Toward Usable and Trustworthy Online Monitoring on e-Health Applications

Youna Jung
Department of Computing and Information Sciences
Virginia Military Institute
Lexington, Virginia, United States
e-mail: jungy@vmi.edu

Abstract — **To enable and validate their effectiveness, many e-health applications track how they are used by patients. While online monitoring can improve the accuracy and quality of e-health applications, there is the potential of serious privacy violations. As e-health applications use online monitoring services, sensitive health data could be exposed to not only the healthcare providers but also the monitoring service providers and third-parties such as advertisement companies against wishes of a user. To prevent privacy loss during online monitoring, as a preliminary work, we came up with the idea of a privacy-preserving online monitoring framework, in short PPoM, that helps both of e-health providers and users specify their own policies and enforce user privacy policies during monitoring in systematic manner. In this paper, we extend the idea of the PPoM framework by describing a motivating example of privacy violation during online monitoring and specifying each component in the framework, and demonstrating a prototype of the secure user browser, called PPoM browser.**

Keywords - e-health application; online monitoring; privacy protection; framework; secure monitoring service; secure browser; toolkit.

I. INTRODUCTION

Online monitoring and analytics are essential techniques to evaluate and enhance the performance of online applications. They help the online service providers improve the usability of online applications by collecting user/usage data and analyzing the performance of applications [1][2][3]. In general, there are three different approaches to online user monitoring: 1) log file analysis on the server side, 2) proxy-based monitoring, and 3) use of monitoring scripts provided by online monitoring/analytics services on the client side [4]. Among those approaches, we focus on the third approach because it is widely used and requires less time and effort to collect, analyze, and visualize user/usage data.

Online monitoring and analytics services, such as Google Analytics [5] and Adobe Analytics [6] have been extensively used in a variety of online application areas, such as e-health [7], e-commerce [8], information retrieval [9], and so on. These monitoring/analytics services enable the tracking and recording of user actions and characteristics, such as mouse clicks, frequency of use of an application, time spent in a particular page, media viewed, page navigation sequences, content entered into a textbox, location information, whether a mobile device is being used, and so on.

As e-health applications are becoming ingrained into the everyday life of many people, we focus on the healthcare and wellness domains among various application areas. According to Eysenbach, e-health is an umbrella term that includes a variety of online healthcare applications and systems that use information technologies, such as e-Learning for healthcare, e-Diagnosis, e-Prescribing and online health interventions [10]. It is an emerging field at the intersection of medical informatics, public health and business, referring to health services and information delivered or enhanced through the Internet and related technologies. By using advanced information technologies, including electronic data management and rich interaction skills, e-health applications are capable of 1) providing personalized services, 2) reducing healthcare cost, 3) ensuring easy access regardless of time and place, 4) ensuring consistent quality of services over time, 5) enabling automated data collection/analysis, and 6) demonstrating the potential for having more honest self-reporting by patients. Many e-health applications have been used for online healthcare education [11], healthcare research [12] and recruitment of its participants [13], collecting healthcare data for research or national healthcare purposes, and conducting healthcare interventions to facilitate disease prevention, disease self-management, and health promotion [14].

To accomplish the purposes of e-health applications, most e-health applications provide several of the following functionalities: 1) self-assessment or self-profiling to recognize individuals' health-related status and in turn provide personalized messages and/or healthcare services, 2) continuous communication with patients/users using interactive tools such as online trackers, and 3) wide dissemination of information related to health and safety, presented in text and/or multimedia format. To provide the required functionalities, on one hand, detailed monitoring is critical to confirm that e-health applications are correctly used and to validate their efficacy. In order to do so, e-health applications must collect detailed, and often identifiable, user data including health information. On the other hand, the protection of user privacy is however critical since e-health applications often deal with very sensitive private data, including health status, medical records, and family health histories. Control over the sharing of such information is of the utmost importance and urgency because indiscriminate monitoring, if inconsiderate of user privacy, may result in private health data being used for unwanted purposes and/or shared with unknown people [2][15][16]. In case of e-health

applications, even generic usage data can violate privacy if disclosed. For example, disclosure of the login frequency into an online treatment application for substance abuse can unintentionally reveal a user's medical status. Consequently, it is urgent and critical for research to examine how we can simultaneously achieve these two important yet opposing goals -- monitoring identifiable user data while protecting user privacy.

To enable e-health applications to conduct trustworthy user monitoring without concern for loss of privacy, in this paper, we enhance the Privacy-Preserving online Monitoring (PPoM) framework [1]. In the PPoM framework, online monitoring services collect user/usage data based on users' policies. In addition, users can verify user/usage data being monitored in real-time and strictly enforce user policies on the client side by controlling outgoing messages set from users' browsers. To support non-IT medical staff who do not have enough knowledge and skills on Information Technologies (IT), the PPOM framework provides intuitive and semi-automatic tools that enable them to generate privacy policies and insert monitoring code into their e-health applications. The rest of this paper is organized as follows. In Section II, the limitations of existing online monitoring approaches and the necessity for a secure monitoring in e-health applications are identified with an example scenario. In Section III, the overall architecture of the PPoM framework is described and the detailed functionalities and methods for each component are specified. In Section IV, how the PPoM framework mitigates the privacy vulnerabilities described in Section II from the perspective of e-health service providers. In Section V, we present a prototype of the privacy-preserving browser as a first step in the development of the PPoM framework. The evaluation plans is described in Section VI and related work are introduced in Section VII. The conclusions and future work are presented in Section VIII.

II. MOTIVATION

A. Limitations

Existing online monitoring approaches on e-health applications have two major problems, as follows:

1) Lack of systemactic methods to verify and enforce privacy policies mutually agreed by users and providers: To protect user privacy during online monitoring, a user needs to specify his/her preference in data disclosure while the administrators of an e-health application specify their privacy policy describing what kinds of user data might be monitored, what those data are used for, who those data will be shared with, and how user data are maintained. Users and administrators can specify their policies using policy languages such as P3P [20] and WS-XACML [21]. Once a user agrees to an application's policies, the enforcement of agreed policies has been primarily relied on the honor system [17] within the application without any external verification process. To ensure user privacy, the federal Health Insurance Portability and Accountability Act (HIPAA) [18] stipulates that a healthcare component must not disclose protected health information to another component (HIPAA 164.105.(a)(ii)) with only a few exceptions (HIPAA 164.512). However, it is difficult to expose violations of HIPAA regulations within e-health applications in existing approach. If a provider embeds monitoring code and/or third-party data-collecting ads in its webpages, private data can easily be released regardless of users' wishes. Although this is an obvious violation of HIPAA rules, there are no solutions to systematically detect the application's fraud and prevent user data from undesirable use and disclosure.

To protect user privacy from undesirable use, some online applications anonymize/de-identify user data by deleting identifiers in original data but such anonymized data can often be re-identified/de-anonymized [19]. It is hence not enough to hide user identifiers and we need a new method not to share critical information based on user preferences. In addition, anonymization might not be applicable to some e-health applications that require identifiable user data for personalized services. Without a strong enforcement method, many users are unlikely to consent to online monitoring.

2) Need for professional IT knowledge and skills: At present, professional IT knowledge is needed for developing monitoring-enabled e-health applications with a privacy policy. For example, to specify privacy policies of an e-health application, an application administrator must understand privacy policy languages, such as P3P [20] or WS-XACML [21] and be able to precisely specify the application's policy in that language. In addition, to use an online monitoring service, the administrator must understand the client-side monitoring code (e.g., in JavaScript), and be able to manually insert privacy-preserving code into the original source code (possibly using different languages) for each web object or webpage being monitored. Hence, administrators of e-health applications need to understand at least one language to integrate privacy-preserving monitoring into applications. This is however an impractical expectation for many non-IT administrators, such as doctors, nurses, health educators and communicators. Not only for non-IT clinical staff who manage e-health applications but also for average users who use e-health applications, it is difficult to exactly specify privacy policies and enable their applications/browsers to protect user privacy. The lack of IT knowledge of administrators and users of e-health applications significantly increases the need for easy-to-use tools for a privacy-preserving framework.

In Figure 1, an example scenario shows us the existing monitoring approaches may not be able to prevent privacy leakage. The privacy vulnerabilities in the scenario is as below:

1) Users and/or administrators have to create their own policies manually. To do so, they must first learn

policy languages, such as P3P, APPEL, XPref, or WS-XACML, prior to online monitoring.

2) An administrator needs to manually insert monitoring code to conduct online monitoring/analytics but the task requires IT knowledge about monitoring code and web programming languages.
3) Both users and e-health applications should have their own privacy policies written in existing policy languages. Due to the lack of a standard data schema to specify health data, however, they are unable to define their privacy using a single schema that is shared by the user and the application to express privacy policies in a consistent manner. (In Figure 1, AP^- and UP^- represent the application policies and the user policies in the absence of a standard health data schema.)
4) User preferences regarding data sharing with third-parties might be ignored during online monitoring.
5) There is no way to detect applications' mistakes or fraud. For example, untrusted policies are represented as AP_f^- in Figure 1, but it is not possible to enforce the mutually agreed privacy policies systematically. Currently, policy "enforcement" amounts to trusting the application.

B. Requirements

Towards trustworthy and highly usable online monitoring in e-health applications, the following requirements should be satisfied:

1) For strict enforcement of user privacy policies

- Online monitoring services that are aware of user privacy policies rather than application policies.
- Verification methods to ensure that an application complies with policies mutually agreed by providers and users on user side.
- Enforcement methods to protect user privacy on user side in case of privacy violation during online monitoring.

2) For practical use by non-IT users and staff

- User-friendly interfaces to intuitively specify privacy policies and monitoring objects.
- Automatic generation of privacy policies for e-health applications.
- Systematic conversion of existing applications to privacy-preserving applications in which privacy-aware monitoring code is embedded.

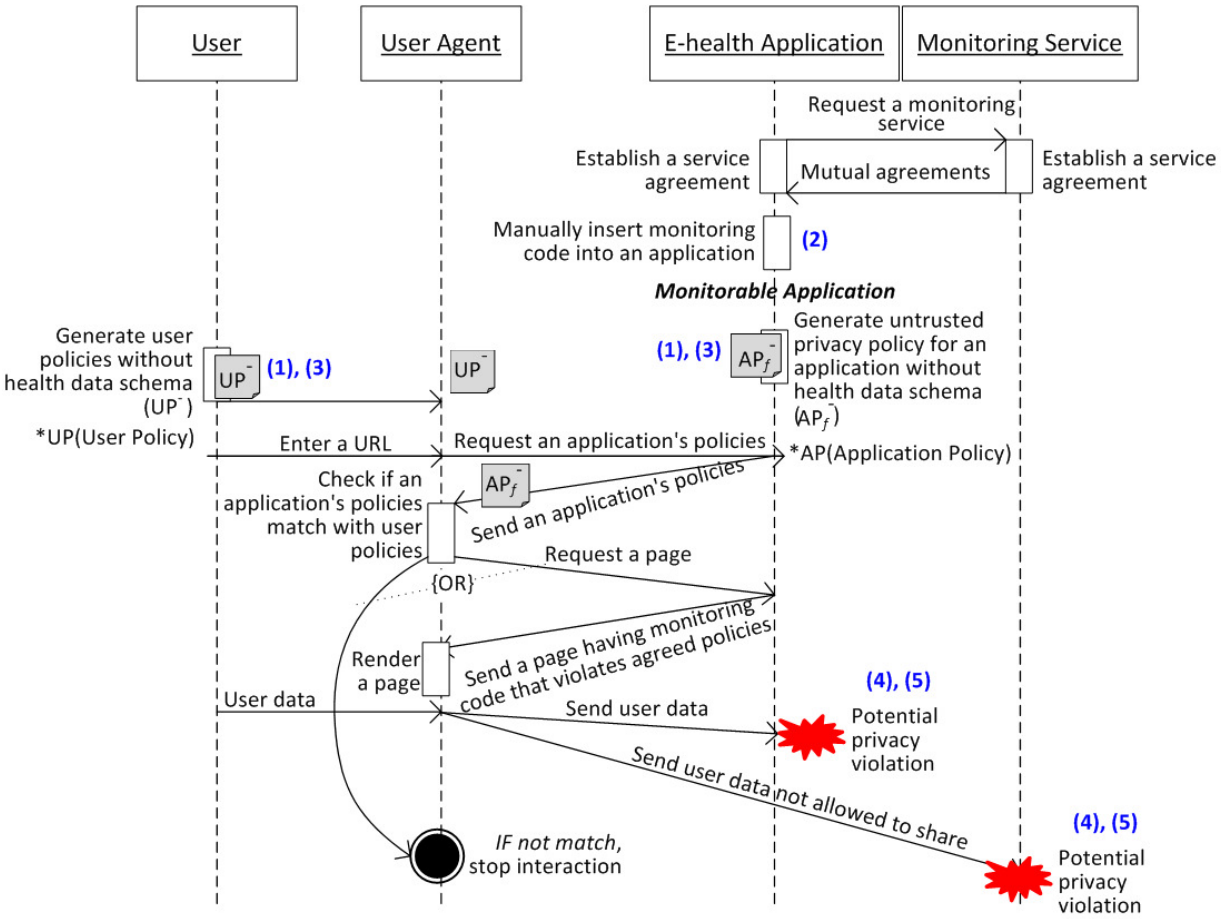

Figure 1. Existing online monitoring approach (User agent is a software agent in a user browser that acts as a client of the application server and captures a user's privacy preferences from a user policy (UP)).

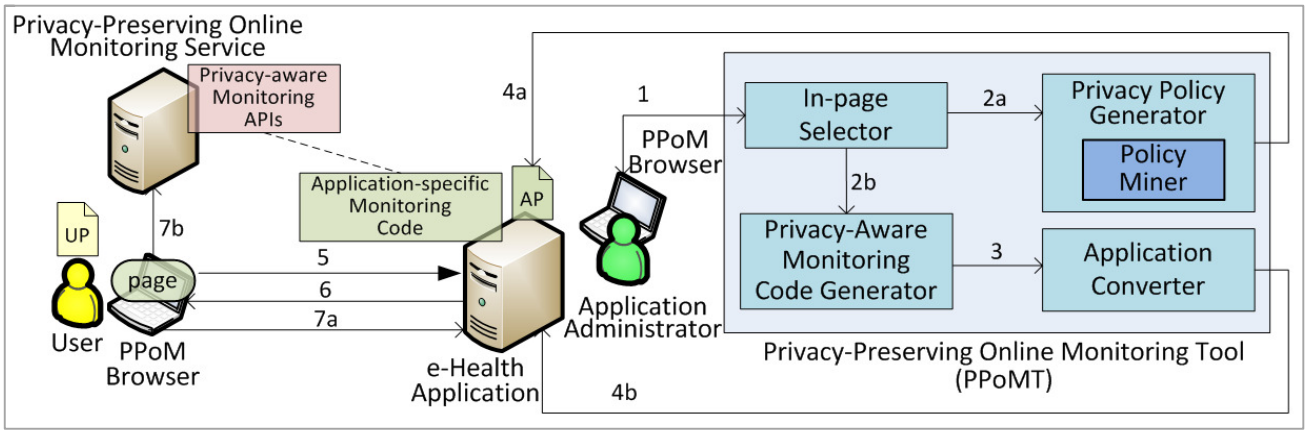

Figure 2. Overall architecture of PPoM framework.

III. Overall Architecture

To fulfill the requirements described above, the idea [1] of the PPoM framework that rigorously protects user privacy by referring user policies and enforcing them on user side was proposed as shown in Figure 2. In this paper, we describe the details of individual services that include the privacy-preserving online service (PPoM service), user browser (PPoM browser), and online monitoring tool (PPoMT).

A. Privacy-Preserving Online Monitoring Service

The problems of existing online monitoring services are twofold. First, users are forced to choose between two options: opt-in or opt-out, and not allowed to choose data to be monitored in fine-grained level. Second, user data are collected based on the decision of providers. Once a user gives consent to monitoring (opt-in), all user activities on the web objects in which monitoring code is embedded are captured regardless of users' privacy preferences. None of existing services considers users' privacy preferences during monitoring.

To address the issue of ignorance of user preferences during monitoring, the PPoM service gathers only authorized user/usage data that users allow to monitor. By specifying privacy policies, users determine which data can be monitored. User policies will be then enforced by the PPoM service that understands users' privacy policies and enables selective monitoring based on user preferences. As for other monitoring services, we envision a set of service APIs (Application Programming Interfaces) that can be invoked by code snippets inserted in the original application code in order to monitor events or data of interest. The PPoM service does not capture data that users want to never be monitored. The PPoM service enables the monitoring code snippets inserted into the original application program to interpret user preferences written in the policy language and send only allowed data to the monitoring code on the server side. Unlike the existing approach where user data are collected based on application policies and enforced by the application itself, users will now be able to choose what data can be monitored and the user policies will be enforced by a trustworthy third-party monitoring service.

Using the PPoM service, e-health applications can guarantee effective protection of user privacy by providing a way to enforce user policies in a systematic manner rather than simply providing a written agreement. The concrete enforcement framework assists both monitoring services and e-health applications in building better reputations and accelerates active participation in e-health applications without concerns about privacy leakage. In addition, it can be used to compel and enable e-health applications to obey HIPAA regulations.

B. Privacy-Preserving User Browser

If an e-health application is trustworthy and both users and administrators can correctly specify their own policies, then existing approach may be able to guarantee secure online monitoring. However, this protection presumes the application's honesty and the user's ability to specify privacy policies precisely. Let us assume that an e-health application publishes untrusted policies that differ from its actual monitoring behavior. Once user's policies and the application's policies are matched, a user will start interaction with the dishonest application and his browser will send unconsented data to the application.

However, user privacy must be protected even if a user is exposed to untrustworthy parties, such as indiscriminate monitoring/analytics services that conduct online monitoring in violation of user privacy, naïve applications that do not have their own privacy policies or dishonest applications that write and/or enforce their policies dishonestly. To address the above security requirement, the PPoM-enabled web browser (in short, PPoM browser) protects user privacy on the user side. For easy adoption, we implemented a browser extension that enables existing browsers (e.g., Explorer and Chrome) to perform following tasks:

1) It presents all user data being monitored by online monitoring service(s) in web pages: A PPoM browser

displays all monitoring data and recipients of data on the screen by analysing source code of web pages.

2) It allows a user to decide which usage and his/her data can be disclosed: A PPoM browser inspects data types, data values, and destinations of all outgoing message according to user policies. If it detects malicious monitoring that violates user policies, it alerts a user to unauthorized monitoring. For example, if the user decides to disclose his/her medical history to a first-party (e.g., an e-health application website), then a PPoM browser only allows outgoing messages to the e-health application and blocks other messages to other entities (such as advertisement companies or social networking sites) even if the application does not have its own privacy policy or inserts monitoring code to collect his/her medical history data.

3) It refines the user's privacy policies based on updated user preferences: If a user' privacy policies are naïve or incorrect, the proposed approach based on user policies cannot protect user privacy successfully. However, It is very difficult for average users to specify precise privacy policies because it is hard to understand the correlation between description of user data in privacy policies and web objects in web pages. To enhance the proposed PPoM framework, a PPoM browser allows users to refine their preferred policies by intuitively selecting what user/usage data can be monitored in web pages.

C. Privacy-Preserving Online Monitoring Tool

As pointed out in Section II.A, it is difficult for non-IT administrators to have professional IT knowledge and skills that necessary for trustworthy online monitoring. The PPoM Tools (PPoMT) helps health professionals by enabling them to specify privacy policies for their healthcare applications, and/or to easily convert their existing applications into privacy-preserving applications that analyze user/usage data without violating user privacy. While motivated by the needs of non-IT administrators, this tool could also improve the efficiency of developers who are required to insert monitoring code into their applications.

To achieve the purpose described above, the PPoMT provides the following services:

1) *User-friendly interfaces for selection of web objects*: To allow application administrators to intuitively select web objects to be monitored and specify privacy policies associated with the objects, the PPoMT provides user-friendly interfaces for selection of web objects to be monitored and corresponding policies.
2) *Semi-automatic conversion of existing e-health applications to privacy-preserving applications*: The biggest obstacle to the use of existing monitoring services is that they require manual insertion of monitoring code, often written in JavaScript, into online applications. Even if a privacy-preserving policy language and monitoring client code are successfully developed, they may still remain too complex for use by users other than IT experts. Existing website optimizers (such as Optimizely, Visual Website Optimizer, Loop11, and ABtests) allow users to improve applications through a visual editor without manual modification of source code. However, those tools are mainly focused on page layout and design tasks, such as changes on size/style/color/position of page elements (e.g., buttons and text). They typically do not have functionalities to modify existing applications in order to conduct online monitoring. To overcome this limitation, the PPoMT allows using online monitoring and analytics services without needing prerequisite IT knowledge or skills.

To do so, it is necessary to analyze an application's source code. There are two types of web pages, static web pages and dynamic web pages. Static web pages, called flat pages or stationary pages, are displayed in user browsers as they are stored in an application server, while dynamic web pages are displayed with dynamically changing web content based on environmental or user context such as time, user location, and browsing history. There are two different approaches to the creation of dynamic web pages: server-side and client-side. In the server-side approach, web contents can be dynamically created by server-side scripting languages such as PHP or ASP. JavaScript code, once downloaded into a browser, can also communicate with a server, enabling web contents to also be changed in a browser by real-time interactions with a server after the downloading of a webpage. In the client-side approach, web contents may be dynamically created by client-side JavaScript. In case of server-based dynamic web pages, an administrator needs to enter URLs for each page to monitor. In case of static pages and client-side dynamic pages, an administrator can either upload HTML source code or simply specify page URLs. Once the source code is obtained, the PPoMT analyzes HTML source code exactly and then add intuitive and friendly user interfaces into web pages in order to enable non-IT administrators to select monitoring objects and describe corresponding policies. To support fine-grained privacy-preserving monitoring in web-object level, each monitoring object must be uniquely identified. If an e-health application does not maintain unique identifiers (IDs) for web objects, the PPoMT must create object IDs before creating monitoring code.

Once the code modifications are made, e-health administrators must deploy the modified source code in their server. In the cases of static web pages or client-based dynamic web pages, administrators can easily deploy the updated source code by overwriting the directory that contained existing source code. In the case of server-based dynamic web pages, they will have to manually insert the newly created monitoring code into a PHP snippet that generates the head part of each HTML page.

3) *Semi-Automatic generation of privacy policies for e-health applications*: Many online applications have published their privacy policies to be regarded as a trustworthy application that cares about protecting user privacy. Because of the sensitivity of data being exchanged, it is especially important for e-health applications to gain users' trust. However, specifying a privacy policy is often beyond the ability of most e-health administrators who may be unfamiliar with online privacy policy languages. To help them publish privacy policies for their own applications, the PPoMT creates the privacy policies for a given application by gathering all monitoring objects and corresponding privacy policies that are selected by administrators.

Each service listed above is implemented as an individual tool within the PPoMT: the In-page Selector, the Monitoring Code Generator, the Privacy Policy Generator, and the Application Converter. The detailed explanations for each tool are following:

1) In-page Selector: It is a server-side software module that is capable of generating modified webpages that have user-friendly interfaces for selection of web objects to be monitored and corresponding policies and delivering user selections to the Privacy Policy Generator and the Monitoring Code Generator. It sends modified webpages to the PPoM browser of an administrator. By clicking on web objects that are displayed in the PPoM browser (e.g., a button, link, text, image, or video), the administrator can select which objects to monitor without any IT knowledge and skills. Additionally, he can declare corresponding privacy policies.

2) Monitoring Code Generator: It generates privacy-aware monitoring code by receiving an administrator's selection on monitoring objects and associated privacy policies from the In-page Selector. The generated code will be varied depending on monitoring services.

3) Privacy Policy Generator: It helps non-IT administrators who may be unfamiliar with online privacy policy languages publish privacy policies for their own applications. Using the Policy Miner, it derives the privacy policies for a given application by analyzing all monitoring objects and corresponding privacy policies that are selected by the administrator.

4) Application Converter: It helps non-IT administrators to update source code of an existing application by assisting them in inserting the privacy-aware monitoring code generated by the Monitoring Code Generator. Once the code modifications are made, an administrator needs to deploy the modified source code in their server.

IV. PRIVACY-PRESERVING MONITORING USING PPOM

For better understanding of how the PPoM framework protects user privacy during online monitoring and supports non-IT administrators and users, we present its operation flows and example use cases in this section.

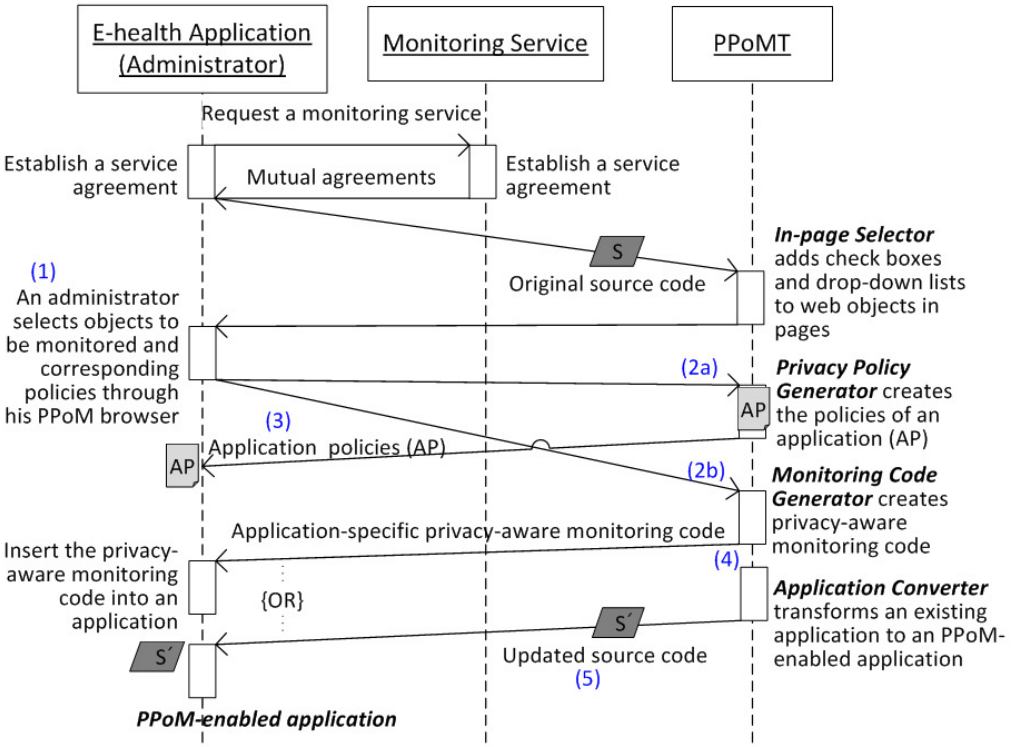

Figure 3. Operation flow of administrators of e-health applications in PPoM.

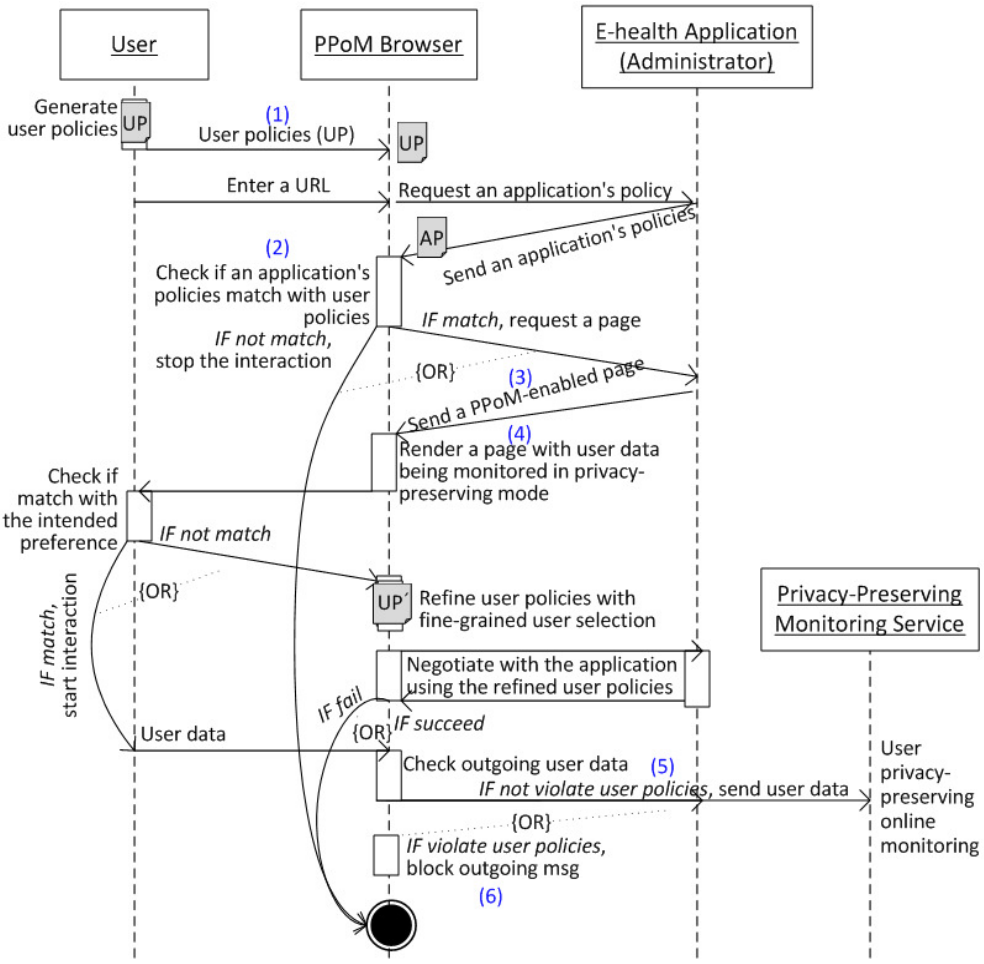

Figure 4. Operation flow of users in PPoM.

A. For online healthcare providers

Let us assume that Alice, a medical doctor, administrates an e-health application that assesses the impact to people following exposure to traumatic events. To trace patients' activities and collect health-related data, Alice wants to conduct online monitoring on her applications, but it is impossible for her to use existing online monitoring services due to lack of IT-related knowledge.

If she uses the PPoMT, she can easily transform her application into a monitoring-enabled application without any IT knowledge. For better understanding, let us look at its operational flow in detail. First, she needs to upload the source code of her application or enter the URL(s) of the application's webpage(s). Second, she selects objects to be monitored through the user-friendly interfaces generated by the In-page Selector, and specifies corresponding privacy policies (Figure 2.(1) and Figure 3.(2)). Third, her selections are delivered to the Policy Generator (Figure 2.(2a) and 3.(2a)) and the Monitoring Code Generator (Figure 2.(2b) and Figure 3.(2b)). Forth, the Privacy Policy Generator then creates the applications' policies by analyzing selected monitoring data and policies, while the Application Converter (which enables the application to perform privacy-preserving monitoring by inserting monitoring code generated by the Monitoring Code Generator into the original source code (Figure 2.(3) and Figure 3.(4)) produces the updated source code. Fifth, Alice deploys the created application policies (Figure 2.(4a) and Figure 3.(3)) and the updated source code in the application server (Figure 2.(4b) and Figure 3.(5)).

B. For Users

Let us assume that Bob is one of Alice's patients. Alice recommends Bob to use her e-health application every week to assess his mental and physical health but he hesitates to use the application due to privacy concern. If Bob uses the PPoM browser, he may want to use an e-health application without privacy loss. Towards this, Bob is first required to specify his own privacy policies and store his policies in his browser before using an e-health application (Figure 4.(1)). His PPoM browser then compares his privacy policies and application policies when he enters a url of Alice's application (Figure 4.(2)). If they match, the application server sends PPoM-enabled pages, which privacy-aware monitoring code is embed in (Figure 2.(5)(6) and Figure

4.(3)). As he interacts with the application, the PPoM browser displays all user/usage data being monitored to enable users to verify privacy protection during online monitoring (Figure 4.(4)). If there is no privacy violation, the privacy-aware monitoring code collects only authorized user data according to policies of him specified by him for the sake of himself (Figure 2.(7a)(7b) and Figure 4.(5)). To ensure enforcement of user policies, his PPoM browser blocks outgoing messages that violate his privacy policies on the user side (Figure 4.(6)).

V. PROTOTYPE IMPLEMENTATION

As a first step in the development of the PPoM framework, we implemented a prototype of the PPoM browser. It analyzes an e-health application's HTML source code in which PPoM monitoring code is embedded, displays a webpage with information about monitoring status, and prevents privacy leakage by blocking outgaining message containing unconsented user/usage data.

An example use of the prototype of PPoM browser is shown in Figure 5. When a user specifies a URL of an e-health application such as the online weight tracker of *sparkpeople.com*, an interaction is initiated. If a user turns on privacy-preserving mode, a PPoM browser renders a web page that is sent by an server of *Spark People* with information about which data being monitored in that page. The PPoM browser displays which user data is being monitored and who is the recipient of the data by using different-colored check marks. In addition, the types of data that are being monitored such as HTTP, Clientevents, and Cookies types of Dynamic data schema are also shown in the status bar (e.g., Your Goal and An answer to the question, How Active Are you? in this example).

If the monitoring data are in accordance with the user's policy, then the user keeps interacting with the application. Otherwise, the user can change the monitorable user data and its disclosure level using one of the following four options: 1) can be disclosed to all parties (*green check mark*), 2) can be disclosed to only first party (*orange check mark*), 3) prohibited to use/disclose it (*red check mark*), and 4) preference is not described yet (*no check mark*). With the specific user selection in web object level, the browser refines user policies and negotiates with the application using the refined policies. If the negotiation is successful, the interaction continues, otherwise the browser stops the interaction with the application. If online monitoring is acceptable to the user (e.g., based on past usage experience and/or trust), he/she can turn off the privacy-preserving mode. Using the PPoM browser, users can easily know what user data are being monitored and prevent user data from unwanted monitoring by turning on privacy-preserving mode.

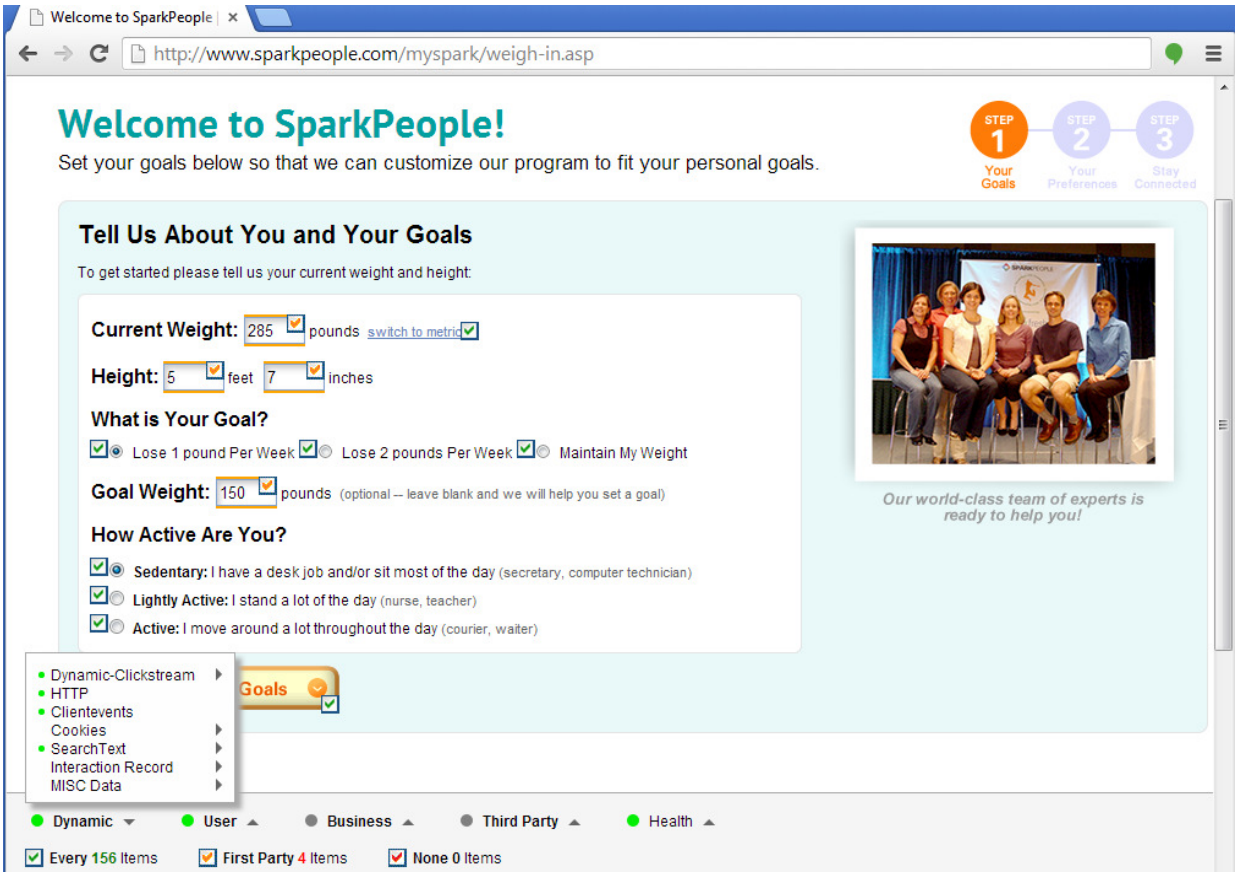

Figure 5. Example use of the prototype of PPoM browser

VI. EVALUATION PLAN

To evaluate the successful blockade ratio of the number of messages containing unauthorized data (*c*+*d* in Table I) to the number of blocked messages among them (*c* in Table I).

TABLE I. OUTGOING MESSAGES FROM A PPOM BROWSER

	Sent	Blocked
Allowed	*a*	*b*
Not allowed	*c*	*d*

In addition, the performance of the proposed privacy-preserving monitoring service will be evaluated by calculating the ratio of the number of all monitoring data (*e*+*g* in Table II) to the number of unauthorized but collected data (*g* in Table II).

TABLE II. USER DATA MONITORED BY THE PPOM SERVICE

	Monitored	Not monitored
Allowed	*e*	*f*
Not allowed	*g*	*h*

To test the developed tool, PPoMT, we plan to compare a user's selections of monitorable data and corresponding privacy preferences with monitoring scripts or source code that generated by PPoMT. The correctness of the generated privacy policies is also required to be verified. To do so, we plan to create a variety of anticipated privacy policies and then compare the anticipated policies with resulting policies that are generated by PPoMT. In Table III, the overall evaluation plans and methods for each components of the proposed PPoM framework are presented. As you can see, we plan to conduct human-subject usability tests in order to test the user-friendliness of the PPoM framework.

VII. RELATED WORK

In this section, we introduce existing work on privacy protection during online monitoring on both the service provider side and the user side.

A. Provider-side privacy protection for online monitoring

To guarantee secure user monitoring, a few methods have been proposed by providers of online applications and monitoring services. Some third-party advertising companies have voluntarily begun to insert an 'Adchoices' icon into their ads to increase user awareness of online tracking. However, it has been found that the icon was not very effective at making users aware of tracking occurrences [21]. As a middleware approach, Privad [22][23] is proposed to conceal a user's activities from an advertising network by interposing an anonymizing proxy between the browser and the ad network. However, the adoption of a proxy-based middleware may not be a feasible solution to small-size e-health applications because of its huge overhead requirements. In addition, it is useless if an e-health application requires identifiable user data to analyze performance of applications at the individual level.

TABLE III. OVERALL EVALUATION OF THE PPOM FRAMEWORK

Component		Measure	Control group	Evaluation Method
PPoM Service		Accuracy	All data collected by existing monitoring services	Percentage of data that can be protected by the privacy-aware monitoring service
PPoM Browser		Accuracy	All data collected by existing monitoring services	Percentage of data that can be protected by the PPoM browser extension
			Monitored data collected by a monitoring service	Completeness of monitored data displayed by PPoM browser extension
			Ideal user policies	Comparison of ideal and refined policies
		Usability	None	Usability survey of PPoM browser
PPoMT	In-Page Selector	Accuracy	Webpages displayed by existing browsers	Completeness of webpage display by the In-page Selector
		Usability	None	Usability survey of webpages modified by the In-page Selector
	Application Converter	Accuracy	Monitored data without privacy violations	Privacy violations of monitored data collected by deployed code
	Privacy Policy Generator	Accuracy	Ideal application policies	Correctness of Policy Miner policies

B. User-side privacy protection for online monitoring

There are two major approaches for privacy protection on the user side, the browser-based approach and the policy-based approach.

1) Browser-based approaches: To protect user privacy in user side, browser-based approaches have been proposed. Adnostic [24], a browser extension, is capable of behavioral profiling and targeting in users' browsers to select effective ads while not sending user data to third-party ad companies. RePriv [25] enables browsers to conduct user interest mining and only share the resulting encapsulated interests with third-parties. Both Adnostic and RePriv have only focused on targeted advertising and/or personalization but have not considered online monitoring services. As a simple solution to indiscriminate online monitoring, using opt-out cookies and/or a blocked-application list have been recommended. Opt-out cookies are, however, fragile because they can be easily disabled or deleted by a third party [26][27]. Setting a block list in a browser can effectively block malicious applications but currently this approach blocks any listed application in its entirety and does not support fine-grained blocking at the data level.

2) Policy-based approaches: As a policy-based protection approach, Privacy Bird [28] has been proposed. It is a P3P user agent that reads P3P policies of online applications and lets users know whether the application policies and user preferences are matched. If policies are not matched, a bird icon turns red. A user can get information by clicking on a red bird icon. However, Privacy Bird is only able to check the acceptability of application's P3P policies, so users cannot check all user data monitored at the application's data level. Moreover, it does not support policy refinement. HummingBird [30] is a privacy-preserving online social network system. Unlike Twitter where all tweets are visible to everyone, it restricts accesses to tweets based on user-defined access control list (ACL). In HummingBird, tweeters encrypt their tweets and then the HummingBird server distributes a decryption key to only authorized followers. To do so, the Hummingbird client deals with key management and tweet encryption. However, it does not encrypt other user data such as user profile and also is not concerned with user privacy policies at all.

VIII. CONCLUSION AND FUTURE WORK

Although many e-health applications enable people to access healthcare services in easy and convenient way at the reduced cost, the lack of reliable and effective methods of privacy protection has been the biggest obstacle to the growth of e-Health applications. Without a suitable solution, people keep hesitating to use e-Health applications due to privacy concerns. To address the privacy protection issue above, the PPoM framework was proposed as a preliminary work in order to protect user privacy in both the application side and the user side. By using the PPoM framework, secure online monitoring can be guaranteed in the entire process of online monitoring, and in turn it accelerates practical use of e-health applications. Towards this goal, in this paper, we enhance mechanisms for each component in the PPoM Framework and present a prototype of the PPoM browser as a first step in the development of the PPoM framework but a few challenges still need to be pursued in the future:

- Development of the privacy policies to specify preferences on healthcare data in fine-grained level.
- Development of a privacy-preserving monitoring service that protects user privacy based on user policies.
- Development of the PPoMT for non-IT medical staff.
- Evaluation of individual components and an integrated framework.
- Field test associated with actual clinical trials.
- Development of a threat model for PPoM and security test using a threat model.

REFERENCES

[1] Y. Jung, "Privacy-Preserving Online Monitoring Framework for e-Health Applications," Proc. Fourth International Conference on Global Health Challenges (GlobalHealth), IARIA, Nice, France, 2015, pp. 18-24.

[2] J. R. Mayer and J. C. Mitchell, "Third-Party Web Tracking: Policy and Technology," Proc. IEEE Symp. on Security and Privacy (SP '12), IEEE Press, 2012, pp. 413-427, doi=10.1109/SP.

[3] B. A. Schroeder, "On-line monitoring: A tutorial," IEEE Computer, vol. 28, no. 6, pp. 72-78, 1995.

[4] N. Schmucker, "Web Tracking," SNET2 Seminar Paper (Online), 2011. http://www.snet.tu-berlin.de/fileadmin/fg220/courses/SS11/snet-project/web-tracking_schmuecker.pdf [retrieved: May, 2016].

[5] Google Analytics. http://www.google.com/analytics/index.html [retrieved: May, 2016].

[6] Adobe Analytics. http://www.adobe.com/sea/solutions/digital-analytics.html [retrieved: May, 2016].

[7] M. N. K. Boulos et al., "CAALYX: a new generation of location-based services in healthcare," Int. J. Health. Geogr., vol. 6, no. 1, March 2007, pp. 1-6, doi:10.1186/1476-072X-6-9.

[8] C. Hoelscher and H. Dietrich, "E-Commerce Personalization and Real-Time Site Monitoring," in Designing personalized user experiences in eCommerce, Kluwer Academic Publishers, 2004, pp. 95-117.

[9] C. L. Borgman, S. G. Hirsh, and J. Hiller, "Rethinking online monitoring methods for information retrieval systems: from search product to search process," J. Assoc. Inf. Sci. Technol., vol. 47, no. 7, pp. 568-583, July 1996.

[10] G. Eysenbach, "What is e-health?," J. Med. Internet. Res., vol.3, no. 2, 2001.

[11] J. M. Bernhardt and J. Hubley, "Health education and the Internet: the beginning of a revolution," Health. Educ. Res., vol. 16, no. 6, pp. 643-645, 2001.

[12] E. M. Daley, R. J. McDermott, K. R. B. McCormack, and M. J. Kittleson, "Conducting web-based survey research: a lesson in internet designs," Am. J. Health. Behav., vol. 27, no. 2, pp. 116-24, March-April 2003.

[13] D. F. Duncan, J. B. White, and T. Nicholson, "Using Internet-based Surveys to Reach Hidden Populations: Case of Nonabusive Illicit Drug Users," Am. J. Health. Behav., vol. 27, issue 3, pp. 208-218, May-June 2003.

[14] K. E. Evers, "eHealth promotion: the use of the Internet for health promotion," Am. J. Health. Behav., vol. 20, issue 4, pp. 1-7, March-April 2006.

[15] M. Bilenko and M. Richardson, "Predictive client-side profiles for personalized advertising," Proc. ACM SIGKDD conf. on Knowledge discovery and data mining (KDD '11), ACM New York, 2011, pp. 413-421.

[16] A. McDonald and L. F. Cranor, "Beliefs and Behaviors: Internet Users' Understanding of Behavioral Advertising," TPRC 2010 Social Science Research Network (SSRN), August 16, 2010, pp. 1-31, http://ssrn.com/abstract=1989092.

[17] Wikipedia, "The Honor System," http://en.wikipedia.org/wiki/Honor_system [retrieved: May, 2016].

[18] U.S. Department of Health and Human Services Office for Civil Rights, "HIPAA Administrative Simplification Regulation Text, 45 CFR Parts 160, 162, and 164 amended through March 26," 2013, pp. 1-115, http://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/administrative/combined/hipaa-simplification-201303.pdf.

[19] P. Ohm, "Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization," UCLA Law Review, vol. 57, pp. 1701-2010, August 2009.

[20] P3P 1.1. http://www.w3.org/TR/P3P11/ [retrieved: May, 2016].

[21] WS-XACML 1.0. http://xml.coverpages.org/Anderson-WS-XACML v10.pdf [retrieved: June, 2015].

[22] M. Hastak and M. J. Culnan, "Online Behavioral Advertising "Icon" Study," Future of Privacy Forum (online), 2010, http://www.futureofprivacy.org/2010/02/15/online-behavioral-advertising-icon-study [retrieved: June, 2015].

[23] S. Guha, B. Cheng, and P. Francis, "Privad: practical privacy in online advertising," Proc. USENIX conf. on Networked systems design and implementation (NSDI'11), USA, 2011, pp. 169-182.

[24] A. Reznichenko, S. Guha, and P. Francis, "Auctions in do-not-track compliant internet advertising," Proc. ACM conf. on Computer and communications security (CCS '11), New York, 2011, pp. 667-676, doi=10.1145/2046707.2046782.

[25] V. Toubiana, H. Nissenbaum, A. Narayanan, S. Barocas, and D. Boneh, "Adnostic: Privacy Preserving Targeted Advertising," Proc. Symp. on Network and Distributed System Security, March 2010, pp. 1-21.

[26] M. Fredrikson and B. Livshits, "REPRIV: Re-Envisioning In-Browser Privacy," Proc. IEEE Symp. on Security and Privacy, May 2011, pp. 1-15.

[27] P. Leon et al., "Why Johnny can't opt out: a usability evaluation of tools to limit online behavioral advertising," Proc. ACM Conf. on Human Factors in Computing Systems (CHI '12), USA, 2012, pp. 589-598, doi=10.1145/2207676.2207759.

[28] M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle, "Flash cookies and privacy II: Now with HTML5 and etag respawning," Social Science Research Network, July 2011, http://dx.doi.org/10.2139/ssrn.1898390.

[29] Privacy Bird, http://www.privacybird.org [retrieved: May, 2016].

[30] E. D. Cristofaro, C. Soriente, G. Tsudik, and A. Williams., "Hummingbird: Privacy at the Time of Twitter," Proc. of the IEEE Symposium on Security and Privacy (SP'12), IEEE Computer Society, Washington D.C., USA, 2012, pp. 285-299, DOI=10.1109/SP. 2012.26