474 San Francisco Twitter accounts.

Figures 2a and 2b show that, although the shape of the ROC curves are similar across areas, the accuracy of the attack based on location is even higher for other areas than San Francisco. Our analysis of correlated accounts in each area shows that most differences come from the fact that some areas such as San Francisco and New York have more users whose posts in Flickr or Yelp have no location in common with posts in Twitter. This observation is especially true when matching accounts from Yelp to Twitter. In San Francisco and New York, many people work and live away from the neighborhoods in the city center, where people often go out. If a given user mainly tweets during her daily activities and not when she is out in restaurants, the location of her tweets will have little overlap with the places of restaurants she reviews on Yelp.

The comparison of Figures 2a and 2b shows that the accuracy of matching accounts from Flickr to Twitter is higher than from Yelp to Twitter. This difference comes, most likely, from the nature of these sites. Users of Flickr and Twitter have more unique location profiles, because they can post or take a picture from anywhere, whereas Yelp reviews come from a large, but fixed set of locations (which correspond to the address of the reviewed restaurants). Indeed, §3 showed that only 77% of Yelp profiles are unique as opposed to 96% unique profiles for Flickr. Moreover, Flickr users tend to post from more locations. Finally, Flickr posts have more common locations with the corresponding Twitter account than Yelp posts do.

3.4 Implications

We now study the implications of these results for users. In particular, which properties of a user's location profiles can help prevent the attacker from successfully correlating her accounts. Although the location profile is a powerful feature for correlating accounts of a single user across sites, the results in Figures 2a and 2b show that we cannot identify all of our ground truth users with low FPR. We use the rank metric (defined in §2.4) to split users in three groups according to the difficulty for an attacker to identify the correct account from the set of candidate accounts. We define as *vulnerable* the set of users with rank smaller 750; *medium vulnerable* users with rank between 750 and 7,500; and *protected* as users with rank higher than 7,500. We check how many locations users in each of these groups post from, and how the number of common locations between the location profiles of the two correlated accounts affects the rank. To investigate this we use all true account matches from Flickr to Twitter and Yelp to Twitter in the San Francisco area.

We find that protected users generally post from fewer locations, only 36% of protected users post from more than five locations, whereas 70% of vulnerable users post from more than five locations. Moreover, 95% of protected users have no common locations between the location profiles of their accounts across sites; whereas all vulnerable and all medium vulnerable users have at least one location in common. These results suggest that one approach to protect against this attack is to minimize the number of common locations across sites. In fact, there is an 80% probability of the correlated Twitter account to have a rank lower than 750 given a Flickr account when the two accounts have posts in three common locations (this probability is 69% to have rank lower than 375 and 47% lower than 50). If the user in the two accounts posted from more than six locations in common, then these probabilities increase to 85%, 76%, and 58% respectively.

Thus, the number of common locations across sites is the most important property that makes users vulnerable to the account correlation and even posting from a few common locations can already be enough to identify a small set of candidate correlated accounts.

4. TIMING PROFILES

Many third-party applications, in particular on mobile devices, allow users to automatically send updates to different sites simultaneously. For example, when Instagram uploads to Flickr, it can automatically tweet a pointer to the photo. We exploit this behavior to correlate accounts based on the timestamps of such automated posts.

In this section, we focus on Flickr and Twitter datasets because Yelp only gives the date, and not the exact time of each post. Generally, we aim to find accounts where one or more timestamps of Flickr photos match the timestamps of tweets. However, even for simultaneous posts, timestamps may differ slightly due to processing delays and desynchronized clocks. Hence, we consider a time small window around each timestamp to declare that the timestamp of the photo and that of the tweet match. The question is what an appropriate window size is. If the window is too small, we might

miss true post matches, whereas a larger window may report many false matches.

To answer that question, we investigate the timestamp differences we see in our ground truth set, considering all the GT Twitter - Flickr pairs. For each account pair (a, b), $a \in Twitter$, $b \in Flickr$, where $user(a) = user(b)$, if the list of timestamps of posts in a is $tstmps(a) = \{t_1, t_2, t_3\}$ and in b is $tstmps(b) = \{T_1, T_2, T_3\}$ and $t_1 < t_2 < T_1 < t_3 < T_2 < T_3$, then we define the set of timestamp differences as the set of differences between timestamps of two consecutive posts on different sites $td(a, b) = \{T_1 - t_2, t_3 - T_1, T_2 - t_3\}$. This set contains all the timestamp differences between posts on the two sites potentially corresponding to the same content (e.g., a photo on Flickr and its link on Twitter). Note that in this example, if T_2 represents a Flickr image post, and t_3 the automated Tweet for the image post, then $T_2 - t_3$ represents the delay resulting from desynchronized clocks between Flickr and Twitter.

We investigate what is an appropriate threshold for this delay between posts across sites so that we detect automated posts with low false positives. We manually investigate the content of posts with timestamp differences smaller than 30 s, as we consider 30 s a safe upper bound for the maximum delay between automated posts. We can differentiate automated posts from the others as they have similar texts, and the metadata attached to tweets contains the name of the application that generated it. We find that most posts with timestamp differences larger than 5 s are not automated. We thus investigate TPR of applying thresholds ranging between 1 s to 5 s to match accounts.

We define the timestamp similarity score $s(a, b)$ between accounts a and b as the number of timestamp differences in $td(a, b)$ that are lower than a given delay threshold. We experimented with normalizing this value by the size of $td(a, b)$, but it did not improve the matching quality. We set FPR to 1% and measure TPR for thresholds ranging from 1 s (which includes all timestamp differences between 0 s and 1 s) to 5 s for accounts in Flickr GT_1^{SF} to accounts in Twitter SN_2^{SF}. The 1 s threshold has the highest TPR (13%) while 5 s has the lowest TPR (12%). Hence, we use a 1 s threshold to correlate accounts based on timing.

The reason TPR is never higher than 13% is because only few users in our datasets use automated posts. When users do use automated posts, however, we often find a perfect match. In our dataset, all the users with more than four timestamp matches have a rank of one. This means that even if users only use automated posting for a brief period or just to test them, an attacker can correlate their accounts with a very high precision. As applications such as Instagram and Foursquare become more popular, we expect the accuracy of matching by timing to increase.

5. LANGUAGE PROFILES

The final type of feature we consider for correlating accounts is textual data. This approach builds on existing work that demonstrate that free-form text can exhibit characteristics sufficiently unique to identify an author [15]. To explore this potential, we examine correlating Yelp reviews and Flickr photo descriptions with Twitter posts. We do not explore exact text matches because these are usually automated, and we capture these cases with timing matching.

For each Yelp account we consider the joint set of all the reviews; for each Flickr account we consider all the descriptions, tags and titles attached to a photo; and for each Twitter account we consider all the tweets with the exception of re-tweets and tweets that share links (as the text represents the title of the article in the link and not something that the user wrote). In the GT_1^{SF} and SN_2^{SF} datasets, we find an average of 546 distinct words per Twitter account, 730 per Yelp account, and 516 per Flickr account. Note that these words may contain punctuation, and are case-sensitive. If we remove punctuation and disregard case, we have 394 distinct words per Twitter account, 218 per Yelp account and 480 per Flickr account.

There are tens of millions of distinct words found in the posts of the three social network accounts, and many do not appear across all three accounts (only 200,000 of the roughly 40 million case-sensitive words in Twitter, along with punctuation, appear in Yelp and Flickr). Hence, it is important to first apply a pre-filter to reduce the number of words for several reasons: *(i)* to reduce the total number of words to a computationally-manageable size, *(ii)* to remove words that do not appear across multiple accounts, which would not significantly affect user account correlation, but could de-emphasize the words that do significantly affect the correlation, *(iii)* to remove common words (i.e. “and", “the") that may not be user-discriminative, and *(iv)* to account for case-sensitivity and punctuation. We recognize, however, that certain users may prefer certain combinations of case and punctuation in their writing style, potentially making case and punctuation user-discriminative features. After removing words that do not appear in both Yelp and Twitter, or Flickr and Twitter, we conduct two investigations based on the aforementioned points. First, we investigate the effects of punctuation and case-sensitivity of words. Second, we investigate the effect of removing the most frequent words between Yelp and Twitter, and Flickr and Twitter. The pre-filtering approach of removing punctuation and case-sensitivity, along with the top 1,000 most frequent words, gives the optimal results.

We build probabilistic language models for each Twitter user by constructing histograms of word unigrams, and normalizing them by the total word count per user such that each histogram represents a unigram probability distribution. We choose word unigrams as the unit for our models because our experiments show no further improvements when broadening to higher n-grams (i.e., multi-words). The reason why higher n-grams and other stylometry methods are less effective is because *(i)* the pre-filtering already removes what often links words together, and *(ii)* tweets consist mostly of keywords with fewer stylistic expressions. To measure the similarity between the Yelp and Twitter or Flickr and Twitter accounts, we accumulate the probabilities of each word in the Yelp or Flickr text from the language model of the Twitter account. This approach is a general version of the approach implemented by Stolcke [16].

In general, the language-based results are significantly worse compared to the location-based results, and achieve only a 6% TPR at 1% FPR for matching Yelp to Twitter accounts, and 10% for matching Flickr to Twitter accounts. The small TPR from Yelp to Twitter likely comes from the fact that the same user may adopt drastically different kinds of textual structure when writing Yelp reviews (typically complete paragraphs using words mostly found in the English lexicon) versus when tweeting (typically short sentences with fewer standard words). Correlating accounts from Flickr to Twitter is better than from Yelp to Twitter possibly because the short description of the photos may be more similar in style and topic to tweets than reviews.

6. COMBINING FEATURES

The previous sections discuss matching accounts across sites with one *individual* feature at a time (location, timing, or language). We now use all three features *simultaneously*. The premise here is that combining the individual metrics should *(i)* achieve stronger correlation by leveraging their respective strengths, while *(ii)* making it harder for users to defend against such attacks. We then compare

the results obtained by combining the three features with existing attacks that exploit usernames to match accounts.

6.1 Method

To assess the performance of combining multiple features to identify accounts that belong to one user across sites, we use a binary logistic regression classifier [17], a popular technique for predicting dependent variables that lie within a finite range of values (which is the case of our similarity scores that range from 0 to 1). For a pair of accounts in different sites, the classifier takes as input the similarity scores of each feature (using the best settings for each feature as discussed in §3, §4, and §5) and predicts whether the pair of accounts is a match (i.e., belong to the same user) or not, as well as the probability of a match. We build classifiers for different combinations of features. For matching Yelp to Twitter, we build three classifiers using location and language (one classifier using location alone, another using language alone, and a third combing these two features). For Flickr to Twitter, we build six classifiers with different combinations of location, language, and timing.

We build our training and test sets from a dataset with all pairs of accounts in GT_1^{area} and $\widetilde{SN}_2^{area}$. As a result, we obtain an imbalanced training and test sets with fewer cases of account pairs that are true matches (only $|GT_1^{area}|$) and significantly more account pairs that are not matches ($|\widetilde{SN}_2^{area}| \times |GT_1^{area}| - |GT_1^{area}|$). This imbalance is representative of real-world datasets (where we expect the number of true matches to be orders of magnitude smaller than the total possible account pairs between two sites). We then evaluate the accuracy of each classifier using 10-fold cross validation.

6.2 Accuracy

We compare the accuracy of classifiers using different combinations of features. We only present results for users in the San Francisco area, but the conclusions are similar for other areas. Table 3 presents the classification accuracy of each classifier for matching accounts from Flickr to Twitter and from Yelp to Twitter. This table also includes results for usernames for discussion in §6.3. The table presents the average TPR corresponding to 1% FPR across the ten runs of cross validation as well as the 95% confidence interval computed with vertical averaging [18].

Table 3: Comparison of the TPR for different classifiers at 1% FPR for matching Flickr and Yelp accounts to Twitter.

	TPR at 1%FPR	
Feature	**Flickr-Twitter**	**Yelp-Twitter**
Timing (T)	13±3%	-
Language (Lang)	10±3%	6±3%
Location (Loc)	60±6%	44±6%
Username (U)	77±3%	7±4%
Loc, Lang	60±6%	42±6%
Loc, T	70±3%	-
Loc, Lang, T	63±5%	-
Loc, U	86±2%	44±6%
Loc, Lang, U	86±2%	44±7%
Loc, T, Lang, U	88±2%	-

TPR for classifiers based on individual features–location, timing, and language–are practically the same as the results in §3, §4, and §5, respectively. The small differences come from the fact that here we present results from the 10-fold cross validation, whereas earlier sections simply computed TPR for the entire dataset. The comparison between Loc and (Loc, Lang) when matching Flickr and Yelp with Twitter shows that language doesn't improve TPR when combined with location (in fact, it seems to reduce TPR slightly when matching Yelp to Twitter accounts). Hence, at low FPR, language doesn't help to identify more correlated accounts than the ones location already identifies. We note that when we consider a higher than 10% FPR, adding language to location can increase the TPR by 10% for Flickr to Twitter matching. Timing, however, is more powerful than language. When we combine timing with location TPR improves by 7% over location alone. This increase shows that, when present, timing can very precisely identify true matches which helps improve the TPR especially for low FPR. The combination of location, language, and timing increases the TPR over the entire range of FPR. Timing improves TPR when FPR is low, whereas language helps when FPR is high. At 1% FPR, the highest TPR we achieve for matching Flickr accounts to Twitter is 70% when we combine location and timing. The highest TPR for matching Yelp accounts to Twitter is 44% when using location alone. With the best combination, for the Flickr to Twitter matchings, 17% of the ground truth users can be identified in the top 10, 27% in the top 50 and 33% in the top 100, while for the Yelp to Twitter matchings, 1% can be identified in the top 10, 4% in the top 50 and 7% in the top 100.

6.3 Comparison with username matching

This section compares the accuracy of our classifiers, which only use features extracted from innocuous user activities, with the state-of-the-art technique to match accounts across sites: matching based on the username. We compute the similarity between two usernames using the Jaro distance [19], which is the state-of-the-art distance in record linkage to measure the similarity between two names. Perito et. al [4] showed that the Jaro distance performs well to match usernames across different sites as well.

Table 3 also shows the average TPR at 1% FPR for matching accounts from Flickr to Twitter and from Yelp to Twitter based on usernames. We first note that usernames alone achieve 77% TPR for matching accounts from Flickr to Twitter. When matching Yelp accounts to Twitter, however, usernames only reach 6% TPR, which is lower than any of the other features we consider. Usernames achieve high accuracy to match accounts in Flickr to Twitter, because many users often use the same or similar usernames on these two sites. On the contrary, Yelp users often select as usernames just their first name and the initial of their last name or some alias, reflecting their desire to maintain their reviews pseudo-anonymous.

When we compare matching based on usernames with the combination of location, timing, and language for matching Flickr to Twitter accounts we observe that the TPR of usernames is higher than that of the combination of the three other features together. If we combine usernames with the other three features, we obtain even better results (TPR increases to 88%). Username is clearly a powerful feature to match Flickr and Twitter accounts today. We should not forget, however, that it is easy for users who want to hide to obfuscate their identity by simply selecting different usernames. So, the accuracy of usernames to match accounts across sites can decrease drastically as soon as users realize that correlating information across sites represents a real threat to their privacy. In the Flickr to Twitter dataset, we already find 11% of users that cannot be matched with usernames, but can be matched using location.

Given that users in Yelp select usernames that do not reveal their identity, when matching Yelp to Twitter accounts, location alone achieves a much higher TPR than usernames (44% vs. 7% TPR for 1% FPR). Username does not even help increase TPR when combined with location (see Table 3). In fact, out of all detected matches between Yelp and Twitter 78% are only identified by loca-

tion. Our approach of using features based on innocuous user activity should always work better than usernames for sites like Yelp, where users do not use their true identity.

7. DISCUSSION

Our results in Sections 6.2 and 6.3 demonstrate the power of our attack model, which provides a high match quality even when tuned for the low false positive rates that such needle-in-a-haystack challenges require. We now discuss our results further in terms of realistic attack models, availability of the features we exploit, and potential defenses users may take.

Attack Model: Given our matching accuracy we see two attack models as particularly relevant. First, our correlation scheme allows an attacker to find further accounts that belong to a specific target individual by quickly winnowing down from a large initial starting set to a much smaller number of candidate accounts suitable for manual inspection. While she may still need to invest non-trivial effort into the final verification step, the automatic prescreening nevertheless enables an attack that would not be feasible at all otherwise.

Second, it is possible to attack a group of people rather than a specific individual. An example here is finding employees of a large company that might be vulnerable to bribery (maybe because of gambling habit that indicates money problems) or extortion (maybe because of a medical condition, or an affair). In such a model, the attacker would start with the set of company employees, e.g., on LinkedIn; correlate them with other social networks, and potentially further public records, to collect more personal information; and eventually link all that to relevant target sites such medical forums, addiction advise networks, or dating sites.

Feature Availability: Most social networks provide the features our attacks exploit. For example, Facebook posts carry timestamp information, and Facebook check-ins come with location information. Likewise, both Google+ posts and Youtube videos make the upload time available, and either can include location in its metadata. However, even if an attacker does not have direct access to some of the features on a particular network, often she might still infer it from the posted content itself. For example, with LinkedIn we could get a suitable location profile from the places somebody has previously worked. More interestingly, the multimedia community is developing a range of approaches to accurately determine location information from content, such a photos, videos, and metatags [5, 6, 7, 8, 20, 9]. Currently, only 1% of all the tweets have geotags and only 5% of the active Twitter users have at least one post geotagged. Since our results show that we only need coarse-grain location information to correlate users' accounts, we believe that these techniques can be reliably used in our attack to infer the location of posts when geotags are not available.

One can also collect the necessary features outside of social networks. A particular privacy threat concerns mobile applications with access to a user's current location. If that information is provided back to the application developers (as is typical, for instance, for map and search services), they can identify users by associating corresponding location profiles with social identities. As we have seen, even coarse locations, like zip codes, convey sufficient information, and hence simple privacy-conscious schemes, such as blurring the resolution, will not protect from such attacks.

While we discuss just three specific features for account matching, there are others that an attacker can exploit in a similar way. In particular, content may indirectly reveal further personal information that can help guide the matching process, such as "Happy Birthday" greetings from friends that reveal a person's birthday, even if she does note make the date itself publicly available. Another possibility concerns matching based on interests as inferred from the context and the content one "likes".

Defenses: As we indicated earlier, it remains hard to defend against de-anonymization attacks that exploit information so intrinsically, and ubiquitously, linked with content. However, there are some countermeasures that can make such attacks less likely to succeed, in particular, from the perspective of an individual who is part of a larger group of potential victims[4].

As a possible defense against our timing matching, applications could slightly delay automated posts, introducing random jitter that makes it harder to find suitable thresholds separating them from manually issued content. Our analysis suggest that a variation interval in the order of 10 s of seconds would prove more than sufficient. We suggest two strategies to avoid becoming vulnerable to location matching. As the more obvious one, it clearly helps not to post to separate sites from the same location because that's what the attack keys on (remember from §3.4 that 95% of protected users do not have any common location between their profiles). A more interesting, and less drastic, countermeasure exploits the fact that one can *correct past mistakes* (i.e., already sharing many locations between accounts) by adding further unrelated locations to the mix. Doing so effectively blurs the link to other networks by adding noise. For example, for a vulnerable user (with a rank less than 750, see §3.4), that has 5 common locations between his accounts, to become medium vulnerable or protected he needs to add respectively around two or seven unrelated locations on one social network.

Finally, we note that defending against account correlation generally gets more difficult as the attacker combines further features, hence making the analysis more robust against noise in any individual feature. There is a fundamental tradeoff here in that *any* useful information that a user publishes will potentially increase the chance of a successful correlation attack.

8. RELATED WORK

A variety of efforts have examined aspects of information leakage related to our work, however none of it exploits implicit activity features attached to the content. Most closely related is a recent series of work aimed at identifying users across different sites, similar in spirit to what we discuss yet with different approaches. Perito et al. [4] explored linking user profiles by looking at the entropy of their usernames. Irani et al. [21] studied finding further accounts of a user by applying a set of simple heuristics to its name. Balduzzi et al. [22] correlate accounts on different social networks by exploiting the friend finder mechanism with a list of 10 million email addresses. Most sites have since severely limited the number of e-mail addresses that one can query. Moreover, we need to have prior knowledge of one's email address to use this method. Iofciu et al. [23] used tags to identify users across social tagging systems such as Delicious, StumbleUpon and Flickr. The authors of [24] show that group memberships present on many social networks can uniquely identify users; they leverage this to identify users visiting malicious web sites by matching their browser history against groups on social sites. Zang et al. [25] did a large scale study of the k-anonymity of the top locations from where users are making phone calls and found that at zip code level the top three locations are almost uniquely identifiable, however they did not further explore how these locations would correlate with social identities.

In another line of work, researchers used publicly available information from a social network site to infer specifics about its users, without however correlating it with further accounts else-

[4]"I don't need to outrun the bear; I just need to outrun *you*."

where. Hecht et al. [26] derived user locations from tweets using basic machine learning techniques that associated tweets with geotagged articles on Wikipedia. Similarly, Kinsella et al. [8] leveraged tweets with geotags to build language models for specific places; they found that their model can predict country, state, and city with similar performance as IP geolocation, and zip code with much higher accuracy. Crandall et al. [7] located Flickr photos by identifying landmarks via visual, temporal and textual features. Chaabane et al. [27] leverage interests and likes on Facebook to infer otherwise hidden information about users, such as gender, relationship status, and age. Further similar works exploit the social network graph to infer such information [28, 29].

Language models have been used for data de-anonymization. For example, Nanavati et al. [10] used language distribution at the n-gram level to de-anonymize reviews in an anonymous review process. Two other recent studies show that text posted on blogs can be de-anonymized [30] and that community reviews could be linked across different sites [31].

More generally, a number of de-anonymization efforts demonstrated the power of correlation. Sweeney [32] de-anonymized medical records with the help of external auxiliary information. Likewise, Narayanan et al. de-anonymized Netflix movie ratings [3]. A similar approach attacks a social network graph by correlating it with known identities on another [33]. Srivatsa et al. explored how mobility traces can be de-anonymized by correlating their contact graph with the graph of a social network [34]. On a more fundamental level, Bishop et al. [35] discuss the need to consider external knowledge when sanitizing a data set.

Finally, in our previous work we presented the more general threat of correlating data available on the internet [20] and we investigated the privacy implications of geotagging [36], showing cases where using online data and services can help launch real-world attacks (cybercasing).

9. CONCLUSION

In this work we present a powerful set of techniques for correlating user accounts across sites, based on otherwise innocuous information like location and timing patterns. Our approaches work independent of standard privacy measures, such as disabling tracking cookies or using anonymizing proxies. For our study, we collected data from the three social networks Twitter, Flickr and Yelp, including extensive ground truth of 13,629 users with accounts on both Twitter and Flickr and 1,889 users with accounts on both Twitter and Yelp. Our results go beyond prior work by not relying on more obvious, user-chosen information (e.g., usernames [4]) and by evaluating the power of the correlation in real-world scenarios. We show for example that, using the location information, we can correlate 60% of Flickr accounts with their corresponding Twitter accounts, while only introducing a small percentage of falsely correlated accounts. Moreover, our results show that we only need coarse-grained location information to link a relevant number of accounts. Combining all features together gives comparable results with matching on usernames for Flickr to Twitter correlation, and can identify 37% more correlated accounts for Yelp to Twitter correlation.

The privacy implications of our results are two-fold. First, we point out that it is the *aggregate* set of a user's complete online footprint that needs protection, not just content on individual sites. Second, we find that it is hard to defend against such attacks as the information that enables them often comes intrinsically with the very activity one *wants* to publish.

While our work examines a specific set of web sites and correlation techniques as case studies, it demonstrates the broader potential, and risk, of cross-site correlation. Our approaches remain conceptually simple, yet we expect that soon more sophisticated variants will emerge for exploiting the increasing volume of innocuous user information that web sites now offer via convenient APIs. In particular, we anticipate that automated content analysis technology—such as face recognizers and natural language processing—will enable correlations more powerful than what we demonstrate here. As such, we see our contribution less in the specific performance numbers that our experiments yield—which will always vary between users, features, and sites—but primarily in pointing out that identifying users by their posting activity indeed poses a real threat. From a research perspective, we encourage our community to devise novel privacy protections that take such threats into account and, where hard to prevent, at least support users in understanding their vulnerability.

Acknowledgements

This work was supported by the National Science Foundation under grant CNS-1065240, and by the Agence National de la Recherche grant C'MON and was carried out at LINCS (`www.lincs.fr`) and ICSI (`www.icsi.berkeley.edu`). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors or originators and do not necessarily reflect the views of the NSF or ANR. We would like to thank Patrick Loiseau for his valuable feedback during all stages of the project, as well as the anonymous reviewers for their helpful comments.

10. REFERENCES

[1] Social Intelligence Corp., http://www.socialintel.com/.

[2] R. Schmid, "Salesforce service cloud – featuring activision," September 2012, http://www.youtube.com/watch?v=eT6iHEdnKQ4&feature=relmfu.

[3] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy (S&P)*, 2008.

[4] D. Perito, C. Castelluccia, M. Ali Kâafar, and P. Manils, "How unique and traceable are usernames?" in *Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS)*, 2011.

[5] "Yahoo! placemaker," http://developer.yahoo.com/geo/placemaker/.

[6] "geonames.org," http://geonames.org.

[7] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg, "Mapping the world's photos," in *Proceedings of the 18th International Conference on World Wide Web (WWW)*, 2009.

[8] S. Kinsella, V. Murdock, and N. O'Hare, ""I'm eating a sandwich in Glasgow": modeling locations with tweets," in *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents (SMUC)*, 2011.

[9] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, 2010.

[10] M. Nanavati, N. Taylor, W. Aiello, and A. Warfield, "Herbert west: deanonymizer," in *Proceedings of the 6th USENIX Conference on Hot topics in Security (HotSec)*, 2011.

[11] "Bing Maps API," http://www.microsoft.com/maps/developers/web.aspx.

[12] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, pp. 11–21, 1972.

[13] B. Picart, "Improved Phone Posterior Estimation Through K-NN and MLP-Based Similarity," Idiap Research Institute, Tech. Rep., 2009.

[14] S.-h. Cha, "Comprehensive survey on distance / similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[15] V. Keselj, F. Peng, N. Cercone, and C. Thomas, "N-gram-based author profiles for authorship attribution," in *Pacific Association for Computational Linguistics*, 2003.

[16] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *Proceedings of Int'l Conference on Spoken Language Processing*, 2002.

[17] M. Tranmer and M. Elliot, "Binary logistic regression," *Cathie Marsh for Census and Survey Research, Paper 2008-20*.

[18] F. J. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," in *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, 1998.

[19] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *Proceedings of IJCAI-03 Workshop on Information Integration*, 2003.

[20] G. Friedland, G. Maier, R. Sommer, and N. Weaver, "Sherlock Holmes' evil twin: on the impact of global inference for online privacy," in *Proceedings of the 2011 Workshop on New Security Paradigms Workshop (NSPW)*, 2011.

[21] D. Irani, S. Webb, K. Li, and C. Pu, "Large online social footprints–an emerging threat," in *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03 (CSE)*, 2009.

[22] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel, "Abusing social networks for automated user profiling," in *Proceedings of 13th International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2010.

[23] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, "Identifying users across social tagging systems," in *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[24] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *Proceedings of the 31st IEEE Symposium on Security and Privacy (S&P)*, 2010.

[25] H. Zang and J. Bolot, "Anonymization of location data does not work: a large-scale measurement study," in *Proceedings of the 17th annual International Conference on Mobile Computing and Networking (MobiCom)*, 2011.

[26] B. Hecht, L. Hong, B. Suh, and E. H. Chi, "Tweets from justin bieber's heart: the dynamics of the location field in user profiles," in *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI)*, 2011.

[27] A. Chaabane, G. Acs, and M. A. Kaafar, "You are what you like! information leakage through users' interests," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS)*, 2012.

[28] E. Zheleva and L. Getoor, "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles," in *Proceedings of the 18th International Conference on World Wide Web (WWW)*, 2009.

[29] D. Gayo Avello, "All liaisons are dangerous when all your friends are known to us," in *Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia (HT)*, 2011.

[30] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale author identification," in *Proceedings of the 33st IEEE Symposium on Security and Privacy (S&P)*, 2012.

[31] M. A. Mishari and G. Tsudik, "Exploring linkability of user reviews," in *Proceedings of the 17th European Symposium on Research in Computer Security (ESORICS)*, 2012.

[32] L. Sweeney, "Weaving technology and policy together to maintain confidentiality," *Journal of Law, Medicine, and Ethics*, vol. 25, no. 2–3, pp. 98–110, 1997.

[33] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy (S&P)*, 2009.

[34] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2012.

[35] M. Bishop, J. Cummins, S. Peisert, A. Singh, B. Bhumiratana, D. Agarwal, D. Frincke, and M. Hogarth, "Relationships and data sanitization: A study in scarlet," in *Proceedings of the 2010 Workshop on New Security Paradigms (NSPW)*, 2010.

[36] G. Friedland and R. Sommer, "Cybercasing the Joint: On the Privacy Implications of Geo-Tagging," in *Proceedings of the 5th USENIX Conference on Hot Topics in Security (HotSec)*, 2010.