Social Forensics: Searching for Needles in Digital Haystacks

Iasonas Polakis*, Panagiotis Ilia†, Zacharias Tzermias†, Sotiris Ioannidis†, Paraskevi Fragopoulou†

* Columbia University
New York, NY, USA
polakis@cs.columbia.edu

† FORTH
Heraklion, Greece
{pilia, tzermias, sotiris, fragopou}@ics.forth.gr

***Abstract*—The use of online social networks and other digital communication services has become a prevalent activity of everyday life. As such, users' *social footprints* contain a massive amount of data, including exchanged messages, location information and photographic coverage of events. While digital forensics has been evolving for several years with a focus on recovering and investigating data from digital devices, *social forensics* is a relatively new field. Nonetheless, law enforcement agencies have realized the significance of employing online user data for solving criminal investigations. However, collecting and analyzing massive amounts of data scattered across multiple services is a challenging task.**

In this paper, we present our modular framework designed for assisting forensic investigators in all aspects of these procedures. The data collection modules extract the data from a user's social network profiles and communication services, by taking advantage of stored credentials and session cookies. Next, the correlation modules employ various techniques for mapping user profiles from different services to the same user. The visualization component, specifically designed for handling data representing activities and interactions in online social networks, provides dynamic "viewpoints" of varying granularity for analyzing data and identifying important pieces of information. We conduct a case study to demonstrate the effectiveness of our system and find that our automated correlation process achieves significant coverage of users across services.

I. Introduction

As the popularity and use of online social networks (OSNs) has increased to the point of becoming a social norm, these services have also become platforms for conducting nefarious activities as well as exhibiting offensive behavior (e.g., cyberbullying). As such, even malicious individuals (not only cybercriminals, but perpetrators of physical crimes) have adopted these technologies. The explosive growth of OSNs has, in a sense, created the first *digital generation* consisting of people of all ages and backgrounds. People are creating their digital counterparts for interacting with others, for both recreational and professional reasons, and disclose a vast amount of data in an attempt to fully utilize these services.

This behavior has raised the concern of the research community in terms of user privacy and the wide range of threats users expose themselves to, ranging from identity theft to monetary loss. While the amount of personal information disclosed by users [1] or leaked by services [2] is troubling, in certain cases it can have a positive "side-effect". Law enforcement agencies have been able to solve criminal cases after extracting the digital footprints of users, as they contained clues that ultimately led to the discovery of the perpetrators.

Ideally, users will learn to be more privacy-aware, and limit the visibility scope of their personal information to a well-defined set of friends [3]. In such a scenario, when agencies *lawfully*[1] acquire a suspect's device they will still be able to extract useful data from the accounts. Accordingly, our goal is twofold; first and foremost, to create a system that can assist digital investigators in collecting and analyzing user data from OSNs and services, which can also be employed by researchers conducting experiments in social networks. Second, to increase user awareness regarding privacy threats, like how seemingly unrelated accounts from different services can be associated and traced back to them even if under different names, or the ways that disjoint types of information can be correlated.

Social forensics tools aim to facilitate the discovery of this digital "trail of breadcrumbs", and extract data that can guide criminal investigations towards uncovering crucial information. Even though a multitude of digital forensics tools exist, they mostly focus on recovering deleted files or information from the device's volatile memory, and little research has been done regarding social forensics. In this paper we present a framework that demonstrates the effectiveness and feasibility of an automated and extensive toolset for assisting forensics analysts and researchers in this daunting task.

We have designed and implemented our modular framework with the following usage model in mind: the authorities seize the digital devices (be it desktop, laptop or just hard drives) of someone suspected for a crime[2] and wish to acquire all the information regarding its online activities. Social forensics analysis presents three major challenges: (i) acquiring as much data as possible from the suspect's online accounts and relevant local artifacts, (ii) correlating contacts across services, and (iii) visualizing this extensive collection of data. Our modular framework handles all three tasks.

The core functionality of any forensics analysis tool is the extraction of user data. We create a series of modules, each

[1] We are referring to the acquisition of warrants through the proper channels, and not unlawful mass user surveillance.

[2] For the remainder of this paper, we will refer to this person as the *suspect* for reasons of simplicity.

designed for extracting data from a specific service. When available, we take advantage of public APIs. In the remaining cases, we build custom crawlers for acquiring the data.

The correlation of users across services is a very crucial, and challenging, aspect of social forensics. Our correlation component follows a series of techniques for mapping user accounts from different services. First, we extract the email addresses of the suspect's contacts from Facebook as well as all the other communication services (e.g., Gmail). We then use various social networks as oracles for mapping emails to profiles. Furthermore, we employ data from a social directory site where users create a profile page with links to their social accounts, to further improve our correlation results. Finally, we also use fuzzy matching techniques for matching user names and email handles, and guessing missing information for different services.

The datasets collected during the data extraction process contain a wide range of different types of information regarding online activities. Existing visualization tools for social networks usually focus on the depiction of graph-related data. Nonetheless, various visualization libraries exist, and can handle multiple types of data. As such, we build upon existing libraries and create a visualization framework designed specifically for visualizing data representing user activities in OSNs and communication services. Furthermore, the massive amount of data necessitates the creation of dynamic viewpoints of varying granularity, that facilitate surveying aggregated statistics, as well as focusing on specific users or interactions.

Finally, while our framework is built for facilitating lawful procedures, these techniques can also be used with malicious intent. Personalized attacks, are an increasing threat for users and corporations (e.g., cyber-espionage [4]), but are believed to be inherently limited and small-scale, as they rely on manual processes. However, our experimental results demonstrate that the large-scale, automated, collection and correlation of user data for personalized attacks is feasible. The effectiveness of our framework also highlights the privacy risks that users face.

Overall, the main contributions of this work are:

- We create an extensive framework for crawling a wide range of popular social and communication services. We employ a series of techniques for automating the process of correlating accounts belonging to the same user across different services. Our visualization framework provides perspectives of varying granularity, and association of disjoint types of activities, for efficient analysis of large collections of social networking data.
- We perform a minimal case study that shows the efficiency of our crawling approach and the effectiveness of our correlation process. Our results demonstrate how disjoint sets of information from multiple services can be associated.
- Our experimental results serve as a cautionary tale for social networks and their defenses: automated, large-scale, *personalized attacks* by cyber-criminals are feasible.

II. Social Forensics

Digital forensics analysis has been a valuable asset in solving crimes in spite of its relatively young "age". Initially, the focus was on analyzing data stored on a computer, and recovering files that suspects had erased. However, as a result of the advances of technology and its use propagating to all aspects of life, nowadays, digital devices contain only a fraction of the available user data that could assist the authorities in solving crimes.

A lot of interaction takes place in online social networking services and over digital communication media such as emails, instant messaging and VoIP networks. Users access information through these devices and save entries about their appointments in digital calendars. As such, a large part of the data is saved online and not on a specific device. Thus, it is mandatory for forensics tools to extract data saved online, and not only extract data stored locally on a device. The goal of social forensics is to target social networking and communication services and extract as much information as possible, regarding the online activities and communications of a suspect. We will also demonstrate that digital services can be leveraged to provide associations of user profiles across services, i.e., identify profiles form different services that belong to the same user.

Multiple reports describe cases where the authorities have resorted to social networks for acquiring information, which has ultimately led to cases being solved (e.g., [5]). Even murder cases have been solved with the use of clues extracted from the suspect's digital communication and online activities [6], [7]. A survey held in 2012, among 600 law enforcement agencies from 48 states in the USA, reported that 92.4% of the agencies surveyed online social services [8]. For 77.1% this was done as part of criminal investigations. This survey reflects the significance of the data available in online services for assisting authorities in solving crimes.

III. Design and Implementation

The core of our framework has been implemented in `Python` as a collection of components. We have designed it in a modular way so it can easily be extended by adding new modules for other social networks and services. In this section, we provide a high-level overview of our system, describe the role of each component, and present technical details regarding the implementation of some of the components we have created. Figure 1 presents the architecture of our framework and the steps that comprise the whole procedure:

1) Stored session cookies and user credentials are used to log into the online services as the suspect.
2) Each crawling component extracts as much data possible from each service that the user has an account for.
3) All extracted data is saved into a MySQL database.
4) The account correlator component:
   a) Pulls the account information of the suspect's contacts from the database.
   b) Uses several techniques to correlate the accounts.

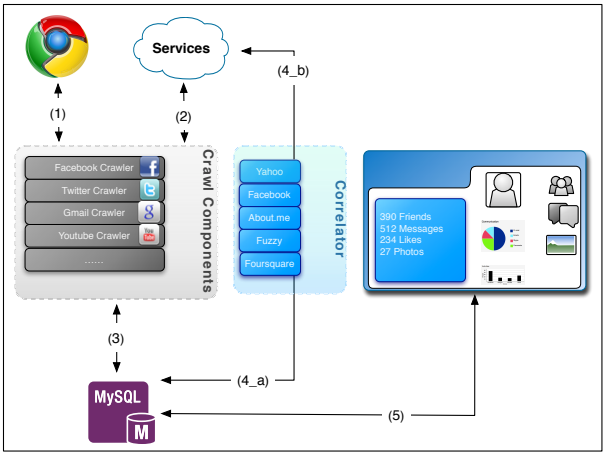

Fig. 1. The architecture of our framework which is comprised of three major components.

5) The data visualization component fetches data from the database asynchronously and dynamically presents the viewpoints requested.

Usage Scenario. We implemented our framework with the following usage scenario in mind. A *forensics analysis investigator* has acquired the suspect's digital device (or hard drive and connected it to a computer) and connected it to the Internet, since the data extraction and correlation components must connect to online services.

Even though we have developed our system as a social forensics framework, it also has another use. One of the most important subjects in news headlines recently, has been the revelation that the NSA [9] (and, potentially, other government agencies) monitors popular social networking sites and other digital communication services. Furthermore, cyber-criminals regularly employ user data found in OSNs for deploying personalized attacks. As such, the result of our case study in Section VI can also help raise users' awareness regarding the feasibility of collecting their data on a large scale, and how correlating disjoint online accounts can result in privacy leakage. The Immersion project [10] similarly explored privacy issues by visualizing email data shared by users.

Automation. An important aspect of such a system, is to fully automate execution. In our current implementation, minimal manual intervention is needed, for authenticating the crawling modules that use public APIs with the services through `OAuth`. This will also be automated in the future.

After the authorization phase, everything else is completed automatically. The framework installs a MYSQL database and creates a series of tables for storing all the information from the suspect's accounts. The libraries required by the crawling component, for example `fbconsole` [11] and Tweepy [12] are downloaded and installed automatically. The libraries for the visualization component are included within the web application.

A. Data collection components

Depending on the targeted service, the corresponding crawling component attempts to extract as much information as possible. In the case of online social services we leverage existing public APIs, if available. Otherwise we create custom crawlers for extracting the data. Here we provide technical details for certain modules.

Log-in process. Our tool uses the credentials saved in the browser's password manager or existing session cookies, to log into the targeted services as the suspect. The password managers of Chrome and Firefox utilize a `SQLite` database as their password manager back-end. The most popular browsers, like Firefox and Chrome, retain this database encrypted using a "master password". However, both browsers allow a user to easily extract manually all the stored credentials and passwords through their user interface. In such a case, all the passwords are displayed in a human readable format. In this work we consider that the investigator extracts the stored credentials and passwords, and imports them into our framework's configuration file, for being used by the forensics tool for logging into social network and communication services. Also, some existing tools (i.e., [13]), can extract all the user passwords directly from the browser's database, and decrypt them. In order to fully automate the password extraction and login process, and thus avoiding human intervention, we plan to extend our framework into this direction, by employing similar techniques. Furthermore, upon retrieving a password, we can test it against the other services as well. Previous work [14] has reported that up to 51% of users reuse the same password across multiple sites.

Furthermore, browser session cookies are also stored in `SQLite` databases found locally in the filesystem. Our framework locates these databases, with regards to the particular browser and operating system, and extracts the stored session cookies. Then, our tool uses the extracted session cookies for trying to log into the targeted services.

Facebook. Once logged in, a custom application is installed in the suspect's profile, so the data can be retrieved through Facebook's Graph API. This application has access to all resources available in the profile. After installation, our system leverages the Facebook Query Language (`FQL`) to extract the data from the user profiles [15]. `FQL` provides an `SQL`-like interface for querying user data, and can evaluate multiple queries in a single API call through `FQL` multi-query requests. Queries are packed as a `JSON`-encoded dictionary and sent as a single request. The response includes a similar dictionary with the respective results.

Twitter. An application that has full access to the profile data has to be installed in the suspect's profile. Twitter poses an extra overhead during the crawling phase, due to its rate-limiting policy. Requests are performed with 10-second intervals, for avoiding potential rate-limiting issues. Protected accounts (whose information is only available to followers) are collected with the highest priority. Next, we focus on accounts with small volumes of data.

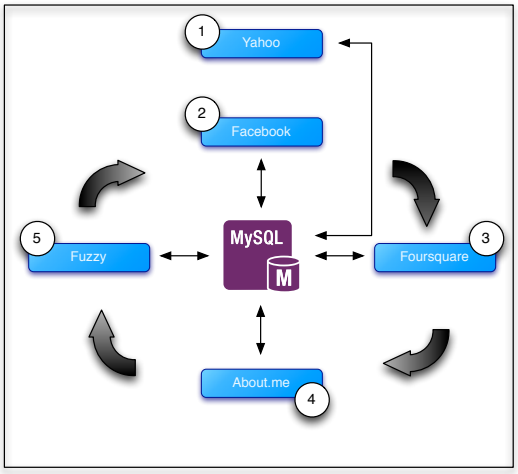

Fig. 2. The account correlation process.

Google+. We utilize the official Google Plus API for extracting the data. Through this API and the `OAuth` authentication method, we extract all the public information from the users' profiles, and the contacts from public circles. The information also includes the name of the city where the user resides. We rely on the Google Geocoding API [16] for converting the city to a pair of geographical coordinates.

Foursquare. Our crawling component is built upon a `Python` wrapper [17] for the Foursquare API. After the `OAuth` authentication is completed and an authorization token is acquired, the crawler extracts the data through API calls that return the data formatted as `JSON` objects.

B. Account correlation component

As already mentioned, our framework visits the online profiles of the suspect in a multitude of services, and tries to collect as much data as possible. This data usually contains information about the contacts of the suspect, his relationships and communication interactions. Each collected profile that belongs to a contact in the social circle of the suspect has its own profile attributes and published data, such as email addresses, age, gender, locations etc.

Each collected contact account does not necessarily correspond to a unique user. Users usually keep multiple accounts in different online services, and thus, more than a single account from the suspect's circle may correspond to the same user. Therefore, investigators can be severely misguided in their investigations, if the aforementioned aspect is not taken into consideration by the forensics tool.

The account correlation component of our tool is of crucial importance as it performs the challenging task of identifying user accounts across different services and mapping them to users' online identities. The correlation component of our framework consists of several modules, where each distinct module leverages a different service or technique. An overview of the correlation process is presented in Figure 2.

At the beginning of the correlation process our tool executes the Yahoo module. This module is executed only once, as it does not have any dependencies, and its output is not affected by the outcome of the other modules. Thereafter, the process executes the remaining modules in a round-robin fashion, as the outcome of each module may affect the results produced by the other modules. Thus, the correlation process executes these modules iteratively until none of those produce new information within a full iteration.

Yahoo. This module is employed for extracting the email addresses of the suspect's Facebook contacts. In general, even though Facebook applications provide a plethora of information about a user's contacts (Facebook friends), they are not allowed to obtain and extract email addresses [18]. This restriction can be bypassed by leveraging a suspect's Yahoo mail account, as it allows one to export Facebook contacts, and include them to the contact book of the Yahoo mail account. This process provides the profile names and the email addresses used by the suspect's friends for setting up their Facebook accounts. This process does not return all the used email addresses, as users are able to restrict the visibility of their email address by changing the default account settings. In a small study with 10 users (suspects), we found that $\sim$70% of the total number of their contacts kept the default setting and their email addresses could be exported.

Facebook. In some cases, as stated, the Yahoo component returns the Facebook account of the suspect's contacts, but not their corresponding email addresses. [19], [20] demonstrated how Facebook can be leveraged as an oracle for mapping a user's email address to his profile. Thus, our correlation component uses this technique for mapping email addresses to Facebook accounts that did not yield results in the Yahoo module. Again, users can specify that they wish to be removed from such searches by changing their privacy settings, but in reality, this functionality is enabled by default and average users do not disable it. This module also creates synthetic email addresses by using certain variations of the given user name (e.g., "`john_doe`", "`john.doe`", "`doe_john`") along with the most common email providers (e.g., "`hotmail.com`", "`gmail.com`", "`yahoo.com`") in an attempt to guess the user's email address.

Foursquare. The official API provides the functionality of searching for Foursquare accounts based on different types of information and can, thus, also be used as an oracle for correlating user accounts. Specifically, the API call takes specific information as a parameter and returns the respective Foursquare account, if such an account does exist. The information that can be passed as parameters to the API call are the user's Facebook ID, Twitter handle, email address, name and phone number. Thus, apart from locating a user's Foursquare account, we can also associate disjoint pieces of information we have collected from other services.

About.me. This online service offers a platform for the users that allows them to create their personal website. This website contains users' personal information and links to their accounts on popular social networking services. Using the

names and usernames extracted in previous steps, we search for about.me profiles that match and we extract the links to profiles on social services. Then we attempt to verify that these accounts belong to the respective user by comparing the account IDs to the identifiers we have correlated previously.

At first, we leverage the website's search functionality for locating the contacts of the suspect, that have an about.me profile. As the search query results are dynamically rendered through Ajax requests, we scrape the results through PhantomJS [21], a headless webkit that also offers a Javascript API. After obtaining the existing user profiles, we extract all the available links pointing towards user's social network profiles.

Fuzzy matching. Some of the services we extract data from don't provide the email addresses of the account's contacts, which would allow us to deterministically correlate user accounts across services. Also, different email addresses may have been used for creating accounts in different services. To overcome this, we compare information collected from different services and match them based on similarity. While this method follows a "fuzzy" approach, we are able to obtain results, as users tend to reuse user names across services, or simple variations of them. For example, a user with a Facebook profile under the name "John Doe" might have an email address handle "john_doe", "johndoe80" etc.

C. Visualization components

Our goal is to develop a visualization platform that offers a wide variety of graphic data representations, while remaining portable. This led us to create it as a web application. The front-end is designed to run on the same machine where the data is stored. The vast amount of data mandates the use of an asynchronous, event-driven model for the front-end, where data is fetched upon request. The front-end is built upon AJAX requests using the jQuery framework [22] for data retrieval and manipulation.

The ever-growing need for complex data visualization has lead to the release of powerful frameworks. D3.js [23] is a JavaScript visualization library capable of rendering a variety of schematics such as Graph layouts and Calendar Views among others. This framework is used for the majority of visualizations incorporated in the front-end. Moreover, we leverage the Google Maps JavaScript API [24] to render location-based information, when available, on a map.

IV. Data Collection

In this section we provide a list of the services from which we collect user data, as well as a description of the types of information acquired. For every online social network, we also collect any information that is reachable for every one of the suspect's contacts. Table I presents an overview of the data collected in each case.

Facebook. This is the main source of information, as it is the most popular online social network, and users tend to reveal a large amount of personal information on it. Our crawling component extracts any of the following data and related metadata that exists: the users information (including locations, education, work), list of contacts and their accessible data, chat logs, status updates, wall posts and comments, photos, videos, check-ins, likes, shares, pages, events and groups, notifications.

Twitter. We first collect the account's information and contact list. That includes the accounts the suspect follows as well as those following the suspect. We also collect the suspect's tweets and any tweets re-tweeted, and all available metadata (e.g. timestamps, location).

Foursquare. We collect the suspect's check-ins along with the corresponding metadata. Specifically, we collect the timestamp, the venue's name, VenueID, and location coordinates. We also collect the list of friends, and any links to their profiles on other networks. Unfortunately, due to limits set by the API and website, we can only retrieve the last 100 check-ins of the suspect's friends.

Skype. We first collect the list of contacts and their disclosed information (which may include location, gender, date of birth). Then we extract the history of chat logs and relevant metadata, call history (and duration) and file exchanges. We also attempt to retrieve any exchanged files that are still located on the hard drive.

Gmail. We collect all emails exchanged with the suspect, and extract the email addresses and names associated with them. For each email we also collect the relevant metadata.

Google. We access the suspect's account and extract the relevant information from Google calendar and Google Docs. We collect all calendar entries (which may contain a location, a description, and other users attending), and download accessible documents and metadata about which other contacts have access to the documents).

Google+. We first collect the suspect's contacts contained in the various "circles" (i.e. contact groups), and the suspect's activities; posts, comments, shares, and "+1"s (similar to likes in Facebook). We extract publicly available data from the accounts of the contacts, as well as any accounts that have commented on the suspect's profile (even if they are not part of one of his circles).

Youtube. We first collect the suspect's information. Then we extract the history of watched videos, and channel subscriptions, playlists, uploaded videos and their comments and favorite videos.

Dropbox. We first locate the Dropbox folder, depending on the suspect's operating system, by retrieving the information from the application data. Then, by traversing the Dropbox directory tree, we extract all the files with their corresponding metadata. We also keep the application data that can be used for other aspects of forensic analysis [25].

A. Mobile devices

The use of smartphones has become part of everyday life. Consequently, a plethora of useful information regarding a person's online activities and communication can be extracted from such devices. Modern smartphones' offer Internet connectivity on-the-go, allowing users to access digital services

TABLE I
TYPE OF DATA COLLECTED FROM EACH SERVICE.

Data	Facebook	Twitter	Foursquare	Skype	Gmail	Google	G+	Youtube	Dropbox
Name	✓	✓	✓	✓	✓	✓	✓	✓	✗
Username	✓	✓	✗	✓	✗	✗	✓	✓	✗
Email	✗	✗	✓	✗	✓	✗	✗	✗	✗
Birthday	✓	✗	✗	✓	✗	✗	✓	✗	✗
Sex	✓	✗	✗	✓	✗	✗	✓	✗	✗
Location/Places	✓	✓	✓	✓	✗	✓	✓	✓	✗
Geo-Locations	✓	✓	✓	✗	✗	✗	✓	✗	✗
School/Education	✓	✗	✗	✗	✗	✗	✓	✗	✗
Work/Position	✓	✗	✗	✗	✗	✗	✓	✗	✗
Phone Number	✗	✗	✗	✓	✗	✗	✗	✗	✗
Contacts/Friends	✓	✓	✓	✓	✓	✓	✓	✓	✗
Photos/Videos	✓	✓	✗	✗	✗	✗	✓	✓	✗
Tags/Descriptions	✓	✓	✓	✗	✗	✓	✓	✓	✗
Chat/Messages	✓	✓	✗	✓	✓	✗	✗	✗	✗
Calls & Duration	✗	✗	✗	✓	✗	✗	✗	✗	✗
Posts/Comments	✓	✓	✓	✗	✗	✗	✓	✓	✗
Dates/Timestamps	✓	✓	✓	✓	✓	✓	✓	✗	✗
Likes/Shares/RTs	✓	✓	✗	✗	✗	✗	✓	✓	✗
Groups/Pages	✓	✗	✗	✗	✗	✗	✓	✓	✗
Links	✓	✗	✓	✗	✗	✗	✓	✓	✗
Files & Metadata	✗	✗	✗	✓	✗	✓	✗	✗	✓

from any location. As a result, these devices contain a vast collection of data, ranging from e-mails and browsing history to chat logs and contact information, that could be extremely useful to analysts through the course of an investigation. As part of our toolset we have implemented a data extracting module for Android, as it is currently reported as the most widespread smartphone operating system [26].

Components. Our data extraction module consists of two components; An ADB [27] script that pulls raw databases from the device and an Android app which extracts and uploads crucial information of the suspect to our framework.

Device access and permissions. As a court can compel the suspect to unlock the mobile device [28], we consider the analyst will be able to gain access to the device. For an optimal data extraction process, the device should be rooted, otherwise certain database files cannot be accessed. However, this is not an obstacle, as the device can be rooted without loosing any data (some devices might require a data backup during the process). Nonetheless, root access is not required for the module to operate.

Data collection. The data extraction process is automated and integrated with the other components of the toolset, and all extracted data is added to the database and handled by the correlation and visualization modules. The component pulls the following data:

- Contacts: Name, phone number, email, address, instant messenger names and further contact details.
- Call Logs: The full history of calls is collected with the corresponding metadata (type, timestamp, etc).
- SMS History: messages received by any application are collected, with the corresponding metadata. We also handle two popular apps: Viber and WhatsApp.

Account Correlation. Mobile devices allow users to combine user profiles from multiple services within a single contact on the device. Using information from the extracted contacts, we can further improve the results of the correlation process presented in Section III-B.

Location data. Geo-location data logged on smart phones can provide forensic investigators with interesting information regarding the whereabouts of a person at specific times. However, newer versions of Android do not maintain a database file containing a history of the device's GPS readings.

Interpreting the GPS data to compute a location is handled centrally by the Android OS, and an interface allows apps to query the location without having to process the data that might originate from GPS satellites, cell towers, or Wi-Fi networks. Furthermore, the kernel doesn't use a cache to store recent locations but a bound system service: an IPC mechanism for different processes to receive data from the service that actually computes the location. As such, the application decides to store the received location data or not. An extensive study of the most popular apps is required to determine which apps save geo-locational data by default. Overall, the most useful piece of location information readily available, is the search history of the Google Maps app, that contains the suspect's location queries.

(a) User granularity (b) Service granularity

Fig. 3. Aggregated statistics perspective: details regarding the most interesting activities, at a user-granularity and service-level granularity.

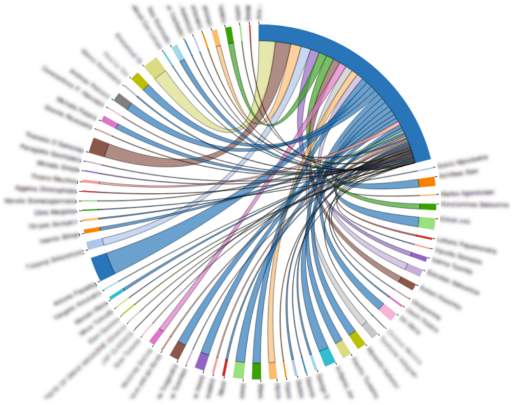

Fig. 4. Overall communication between the suspect and contacts. The volume of communication determines the width of the connection.

V. Activity Visualization

In this section we describe the various methods for visualizing our collected data. The plethora of services that can be used by suspects require a grouping of this disjoint information into a unified set, where actions across services are correlated (e.g. what type of communication does the suspect have with user X from all services). Furthermore, the abundance of available information necessitates the ability to shift focus to specific activities (e.g., status updates on Facebook), and interactions (e.g., users with the largest amount of shared activities with the suspect). Thus, we provide the analyst with dynamic "perspectives" of varying granularity, with aggregated correlations as well as fine-grained views of the collected data. We have several viewpoints for creating the different perspectives.

Aggregated. We present aggregated statistics regarding the most interesting activities from all the services. With one glance, the analyst can see which services the suspect mainly uses, and what data is available. Figures 3(a) and 3(b) depict the aggregated statistics presented in this viewpoint. Specifically, we can see the most important types of data across services and a more detailed description of activities per service, respectively.

Service. We focus on a specific service, and present aggregated statistics regarding the user's activities. A list presents the contacts with the most communication with the suspect, and a graph depicts the structure of the social graph and the interconnections between all contacts. Node size is based on the number of connections the contact has. Thus, the analyst can immediately recognize heavily connected users or outliers. The graph can plot contacts of a specific service or a combined view of all services where the contact's of each service have a common color. Each node represents a user and, when clicked, presents the contact's name and photo. Also, a search function dynamically detects and highlights nodes in the graph, allowing investigators to quickly identify contacts of interest. In Figure 4 we present a screenshot of a graph that visualizes the total communication between suspect and online contacts. The amount of shared activity defines the width of the connector. This enables the users with the most communication to be easily identified and scrutinized. When the connector is clicked, a window presents all the shared activities.

User. A very important viewpoint is that which focuses on a specific user. Once the analyst has identified online contacts that might be of interest, he can use one of two perspectives. First, one can select the contact and be redirected to an aggregated statistics viewpoint, containing all information available regarding the actions of that contact across all services (based on the number of accounts that have been associated during the correlation phase). Second, the analyst can choose to focus only on the shared activities the contact has with the suspect across all services. That includes, chat messages, emails, wall posts, shared photos, etc. The coarse-grained perspective presents aggregated statistics, while the fine-grained perspective allows to focus on a specific type of activity. In both viewpoints, the investigator can ultimately view all individual activity and communication resources, e.g., exchanged messages, pages "liked", or Skype calls. Furthermore, the viewpoints can be dynamically configured to visualize data from one or all services.

Timestamp. Time is an important factor for visualizing relevant data. Every perspective contains a color-coded calendar depicting the amount of activity a user has conducted on a specific date, which allows a fine grained overview of a specific period. A histogram presents an overview of the activities. The analyst might wish to focus on the activities of the suspect during a specific time period which is of interest, or a specific service. As such, certain viewpoints can dynamically change and focus on a specific time window.

Content. A word cloud provides a quick view of the most common words contained in the suspect's communication, which can be across services or focused on a specific service or user. Thus, recurring topics can easily be spotted. In the case of Twitter, we also create a word cloud with the hashtags (topics) of the suspect's tweets. This can reveal subjects that the suspect tends to follow or comment on (e.g., politics, religion)

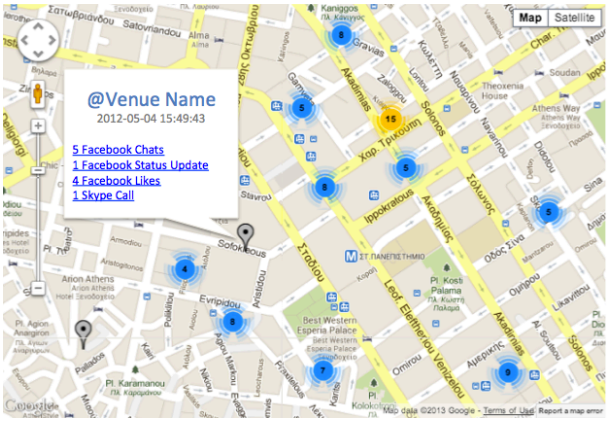

Fig. 5. An aggregated city-level view, with details of a specific check-in and the associated activities.

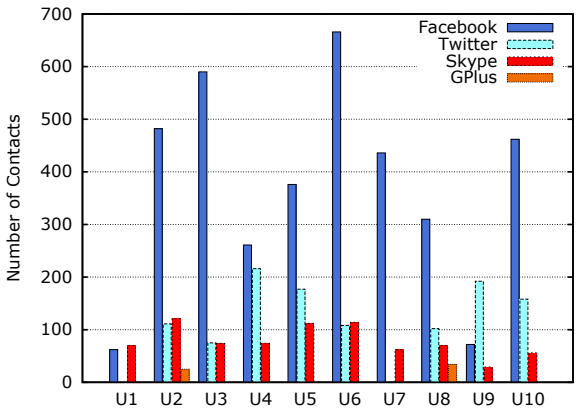

Fig. 6. Number of contacts per service.

and are relevant to the analyst's investigation. Clicking on one of the terms will fetch all the messages, emails, tweets or posts containing the term.

Furthermore, we also follow a more targeted approach, by employing the list of keywords that the US Department of Homeland Security searches for in social networks [29]. Specifically, we search for 377 keywords that belong to 9 categories, ranging from terrorism to drug-related incidents. Occurrences are broken down per-category and per-service. By clicking on the respectful information, the analyst is presented with the resources that contain the keywords.

Location. An important piece of information is the suspect's location. Using information from the suspect's check-ins and residence we plot a map with the locations he has visited, and also visually annotate the amount of times each location has been visited. Furthermore, the analyst can also define a time window, within which all of the suspect's activities are correlated with that location. For example, with a time window of one hour, by clicking on the location marker, a window will inform of all the activities (e.g. chat, Skype calls) the suspect conducted up to one hour after the check-in. Thus, the analyst can associate important activities to specific locations or even search for patterns of activities at certain locations. Figure 5 depicts a closer view of a specific region, with the information window for a specific check-in. The window presents the name of the venue, the check-in timestamp and a series of activities that have been completed within a one-hour time window. All elements are click-able for presenting the resources of interest.

Furthermore, as a specific period might be of interest, we can plot the check-ins conducted during a specific time-period. Also, the investigator can select a contact, a distance X and a time duration T, and the map presents check-ins that the suspect and the contact conducted with a time difference up to T at venues that have a max distance of X.

Photographs. Photos found in social networks can be valuable in criminal investigations, as demonstrated in the case of the Vancouver riots [5] where vandals were identified through photos posted in social networks. The investigator can select to view all the photos collected from the suspect's profiles. Any available user tag information is also presented, and statistics show the contacts with the most common photos with the suspect. Previous work has proposed fine-grained access control mechanisms [30] that could potentially impact the effectiveness of this stage, however, such mechanisms have not been deployed in practice.

VI. Evaluation - Case Study

We have conducted a small user study, with participants from our lab, for evaluating the performance and effectiveness of our forensics tool. As expected, the number of users that took part in the study is small (10), as users tend to be wary of explicitly disclosing personal information. It is quite challenging to find volunteers that are willing to give access to their social networking and email accounts, even if it is made clear that all the collected data will only be automatically processed and then deleted. Because of the intrusive nature of the framework, we do not fully employ our toolkit in the first experiment, but only try to get insight on the properties of users' social graphs for evaluating our data collection process. Then, we conduct an in depth experiment with the accounts of one of the authors.

For the first study, we collect information regarding user connections, and the public information included in the profiles of users' contacts, in 4 popular social networks and messaging services, namely Facebook, Twitter, Google+ and Skype. While previous work on social forensics has focused solely on Facebook, we find that other services are also quite popular and can provide an abundance of useful information. The exact number of contacts each user has, is given in Figure 6. Table II shows the percentage of users that actively use each service, the average number of contacts per service and the time required for collecting their information.

It can be observed that all participants are users of Facebook and Skype, while 90% also have an active Twitter account. Moreover, all participants have a Gmail account but, interestingly, only 2 of them are active in Google+. We also found that users tend to have a much smaller set of contacts in other

TABLE II
PERCENTAGE OF USERS HAVING A PROFILE IN EACH SERVICE, AVERAGE SOCIAL GRAPH SIZE (#CONTACTS) AND AVERAGE TIME (*sec.*) FOR CRAWLING SOCIAL GRAPH AND CONTACTS' INFORMATION.

NETWORK	FACEBOOK	TWITTER	SKYPE	G+
Users	100%	90%	100%	20%
Contacts	371	142	78	29
Crawling Time	185.5	75	1.61	68.3

TABLE III
TOTAL TIME (*sec.*) SPENT FOR COLLECTING ALL THE ACTIVITIES FROM THE SUSPECT'S ACCOUNTS, AGGREGATED NUMBER OF ACTIVITIES (DATA) COLLECTED FROM EACH SERVICE, AND THEIR SIZE IN MB.

NETWORK	FACEBOOK	TWITTER	SKYPE	G+
Activities	16,397	27,333	20,061	57
Size	3.86	4.56	6.54	0.14
Time	402.43	1025.39	403.82	68.30

services when compared to Facebook, with almost 80% less contacts on Skype. This can be attributed to the "general-purpose" nature of Facebook, while all the other services reflect stronger relationships between the users, and not just online acquaintances (especially Skype). Thus, we consider that forensics tools should also leverage these services as they are potential sources of significant data.

The average time spent for collecting users' social graphs, and their contacts' profile information, for each service, is given in Table II. According to the reported times, we can conclude that the crawling process is quite efficient. The crawling module for Facebook, which has the heaviest effort due to the large number of user's contacts and their profile information, requires an average of 185 seconds. On the other hand, the crawler of Skype, which has much less user information, requires less than 2 seconds.

The complete picture regarding the performance of our forensics framework can only be drawn if we are given access to *all of the user's data residing in every service*. This is very intrusive, and thus, we continue with a minimal case study where one of the authors assumed the role of the "suspect", and we run our framework with access to all of the author's accounts. The suspect has 517 Facebook contacts, which is more than the average for adult users (338) [31]. Thus, while the performance is quite efficient, normal users are expected to require even less.

The times required for collecting the data available in the suspect's profiles from all the services are reported in Table III. Interestingly, the multi-query requests of our Facebook crawler are very effective, as all the data was collected in less than 7 minutes (we did not download the actual photos but only their URLs and metadata). This process can be significantly faster if we do not collect the photo information. Nonetheless, collecting all the information from Facebook and only the contact information from the remaining services can be completed in less than 10 minutes. Furthermore, we observe that the crawling module for Twitter accounts is quite time-consuming, as it spends over 17 minutes for completing its process. This is mainly due to the strict rate limiting enforced by Twitter's API. Also, in the case of Gmail the data extraction process took 38 minutes for processing 1,297 emails. This entails downloading the whole email content and not just contact information.

Next, we measure the effectiveness of the correlation process. This process correlates seemingly disjoint user accounts across different services, that actually belong to the same user.

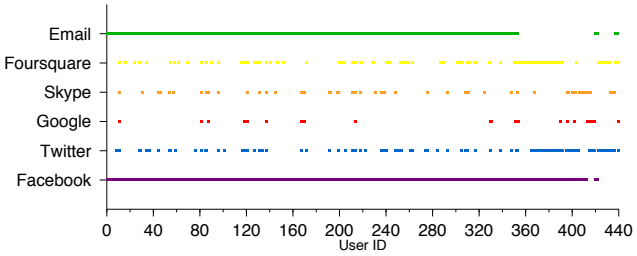

Fig. 7. The associated accounts. A user's account is plotted only if it has been correlated with at least one account from another service.

While our results cannot be used for drawing generalized conclusions due to the small sample, they do offer valuable insight. At first, we manually identify all the user's contacts across the services, to create the "ground-truth". This process requires a large amount of manual work and out-of-band communication with those contacts. The manual verification of the correlated accounts is required, as users may be connected to the suspect within a specific service and not in another, in which they may also have a username completely unrelated to the name registered in the initial service. Thus, we identify the accounts of all those users across the major services and we compare them to the results of the correlation modules.

The results of the experiment assessing our correlation process are presented in Table IV. We consider the suspect's Facebook account as the core dataset for this experiment which has a total of 517 contacts. We also present the number of its Facebook friends that have an account in each one of the other OSNs (*Total_Profiles*) and how many of those users are actually connected with the suspect's profile in that network (*Connected_Profiles*). Some user accounts may have possibly been overlooked during the manual identification of the "ground-truth", or due to users that did not want to reveal the existence of their profiles in other services. The *FB_Correlated* row refers to the number of accounts from each service that have been correlated to the Facebook account of the specific contact. The *All_Correlated* row refers to the overall number of accounts that have been correlated to accounts of any service, also including the *FB_Correlated* accounts. Our correlation modules also discover accounts that belong to the suspect's contacts from other services, that are not connected to the suspect's Facebook account. For example, the suspect may follow a contact on Twitter without being

TABLE IV
NUMBER OF CONTACTS EXTRACTED FROM EACH SOCIAL NETWORK. THE CORRELATION CONTACTS REFER TO THE USERS THAT WERE MAPPED TO A PROFILE THROUGH EACH OF THE CORRELATION TECHNIQUES.

NETWORK	FACEBOOK	TWITTER	SKYPE	GOOGLE+	FOURSQUARE
Connected_Profiles	517	77	64	24	1
Total_Profiles	517	114	113	121	115
FB_Correlated	-	67	45	16	70
All_Correlated	390	98	55	24	70
Completeness	75.4%	85.9%	48.7%	19.8%	60.9%
Discovered	25	13	0	2	38
MODULE	FACEBOOK	YAHOO	FOURSQUARE	FUZZY	ABOUT.ME
FB_Correlated	3	352	63	78	1
Correlated	3	352	98	121	9
Duplicates	0	0	4	5	1
False_Positives	-	0	0	3	0

friends in Facebook. The *Discovered* row contains the number of accounts belonging to the suspect's friends that are not connected in that service. The *Completeness* row indicates the completeness of the correlation and denotes the percentage of accounts of each service that have been correlated to an account of a different service, for users that are Facebook friends with the suspect.

Our system correlated over 75% of the suspect's Facebook contacts to profiles from other services. This means that the data collected for 3 out of 4 users can be significantly enriched by data collected by other services. The best results are achieved for Twitter, where 85.6% of the suspect's contacts are mapped to other services. All the correlations created by our modules are plotted in Figure 7. The points in the plot depict the accounts from various services that have been associated through our correlation process to each user. An alarming finding with privacy implications is that our correlation process is able to map user's Facebook profile to their profiles on different services, even if users have changed their settings to hide their email address from Facebook's search. Furthermore, a large fraction of those users was mapped to their Foursquare account, which further highlights the privacy concerns as location information can expose users to various threats [32].

The lower part of Table IV presents the results obtained by each one of the correlation modules. The row *FB_Correlated* contains the Facebook accounts returned by each module, while the *Correlated* row contains the overall number of results returned by each module, regardless of service (also includes *FB_Correlated* results). Moreover, the *Duplicates* row refers to overlapping results. Specifically, it refers to correlations discovered between accounts, that were also discovered by other modules (i.e., four correlations discovered both by the Foursquare and the Fuzzy modules). The *False_Positives* refer to the accounts returned by each correlation module, that do not correspond to an actual friend of the suspect. We verified manually these profiles in order to identify if they actually belong to a suspect's friend or if they are false matchings. It is also noted that the Facebook module does not return false positives by design, as it can automatically verify if the returned profiles are the expected ones.

Overall, while a larger case study is needed for an accurate evaluation of the tool's efficiency, such a study seems infeasible as users are not willing to grant us complete access to their data. Nonetheless, our case study provides an accurate estimation on the time required for running our forensics tool for a "typical" user, with profiles and data scattered across multiple social and communications services. It highlights the usefulness of the correlation process and the crucial role our correlation plays as part of an automated social forensics toolset. Importantly, it also highlights the privacy risks that users face, and demonstrates the feasibility of automated, large scale personalized attacks. Consequently, this study should work as a strong warning to users, making them aware about the nature of the information they disclose in such services, possible threats, and the need to adopt all available security mechanisms (e.g., two-factor authentication).

VII. RELATED WORK

Forensics. In [33] the authors demonstrate the acquisition of data from the RAM of a desktop PC with a goal of reconstructing the previous Facebook session, by locating some distinct strings. Garfinkel introduced the Forensic Feature Extraction and Cross-Drive Analysis techniques [34] for extracting and correlating information from large sets of images of hard drives. In experiments conducted on 750 drives acquired in the secondary market, the author was able to recover sensitive information ranging from credit card numbers to social security numbers and email addresses.

Mutawa et al. [35] explore what data can be recovered from mobile devices regarding user activities in social networking apps. They reported that both iPhone and Android devices contain a significant amount of valuable data that could be recovered, while Blackberry devices did not contain any such

traces. For example, they were able to recover user IDs, contents of exchanged messages, URLs of uploaded pictures, and timestamps of activities from a directory of the Android Facebook app saved on an external SD card.

In [36] Andriotis et al. investigated the presence of data regarding the use of Wi-Fi or Bluetooth interfaces in system log and database files of Android smartphones. Their results showed that the elapsed time between a criminal activity and the acquisition of the device was critical, as a lot of information was lost from the logs after just a few hours, due to their fixed size. However, database files were found to retain the useful information.

An interesting technique was presented by Mao et al. [37]. They characterized the leakage of information in Twitter and, specifically, if users divulged vacation plans, tweeted under the influence of alcohol or revealed medical conditions. Building activity-classifiers (not only for parsing text) for crime-related topics can assist investigators by highlighting important activities of the suspect.

Data Collection. The work most relevant to ours, which focused on extracting data from social networks in the context of forensics analysis, was by Huber et al. [38]. They extracted data from Facebook through the use of an automated browser and a third-party application, and focused on measuring the completeness of the data their system collected. They also referred to the correlation of users across services, and how analysis of the collected data could be done through graph and timeline visualization. In [39] they also presented connected graphs depicting users that had exchanged messages or had been tagged in the same photo.

Overall, our work presents several differences. First, we provide an extensive framework that extracts data from a large number of social networks and communication services. Second, we have implemented a user correlation component that provides analysts with a unified representation of users, as each contact is represented by their activities that span across multiple services. Third, our dynamic visualization framework with viewpoints of varying granularity enables the analysis of collected data, and focusing on different contacts, services or types of activity. Finally, our evaluation focuses mainly on the accuracy of the correlation process and not as much on the efficiency of the data collection process, as our experiments show that it is completed quickly without presenting any performance bottlenecks.

Account Correlation. The correlation technique where services are used as oracles to map an email address to a profile was first presented in [19], [20]. Also, papers that detect cloned user profiles across social networks [40], [41] have techniques that could be incorporated into our correlation component. Specifically, when the correlation occurs through the fuzzy string matching technique, which might be wrong, profile content and social graph similarities can be used to further ascertain that the correlation is correct.

Data Visualization. Numerous libraries can visualize graphs depicting the structure of social networks. The visualization of online social networks is an active research area, and multiple publications [42]–[44] have focused on implementing visualization techniques. We develop a dynamic framework that associates and visualizes an extensive range of user activities and communication in OSNs, while leveraging various visual libraries. However, there are additional visualizations that can be added. For example, an interesting extension would visualize the social influence between the suspect and online contacts [45].

VIII. Limitations and Future Work

Service login. We have implemented the module for extracting stored credentials and session cookies only for Chrome, but we plan on developing extractors for all the other major browsers found installed in the system.

Moreover, if the suspect's device does not contain any stored credentials or cookies there are possible workarounds. In such a case, our tool supports the manual insertion of user's credentials and also, tries to reuse passwords from other services. These two techniques are supported by the current implementation of our tool.

Manual insertion: available user credentials can be added to a configuration file when acquired through non-technical means: e.g., the suspect reveals the passwords, or the analyst acquires them through social engineering [46]. Law enforcement agencies may request the data or access to the device with a warrant. Nonetheless, companies may not be able to provide access the law enforcement agency even when warrants are provided [47]. In such cases, the agencies cannot access the data or "unlock" the user's device. However, as reported in [28], there are cases where the court has mandated that suspects must unlock their devices or provide their encryption keys and passwords to the authorities.

Password reuse: as users tend to reuse passwords across services [14], [48], if credentials are available for one service, the tool can use them to attempt logging into other services.

Facebook credentials: services allow users to register or login using their Facebook credentials. If the Facebook credentials or session cookie are available, the login module could employ them for logging into other services.

Fuzzy matching. Our current implementation is fairly simplistic. A case study comparing the "distance" values of user names across services for a collection of string matching algorithms [49] can provide insight for creating a more effective module.

IX. Conclusions

The growing importance of data found in online social network profiles for solving criminal investigations necessitates the creation of a complete social forensics framework. We presented our modular system that targets popular online social networks, and consists of components that perform three distinct tasks. First, all data that is reachable from the suspect's profiles is extracted, including the activities of contacts. Second, a series of techniques are employed for automating the correlation process that associates accounts from different services that belong to the same user. This

leads to the creation of abstracted profiles that contain a user's activities regardless the service of origin. Third, our visualization framework provides perspectives that focus on different types of data, and can dynamically change their level of granularity, shifting from aggregated statistics to detailed information. To evaluate the effectiveness of our correlation process, we conducted a minimal case study. While the number of participants is not large enough to extract concrete statistics, the results demonstrate the effectiveness of our process as we were able to reach significant coverage of the set of online contacts.

Overall, our findings highlight the benefit of employing these procedures in criminal investigations, but also stands as a warning about the privacy threats posed to users. Such a framework can be built by cyber-criminals for automatically collecting and correlating user data, and deploying personalized attacks on a large scale.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. We thank Elias Diamantakos for his help in the mobile device modules. This work was supported by the FP7 projects iSocial Marie Curie ITN and NECOMA, funded by the European Commission under Grant Agreements No. 316808 and No. 608533, and the Prevention of and Fight against Crime Programme of the European Commission Directorate-General Home Affairs (projects GCC and ForToo). This work was also supported by the NSF Grant CNS-13-18415. Any opinions, fundings, conclusions, or recommendations expressed herein are those of the authors, and do not necessarily reflect those of the European Commission, the US Government or the NSF.

References

[1] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks," in *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, ser. WPES '05'. New York, NY, USA: ACM, 2005, pp. 71–80. [Online]. Available: http://doi.acm.org/10.1145/1102199.1102214

[2] B. Krishnamurthy and C. E. Wills, "On the leakage of personally identifiable information via online social networks," in *WOSN '09: Proceedings of the 2nd ACM workshop on Online social networks'*. New York, NY, USA: ACM, 2009, pp. 7–12.

[3] F. Adu-Oppong, C. K. Gardiner, A. Kapadia, and P. P. Tsang, "Social circles: Tackling privacy in social networks," in *Symposium on Usable Privacy and Security (SOUPS)*, 2008.

[4] "Symantec - industrial espionage: Targeted attacks and advanced persistent threats (apts)," http://www.symantec.com/threatreport/topic.jsp?aid=industrial_espionage&id=malicious_code_trends.

[5] Mashable, "Vancouver fans riot as canucks lose stanley cup," http://mashable.com/2011/06/15/vancouver-hockey-riot/.

[6] "Department of Law, State of New Jersey. Melanie McGuire found guilty of murder in 2004," http://www.nj.gov/oag/newsreleases07/pr20070423a.html.

[7] "Solving a teen murder by following a trail of digital evidence," http://www.forbes.com/sites/kashmirhill/2011/11/03/solving-a-teen-murder-by-following-a-trail-of-digital-evidence/.

[8] "IACP center for social media, 2012 survey results," http://www.iacpsocialmedia.org/Resources/Publications/2012SurveyResults.aspx.

[9] "The guardian: NSA prism program taps in to user data of apple, google and others," http://www.guardian.co.uk/world/2013/jun/06/us-tech-giants-nsa-data.

[10] "Immersion: a people-centric view of your email life," https://immersion.media.mit.edu/.

[11] "Facebook API client fbconsole," https://github.com/fbsamples/fbconsole.

[12] "Tweepy: Twitter for Python," https://github.com/tweepy/tweepy.

[13] "Chrome Password Decryptor," http://securityxploded.com/chromepassworddecryptor.php.

[14] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse," *Proceedings of NDSS*, 2014.

[15] Facebook., "Facebook Query Language (FQL) Reference," https://developers.facebook.com/docs/reference/fql/.

[16] "Google Geocoding API," https://developers.google.com/maps/documentation/geocoding/.

[17] "Foursquare wrapper." https://github.com/mLewisLogic/foursquare.

[18] "Facebook developers: Email permissions," https://developers.facebook.com/docs/reference/login/email-permissions/.

[19] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel, "Abusing social networks for automated user profiling," in *RAID*, 2010, pp. 422–441.

[20] I. Polakis, G. Kontaxis, S. Antonatos, E. Gessiou, T. Petsas, and E. P. Markatos, "Using social networks to harvest email addresses," in *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2010.

[21] "PhantomJS: Headless WebKit with JavaScript API," http://phantomjs.org/.

[22] "jQuery: The Write Less, Do More, JavaScript Library," http://jquery.com.

[23] "D3.js - Data-Driven Documents," http://d3js.org.

[24] "Google Maps JavaScript API v3," https://developers.google.com/maps/documentation/javascript/.

[25] "Forensic focus: Dropbox forensics." www.forensicfocus.com/Content/pid=429/page=2/#database.

[26] "International business news - android vs. ios," http://www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892.

[27] "Android Developer - ADB," http://developer.android.com/tools/help/adb.html.

[28] "Google and apple won't unlock your phone, but a court can make you do it," http://www.wired.com/2014/09/google-apple-wont-unlock-phone-court-can-make.

[29] "List of searched keywords." http://animalnewyork.com/2012/the-department-of-homeland-security-is-searching-your-facebook-and-twitter-for-these-words/.

[30] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/off: Preventing privacy leakage from photos in social networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15, 2015.

[31] "PEW Research Center - 6 new facts about Facebook," (2014).

[32] J. Krumm, "Inference attacks on location tracks," in *Pervasive Computing*. Springer, 2007, pp. 127–143.

[33] H.-C. Chu, D.-J. Deng, and J. H. Park, "Live data mining concerning social networking forensics based on a facebook session through aggregation of social data," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 7, pp. 1368–1376, 2011.

[34] S. L. Garfinkel, "Forensic feature extraction and cross-drive analysis," *Digital Investigation: The International Journal of Digital Forensics & Incident Response*, vol. 3.

[35] N. Al Mutawa, I. Baggili, and A. Marrington, "Forensic analysis of social networking applications on mobile devices," *Digital Investigation*, vol. 9, pp. S24–S33, 2012.

[36] P. Andriotis, G. Oikonomou, and T. Tryfonas, "Forensic analysis of wireless networking evidence of android smartphones," in *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*. IEEE, 2012.

[37] H. Mao, X. Shuai, and A. Kapadia, "Loose tweets: an analysis of privacy leaks on twitter," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, ser. WPES '11, 2011.

[38] M. Huber, M. Mulazzani, M. Leithner, S. Schrittwieser, G. Wondracek, and E. Weippl, "Social snapshots: digital forensics for online social networks," in *ACSAC*, 2011.

[39] M. Mulazzani, M. Huber, and E. Weippl, "Social network forensics: Tapping the data pool of social networks," in *Eighth Annual IFIP WG International Conference on Digital Forensics*, 2012.

[40] G. Kontaxis, I. Polakis, S. Ioannidis, and E. P. Markatos, "Detecting social network profile cloning," in *IEEE SESOC 2011*.

[41] L. Jin, H. Takabi, and J. B. Joshi, "Towards active detection of identity clone attacks on online social networks," in *Proceedings of the first ACM conference on Data and application security and privacy*, ser. CODASPY '11', 2011.
[42] J. Heer and D. Boyd, "Vizster: Visualizing online social networks," in *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, ser. INFOVIS '05', 2005.
[43] C. Correa and K.-L. Ma, *Visualizing Social Networks*. Springer, 2011, ch. Chapter 11: pp. 307-326.
[44] Z. Shen, K.-L. Ma, and T. Eliassi-Rad, "Visual analysis of large heterogeneous social networks by semantic and structural abstraction," *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, vol. 12, no. 6, pp. 1427–1439, 2006.
[45] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '09'. ACM, 2009.
[46] J. P. Craiger, J. Swauger, and C. Marberry, "Digital evidence obfuscation: recovery techniques," in *Defense and Security*. International Society for Optics and Photonics, 2005.
[47] "The Washington Post - Apple will no longer unlock most iPhones, iPads for police, even with search warrants," http://www.washingtonpost.com/business/technology/2014/09/17/2612af58-3ed2-11e4-b03f-de718edeb92f_story.html.
[48] D. Florencio and C. Herley, "A large-scale study of web password habits," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007.
[49] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.