Augmenting the Web with Accountability

Oshani Seneviratne
CSAIL, MIT
oshani@mit.edu

ABSTRACT

Given the ubiquity of data on the web, and the lack of usage restriction enforcement mechanisms, stories of personal, creative and other kinds of data misuses are on the rise. There should be both sociological and technological mechanisms that facilitate accountability on the web that would prevent such data misuses. Sociological mechanisms appeal to the data consumer's self-interest in adhering to the data provider's desires. This involves a system of rewards such as recognition and financial incentives, and deterrents such as prohibitions by laws for any violations and social pressure. Bur there is no well-defined technological mechanism for the discovery of accountability or the lack of it on the web. As part of my PhD thesis I propose a solution to this problem by designing a web protocol called HTTPA (Accountable HTTP). This protocol will enable data consumers and data producers to agree to specific usage restrictions, preserve the provenance of data transferred from a web server to a client and back to another web server, and more importantly provide a mechanism to derive an 'audit trail' for the data reuse with the help of a trusted intermediary called a 'Provenance Tracker Network'.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous

Keywords

Accountability, Web Protocols, Usable Security and Privacy, Usage Restrictions

1. PROBLEM

In the two decades since the creation of the web, big data silos have created many issues relating to data ownership by making it extremely difficult to share data [1]. The web is also responsible for a tremendous shift in companies' business model, as it enabled them to offer information and services of any kind in exchange for users' personal data. Although access control systems are often successful in managing access to resources on the web, they are ineffective in preventing information leakages as it is very easy to copy and/or aggregate and infer information on the web [2]. Also often times there are adverse consequences when the data consumers use these data items for purposes that the the data publisher did not intend them to be used for [3]. While site specific privacy controls protect the users within the 'walled gardens', most privacy breaches happen when the information is taken out of context [4]. Usage restrictions can be defined as extensions of access control. These take the form of actions that can and have to be performed over data after access has been granted. Examples include "delete after 10 days" or "notify owner upon access or reuse" or "restrict playback to a particular hardware platform". However, such enforcement mechanisms are thought to be overly prohibitive [5]. Therefore, there is a need for a solution that transcends these site-specific privacy controls and do not impose very restrictive rights management controls.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

2. STATE OF THE ART

Various machine readable approaches to describing privacy policies have been proposed over many years. P3P (Platform for Privacy Preferences) protocol was developed at the W3C with the intention of communicating the privacy policies of websites to user-agents who connect with them [6]. The P3P recommendation allows website operators to express their data collection, use, sharing, and retention practices in a machine-readable format. A user-agent can retrieve a machine readable privacy policy from the web server and respond appropriately (for e.g. display symbols or prompt the user for action). However, P3P has several limitations: a complicated language to express policies, and the inability to express preferences on third party data collection, or multiple privacy policies for one web page [7]. These limitations have prevented P3P from wide adoption. In addition to that, there has been lot of research on *enforcement* of usage restrictions after the users have been given access to the resource. In particular, Kumari et al. in their work on 'Distributed Usage Control' propose enforcement of usage restrictions with regards to privacy sensitive personal data at various levels of abstractions in the operating system, i.e. by disabling the print-screen button or not allowing the data item to be copied over based on the usage control policy set [8].

Project DReam describes an architecture where users can use Digital Rights Management (DRM) to control use of content under fair use terms [9]. The system they describe requires the user to connect to an anonymizing agent for authentication and assert fair use on any of the user-owned content. A user interface on the DRM software that is used to manage the content will ask the user to enter whether the reuse is for review purposes, educational uses, parody, or for other purposes. It will also ask the jurisdiction in which the content will be reused. The anonymizing agent will relay this information to the copyright owner for auditing.

Specific to sharing of geo-location data, several proposals on how to negotiate privacy policies have emerged within the IETF and the W3C recently. IETF's GeoPriv proposal attempts to put privacy policies in the hands of users instead of services, where a user transmits her own privacy preferences about how her location data should be used, while the websites are bound by their market or legal obligations to

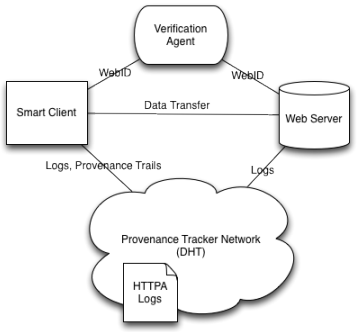

Figure 1: Key Entities and Interactions in HTTPA

respect those preferences [10]. W3C's Geolocation API also advocates that websites disclose their data usage practices to the user [11], although it is rarely practiced by most websites that implement the API [12]. The Simple Policy Negotiation for Location Disclosure proposal describes a system that lets a user have a dialogue with a website that uses her location data before disclosure [13].

Mozilla Privacy Icons takes a simple icon-based approach inspired by the Creative Commons [14]. Instead of specifying every possible type of privacy and data-handling scenario, they specify only a few common privacy scenarios that users can encounter: such as information sharing, storage, monetization, deletion and contact/notification. The icons are designed to be easy to use and be understood by ordinary end-users. As online businesses are looking for ways to build trust and manage consumer expectations through transparency, choice, and accountability, these privacy icons seem to be a timely solution. However, since it is detrimental for sites that violate user privacy to label themselves as such, it would be up to the browser or a browser app to automatically label such sites. Also, users do not ordinarily notice an icon by its absence but only by its presence. Therefore the browser/app should detect the absence of the privacy icons to notify users they have entered a site where their privacy and usage restrictions could be violated. Primelife's "D. Dashboard" overcomes this problem by making a limited assessment of the current web page and creating an icon based upon factors such as the use of 3rd party cookies, the use of P3P, etc [15]. The Dashboard is available in the form of an extension that logs the user's HTTP traffic to a local database and provides a variety of queries for analyzing them.

3. PROPOSED APPROACH

This thesis will address the limitations of current privacy implementations and provide an infrastructure to build more privacy-aware systems on the web by deploying a web protocol called HTTPA. The data provider will present a set of usage restriction choices to the data consumer based on the credentials presented upon data access. The consumer will select the appropriate usage restriction(s) based on the intention(s) of the access and convey this to the provider. This agreement is logged by a trusted third party that is part of the 'Provenance Tracker Network' described in detail below. The consumer is responsible, and will be held accountable for, relaying the usage restrictions when transferring the data to somebody else or posting it to another server. In case something goes wrong i.e. the user misuses some information by violating a usage restriction either intentionally / unintentionally, or not transfer and / or tamper with the usage restrictions available as metadata with the data, it will be possible to construct an 'audit trail' to determine what happened. There will be smart clients on the browsers and smart agents on the servers that will facilitate these processes.

Unlike previous work on information reuse (i.e. DRM, Distributed Usage Control), there is no enforcement mechanism in this work. Rather, the smart clients will advise users on the proper usage of the data. There will be no prevention mechanisms, and if terms of use get violated, the owner of the content can figure out how they were violated and take appropriate action (i.e. request a takedown notice, or give proper attribution). Compared to P3P where the site conveys the privacy policy to the client, Mozilla Privacy Icons, and D. Dashboard where the browser or a browser plugin will determine the acceptable privacy and data usage policies based on the cues set by the site, in HTTPA both the data consumer and the provider participate in the selection of the usage restrictions. While there are some similarities of HTTPA with the architecture proposed in Project DReaM, HTTPA is not designed to manage usage restrictions on copyrighted material alone and it is designed to be applicable to any kind of content on the web. Also, unlike in Project DReaM, HTTPA does not rely on a centralized entity such as the anonymizing agent to route the fair use records through. In HTTPA, all the protocol components, including the provenance trackers, are designed to operate in a decentralized manner. In addition to that, as far as the author is aware of, none of the related work provides a mechanism for constructing a provenance trail built on open web standards.

As shown in Fig 1, HTTPA has three main entities: (1) Smart Clients on the web browsers and web servers that send and receive web data, (2) Verification Agents that vouch for the authenticity of the parties involved in the HTTPA transaction using the WebID protocol [16], and (3) The Provenance Tracker network that facilitates the logging and accountability checking processes. The key components that are used by these entities in HTTPA are as follows:

3.1 Authentication

Authentication is a crucial component in the protocol, not just for access control, but also to find the identity of the users who accessed and transferred resources should their owners claim that someone violated their usage restrictions on those resources. Since the web is a decentralized system, we require a global identity of the entities involved in a transaction. The WebID protocol [16] provides a robust mechanism for authentication in such a setting. An entity that wishes to access a resource using HTTP over TLS (Transport Layer Security) has to go through trusted entity called a *Verification Agent* that was agreed upon by both the data provider and the consumer. The Verification Agent performs authentication on the provided WebID credentials and determines if the data consumer can have access to a particular resource based on the access control policies set by the data provider. The browser based smart client will prove the possession of or access to a private key, whose corresponding public key is tightly bound to the WebID (i.e., a FOAF document) that is being authenticated. The private key is usually associated with an X.509 certificate on the user's computer, while the public key can be typically found on the FOAF profile.

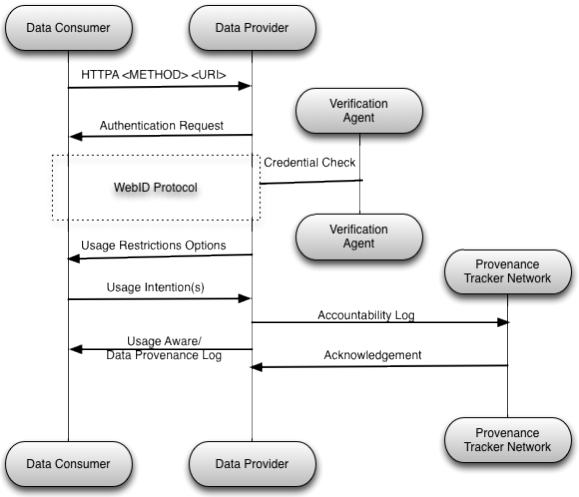

Figure 2: Data Creation HTTPA Method

3.2 Usage Restriction Language

Websites publish privacy policies that communicate planned data handling practices, such as rights of the data, intended purposes of collection, and third parties who may have access to the data collected from the users. Users also have complimentary usage restrictions for what their data can and cannot be used for. For the initial implementation, I used the RMP ontology [17]. This ontology allows specifying usage restrictions and intentions for terms such as 'No Cookies' (the server will not place any first-party or third-party cookies on the user's hard disk), 'No Commercial' (the owner of this data does not want the information on this profile used for commercial purposes), 'No Employment' (the owner of this data does not want the information on this profile used for employment purposes), etc.

The data provider sends the usage restriction options available with a resource to the client. These usage restriction terms are sent as comma separated URIs from the RMP ontology in an HTTP header called "UsageRestrictions". Depending on the policy set by the user, the smart client chooses the correct usage restriction(s) and conveys the acceptance of the usage restrictions to the server in an 'acknowledgment' message. If no such policy is set, the client prompts the user to select usage restriction(s) that best matches the intention of the data access from the values sent by the server. If the user does not select any usage restriction(s)–either by not sending an acknowledgement message back to the server, or not sending any subsequent requests to access the server without any intentions attached with the request–the server checks its' policy for the transfer of data available at the resource. Usually, this policy defaults to 'make no data transfer if the client does not acknowledge the usage restrictions', but the data provider can be flexible and just send the data rescinding the usage restrictions associated.

So far, we have only considered the usage restrictions with relation to a client-server architecture where the client downloads some content from the server. In the case where the client uploads some content to a server via methods such as POST or PUT, a similar usage restriction exchange takes place. However, unlike in the previous case, the server will not be dealing with case-by-case usage restriction selection. Rather, the entire process will be policy driven.

3.3 Provenance Trackers

Provenance Trackers are trusted third parties that log the transactions involving data transfer on the web. A log will be created in the provenance tracker network that includes all the information pertaining to the transaction (i.e. the sender, receiver, digest of the information, time of access, usage restrictions and intentions, etc). The logs will be kept encrypted, and will only be readable by HTTPA protocol components. The network of provenance trackers is used to construct a provenance trail of usage of a certain resource. This enables a data owner to figure out what had happened in case of any instance of misuses.

In the initial implementation, Provenance Trackers are deployed as an overlay network implemented using a Distributed Hash Table (DHT) on Planetlab [18]. The overlay network is trusted, in that the recognition as a 'Provenance Tracker Node' is regulated by a *super provenance tracker*. It also has a low churn rate, and the nodes have near perfect uptime. The main responsibilities of the provenance trackers are logging the transactions, and performing accountability checks. These tasks are described in detail below:

3.3.1 Logging

The Provenance Tracker Network creates an accountability log for every HTTPA transaction between a data provider and a data consumer. Accountability Logs have several characteristics: they are immutable except by protocol components, encrypted, secure, readable only by trusted parties involved in the HTTPA transaction, and have all the records pertaining to a particular data transfer and usage, such as: what data was accessed, the specified intent of access, and the agreed upon usage restrictions.

The key of the entry to the DHT is the hash of the URI that is subject of the HTTPA transaction. The rest of the information pertaining to the transaction is stored as the value. Provenance data is also incorporated into these logs. For example, if resource A was modified and resource B was created, the provenance tracker entry for B has a pointer to the provenance trail for resource A and vice versa.

3.3.2 Accountability Checking

If a user finds that her data was misused and / or the usage restrictions associated with it were violated, she can take recourse by producing a provenance trail with the help of the provenance tracker network through this feature. The provenance tracker network first verifies that the requestor of this information is indeed the owner of the resource. Then it performs several DHT lookups to create a trail of transactions involving this resource.

To illustrate this, let us consider the following example. Suppose a user, Alice, has uploaded a photo on a public photo sharing website with a usage restriction specifying that the photo could not be used for any *commercial* purposes. This restriction may be in the form of Creative Commons Attribution-NonCommercial 3.0 (CC BY-NC 3.0) license . An employee from a large advertising company, Bob, accessed that photo with the intention of using it for personal use. Bob's HTTPA-aware smart client confirmed with the website that the intention of accessing the photo was *non-commercial*, and that he would honor Alice's usage restriction. Bob mod-

ified Alice's photo slightly, and reposted it in an internal company website along with the usage restrictions for *non-commercial* use set by Alice. Carol, another employee of Bob's advertising company sees this picture, and uses it in an advertisement for the company. Carol does not use an HTTPA-aware client, thus there were no warnings as to the proper usage of the photo. Few weeks later, Alice found out that her photo was used in an online advertisement, and she is interested in knowing how her usage restrictions were violated.

With HTTPA, Alice can request the provenance tracker network to construct an 'audit trail' for her by giving the URI of her photo on the photo sharing site that has been inappropriately used for a commercial purpose. The provenance tracker network verifies that Alice owns the resource and looks up accountability logs within the provenance tracker network. It verifies that Bob had agreed to the original usage restriction that Alice had set, he had made some modifications to the photo, reposted elsewhere with Alice's original usage restrictions, and then Carol had reused it in a manner that violates Alice's original terms. Alice can now request the provenance trackers to send Carol a *signed* proof detailing the violations, and ask for a takedown, since the advertisement violated her terms of use of her original photo.

3.4 HTTPA Smart Clients

In order for the protocol to work seamlessly there is a need to develop 'Smart Clients' that facilitate the usage restriction transfer, accountability checks and proper data usage advise upon reuse. The goal here is to minimize the burden to the user as much as possible.

User Agents have smart clients (e.g. browser extensions), and the servers run services or have modules that honor and facilitate the protocol. Smart clients facilitate obtaining the information by communicating the intended use of the data accessed. If the usage restrictions match the intensions, the said information will flow from the data provider to the data consumer. A simplified sequence diagram illustrating an HTTPA transaction for a method such as GET, PUT, POST that can be used to 'create' data is given in Fig 2.

The smart clients also function as interpreters and creators of logs. Currently read-only logs on web servers are used for debugging problems on the server or to generate statistics about how websites are accessed. In HTTPA, the following logs are maintained by the smart clients:

- **Usage-Aware Logs**: These are sent to the data consumer's smart client by the provenance trackers. Based on the usage restrictions set, the smart client can warn the user, if the user's actions violate any of the usage restrictions that were agreed upon.

- **Data Provenance Logs**: These are created by the smart client on the data consumer's end. The smart client helps the user in creating a remix from several different resources gathered from the web, and during this process, it constructs a provenance trail with the URIs of the HTTP resources used in the remix with the understanding that further transfers of the same data may not use HTTPA.

4. METHODOLOGY

The methodology adopted in evaluating this research is to examine how effective the protocol is, in terms of protecting user privacy and how it enables the users to adhere to and preserve the usage restrictions associated with web content. The protocol developed will be agnostic to the specific use case instances and will provide an ecosystem where we can expect the web users to utilize the system in an accountable manner in the absence of enforcement.

Since people like to post things–from news articles to photos and videos–on their favorite social networking site to share with their friends, I believe development of a social networking site/app that implements HTTPA, and using that to study how the data is being transferred will give a good indicator as to the success of this work. Unlike mainstream social networking sites such as Facebook where the 'share' buttons provide a mechanism to keep a count on the number of shares that were done using the Facebook login and find the original poster [19], this user study will allow finding reuse across website boundaries.

5. RESULTS

As a preamble to this work, I have analyzed the extent to which web users violate creative commons attribution licenses on Flickr images when reusing them in their blog posts [20]. I found license violations ranging from 78-94% on three samples of blog websites indexed by Technorati that are linking about 500 Flickr images. Further, I conducted a survey study on number of user-generated content communities that suggested that most reusers of content do not honor the usage restrictions or are not aware of them [21]. I have explored providing a more flexible negotiation of usage restrictions with the intentions of data access–sent as HTTP headers–where the data consumers and the data providers engage in a dialogue about what usage restrictions to agree to [22]. The implementation has two components: A Django python implementation lets a website developer set the usage restrictions to content on the site, either per resource or per resource group or for the entire site, and a python client that selects the usage restrictions based on a policy set. The focus now is to develop a user friendly browser extension that handles that. I have also deployed an overlay network on Planetlab that implements the functionality of the provenance trackers.

6. FUTURE WORK AND CONCLUSIONS

There are many scalability issues such as disk space for storing the logs of all the HTTPA transactions, and latency arising from the extra HTTPA transactions compared to the usual HTTP transactions. The issue of latency can be relieved to some extent by using smart client based cache mechanisms. The usage restriction language used for HTTPA should cover terms from the P3P Preference Exchange Language (APPEL) [23] and should be able to handle other privacy preference languages such as Privacy Preference Ontology (PPO) [24]. I would also like to make the usage restrictions semantically consistent, so that they would not be interpreted and represented differently by different user agents. Accountability checking for media that has a high out-degree (i.e. popular news item that got shared several thousand times, and got changed during the processes of sharing) will pose a challenge to the accountability checking aspect of the protocol, as the provenance trails created will have a lot of pruning to do. Finally, there are many challenges in terms of the adoptability of this technology by major websites, and encouraging users to specify their usage restrictions/intentions

up front. One way to tackle this would be to incorporate a payment mechanism to reward the distributors of data items that honor the protocol. Project VRM, Tipsy and Emancipay projects will be used for experimentation of this idea [25].

This work will address the limitations of current privacy work and provide an infrastructure to build more privacy-aware systems on the web. Government organizations, academic institutions, and businesses are expected to be the early adopters of this accountable web protocol with usage restriction management within their networks. On the longer run, in a similar vein in which the growth of e-commerce websites led to the massive adoption of HTTPS, I envision that HTTPA will be accepted by the larger web community, as privacy problems slowly cripple the growth of the web.

7. REFERENCES

[1] Ching man Au Yeung, Ilaria Liccardi, Kanghao Lu, Oshani Seneviratne, and Tim Berners-Lee, "Decentralization: The Future of Online Social Networking," in *W3C Mobile Social Network Workshop*, September 2008.

[2] Wanhong Xu, Xi Zhou, and Lei Li, "Inferring privacy information via social relations," in *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*, april 2008, pp. 525 –530.

[3] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James Hendler, and Gerald Jay Sussman, "Information Accountability," *Communications of the ACM*, vol. 51, pp. 82–87, June 2008.

[4] Ronald Leenes, "Context is everything: Sociality and Privacy in Online Social Network Sites," *Privacy and Identity, IFIP AICT 320*, pp. 48–65, 2010.

[5] Picot, Arnold and Fiedler, Marina, "Impacts of DRM on Internet Based Innovation," in *Digital Rights Management*, Becker, Eberhard and Buhse, Willms and GÃijnnewig, Dirk and Rump, Niels, Ed. 2003, vol. 2770 of *Lecture Notes in Computer Science*, pp. 288–300, Springer Berlin / Heidelberg.

[6] Lorrie Faith Cranor, "Web privacy with Platform for Privacy Preferences," *Oreilly Books*, Jan 2002.

[7] Electronic Privacy Information Center, "Pretty Poor Privacy: An Assessment of P3P and Internet Privacy," June 2000.

[8] P. Kumari, A. Pretschner, J. Peschla, , and J.-M. Kuhn, "Distributed data usage control for web applications: a social network implementation.," in *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, 2011, pp. 85–96.

[9] Susan Landau, "Support for Fair Use with Project DReaM," *Sun Microsystems Laboratories*, vol. Version 1.0 Rev A, April 2008.

[10] Jorge R. Cuellar, John B. Morris, Deirdre K. Mulligan, Jon Peterson, and James M. Polk, "Geopriv Requirements. Internet RFC 3693," .

[11] Andrei Popescu, "Geolocation API Specification," .

[12] Nick Doty and Erik Wilde, "Geolocation privacy and application platforms," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, New York, NY, USA, 2010, SPRINGL '10, pp. 65–69, ACM.

[13] E Wilde, "Simple policy negotiation for location disclosure," *w3.org*.

[14] Aza Raskin and Arun Ranganathan, "Privacy: A Pictographic Approach," *W3C Workshop on Privacy for Advanced Web APIs*, 2010.

[15] Primelife, "D. Dashboard," `http://www.primelife.eu/results/opensource/76-dashboard`.

[16] Manu Sporny, Toby Inkster, Henry Story, Bruno Harbulot, and Reto Bachmann-Gmur, "Web Identification and Discovery," *W3C Editor's Draft*, 2011.

[17] Ted Kang and Lalana Kagal, "Enabling Privacy-awareness in Social Networks," in *Intelligent Information Privacy Management Symposium at the AAAI Spring Symposium 2010*, March 2010.

[18] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman, "Planetlab: an overlay testbed for broad-coverage services," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 3–12, July 2003.

[19] Mark Kinsey, "Keeping Count of Sharing Across the Web," *The Facebook Blog*, 2009.

[20] Oshani Seneviratne, Lalana Kagal, and Tim Berners-Lee, "Policy-Aware Content Reuse on the Web," in *ISWC 2009*, 2009, pp. 553–568.

[21] Oshani Seneviratne and Andres Monroy-Hernandez, "Remix culture on the web: A survey of content reuse on different User-Generated content websites," in *Web Science Conference at World Wide Web Conference 2010*, April 2010.

[22] Oshani Seneviratne and Lalana Kagal, "Addressing Data Reuse Issues at the Protocol Level," in *POLICY 2011, IEEE International Symposium on Policies for Distributed Systems and Networks*, 2011, pp. 141–144.

[23] Marc Langheinrich and Lorrie Cranor and Massimo Marchiori, "APPEL: A P3P Preference Exchange Language," *W3C Working Draft*, 2002.

[24] Owen Sacco and Alexandre Passant, "A Privacy Preference Ontology (PPO) for Linked Data," in *Linked Data on the Web Workshop at the World Wide Web Conference 2011*, April 2011.

[25] Doc Searls, "Emancipay: A Relationship Management and Voluntary Payment Framework," *Harvard Law Blog*, 2010.

[26] Berners-Lee, Timothy J, "Information Management: A proposal – oai:cds.cern.ch:369245," Tech. Rep. CERN-DD-89-001-OC, CERN, Geneva, Mar 1989.

[27] Catherine Dwyer, Starr Hiltz, and Katia Passerini, "Trust and Privacy Concern Within Social Networking Sites: A Comparison of Facebook and MySpace," in *Proceedings of the Thirteenth Americas Conference on Information Systems, Keystone, Colorado*, 2007.

[28] danah m. boyd and Nicole B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[29] Prema Nakra, "Consumer privacy rights: CPR and the age of the Internet," *Management Decision*, vol. 39, no. 4, pp. 272–279, 2001.

[30] Mozilla, "Privacy Icons," `https://wiki.mozilla.org/Drumbeat/Challenges/Privacy_Icons`.