The Feasibility of Stylometry as a Service

Josh Datko and **Joseph Heenan**
Drexel University
CS680: Privacy in the Electronic Society

Introduction

Maintaining anonymity online is increasingly difficult. While networks like Tor (**?**) will protect an Internet Protocol (IP) address at the network layer, it does not protect the content delivered over Tor. For example, suppose an author desires to provide controversial political commentary but wants to remain anonymous. In this case, she can protect her IP address with Tor and use end-to-end encryption to protect her message en-route, but she could be discovered through stylometry.

Stylometry, a form of authorship recognition, has been found to be extremely effective especially even at Internet-scale (**?**). Thus for those seeking to publish anonymous writings, not only can the delivery path reveal them, but their content can as well. Adversarial stylometry is the active process of manipulating writing to confuse the identity of the author. A set of tools, JStylo and Anonymouth (**?**), have been produced by the Privacy, Security, and Automation Lab (PSAL) at Drexel University to assist authors in anonymize their writing.

Currently the JStylo and Anonymouth integrated open-source project (JSAN) is available to download and can be installed on a machine that runs the Java Virtual Machine (JVM). While JSAN has been successfully demonstrated to be effective in assisting authors with adversarial stylometry (**?**), the constraint of downloading and installing the software on a machine powerful enough to run machine learning algorithms on data sets, limits their adoptability. We propose increase the availability and usability of JStylo and Anonymouth by offering them as a web service.

Migrating to a web service would provide JSAN with several benefits. This web service would provide adversarial stylometry capabilities to embedded devices like tablets or smart-phones which otherwise could not run the software natively. Since there would be no installation, besides a typical web-browser, a web-service would be better suited for time-critical situations where an author must quickly anonymize her document.

An example use-case is an author with a time sensitive document that she needs to release. Because of the sensitive nature, she wishes to apply adversarial stylometry to her document. In this case, a computer with decent enough performance to natively run machine learning techniques may not be available. However, a web service is nearly ubiquitous.

Expanding Stylometric Research

A stylometric web service also expands research opportunities. This service, offered through a modern architecture, would allow researchers to easily perform experiments. An easily accessible tool allows for rapid verification of research results and greater community engagement.

In addition to convenient access, a web service can provide additional diversity in stylometric experimentation. For example, this web service supports multiple authors and provides a better tool for real-time, multiple author stylometry than a stand-alone application.

An effective technique in adversarial stylometry is imitation of another author's writing style. However, since many are not trained author imitators, this technique requires significant cognitive effort. Perhaps a more effective technique, would be to have multiple authors re-write the target work in their natural style. While there has been numerous studies into single author stylometric analysis, multiple authorship is more difficult. This web services provides a platform for multiple authorship research.

Challenges in a Web Service

A privacy enhancing technology that is designed to exist as a web service faces different challenges than an application. Attack vectors open up on the service from active denial-of-service attacks to passive sniffing. A privacy web service must take great care to ensure that service does not compromise its user's privacy. In Section **??** we provide our architecture and analysis the security and privacy.

1 Design

Anonymouth Architecture Review

The architecture of Anonymouth is that of a conventional desktop application. It uses a layered architecture with essentially two main components: the frontend Graphical User Interface (GUI) and the backend, consisting of the Weka machine learning toolkit and Jstylo. Jystlo is PSAL's stylometric analysis software. Since Anonymouth was designed as

a monolithic application, there are direct dependencies between the layers without the use of an adaptation layer.

This design is sufficient for Anonymouth in its current form. As a desktop appliccation, the user can anonymize her and the document locally without any additional attack vectors.

Requirements for Web Service

The architecture for a web service imposes different requirements on an application. First and foremost, since Anonymouth is a privacy enhancing technology, the web service *must* not degrade the privacy protection of the application. This is a challenge when migrating from a desktop to web service since the desktop application does not use any networked communicaiton. However, there is little benefit to using Anonymouth on the web if it compromises one's privacy.

Low-latency. The web service should be near-real-time, i.e. low-latency. We don't specify a time requirement but the response should be in the order of milliseconds to seconds. A high latency system would not be well suited for a web service. An example high latency system could simply entail sending an encrypted email to a third party who uses Anonymouth locally and returns the encrypted result. While this high-latency system seems trivial, note that a naive implementation will reduce the user's privacy. For example, email is very subject to traffic analysis (CITE) and encrypted email is easy to detect (CITE), therefore this act alone would raise suspiction. If this third party was offering a high-latency anonymiziation service for numerous people, the network graph would be easy to detect.

Architecture Qualities

There are several qualities of a system architecture that are considers foundational (CITE lem bass). These are: availability, interoperability, modifiability, performance, security, testability, and usability. In additional we add privacy as an architecture quality in our review.

Availability is the measure of the measure of the accesibility and realibility of the system, e.g. how well does it handle errors. Interoperability refers to the degree with which a system can interact with its components or other systems. Modifiability is important for large systems so that they can be easily maintained and scale. The response of the system to input is its performance. Security is the review of a system's ability to protect sensitive data and prevent unautorized and undesired behaviour. Testability is an important factor not only in development but also in a deployed system to ensure that the system is working as designed. Usability is notoriously bad for security software (CITE How Johnny) and refers to the interaction design. Lastly, by privacy we mean how well a user's anonymitity is protected.

In the following sections, we provide an analysis of each quality and define the requirements for a stylometric web service.

Availability. If the user can not access the web service, then the user is in the same situation as having a device that can not run Anonymouth locally. The entry point to the web service requires a web server. Realistically, this service will never see the traffic that FaceBook or Twitter experiences, but the web server should be handle thousands of requests per day.

A common attack to block availability is a Denial-of-Service (DoS) attack. A DoS attack is akin to receiving a momentarily burst in traffic, i.e. the "Slashdot effect." The difference lies in the motivation for the sudden attention. Thus, a service that is resistant to DoS attacks also scales well to large second-derivative increases in traffic.

There are many implemenation details to achieve availability that are beyond the scope of this paper. We attest that a stylometric service needs to be available and should consider the effects of sudden traffic increases.

2 Security Analysis

3 Privacy Analysis

4 Implementation Details

5 Comparison to Stand-alone Application

6 Evaluation

7 Conclusion

8 Proposal stuff — will be removed

Approach

We propose to build a stylometric service around JStylo and Anonymouth. However, this project is more than just porting JSAN to the web. With a migration to a web service from locally running software, the anonymity attack vector changes significantly. We intend to design and evaluate privacy enhancing techniques for a stylometric web service.

Design

Our design goal is to increase both the anonymity and availability of JSAN through this service. In order to increase the anonymity, we will need to protect the service from traffic analysis, which we intend to do by running a Tor hidden service. To improve the anonymity and effectiveness of JSAN, we intend to incorporate a third-party re-writer and in the future, a document mixer. The document mixer will randomize the target document, potentially with other target documents, and present a re-writer the mixed document. The re-writer will be able to use the Anonymouth-like interface to anonymize the document without being able to infer the author. Also, the mixing will help obfuscate the contents of the document as well. However, for this initial attempt, we will focus on the third-party anonymization service.

Although in this initial effort we do not intent to conduct usability testing, we will consider incentive options for re-writers. Like other anonymity networks, we believe that anonymity is improved with a diverse user base. The challenge is building the user base. While we do not intend to conduct user testing for this initial effort, we will consider various incentive options like actual payment to gamification.

Implementation

Our initial idea is to host this service on Amazon Web Services (AWS). For the user interface, we are planning to implement the front end in AngularJS. Since the existing code is Java, migrating to the Play Framework seems like a natural progression. The text editor functionality will be converted to Firepad, an open source collaborative editor. While real-time collaboration is not necessarily needed, we think that an "Anonymouth" user, an Artificial Agent, would assist the re-writer in edits.

Related Work

This proposal extends the work from Drexel's PSAL, who have made several contributions in this field. In (**?**), the authors present the Anonymouth framework for anonymizing writing. (**?**) confirmed adversarial stylometry is very effective at obfuscated authorship, but also presented techniques to hide the prescience adversarial stylometry. Lastly, (**?**) presented successful adversarial stylometry techniques through obfuscation, imitation, and translation and produced two corpora of texts.

Tor is one of the most successful anonymity services in operation (**?**). We intend to protect the anonymity of users by offering JSAN over a Tor hidden service however, there has been recent research de-anonymizing popular hidden services (**?**). We will provide a discussion on the design and experience of enabling a hidden service, but we will focus on this aspect.

Novelty

There are two significant contributions as a result of this work. The first is a stylometric web service. This will advance the usability and capabilities of JSAN and provide a framework for more expansive user testing in the future. The second contribution is that this is the first framework for multi-author adversarial stylometry. While single author stylometry has been recently studied, multi-author documents have not received significant attention.

Evaluation

This effort is intended to produce several artifacts. The first is an evaluation of the privacy architecture of a cloud web service. Here we intend to describe the challenges and present and evaluate a privacy-by-architecture system. Second, we will provide a survey on various incentive approaches and suggest a possible method. Lastly, we will provide a prototype web service and evaluate it. Specifically we will compare the web service to the standalone tools and evaluate it for accuracy and usability. Additionally, we will experiment with the third party anonymization by re-writing each other's writing samples. We will provide an evaluation on the perceived difficultly of re-writing one's own work to that in the mixed system.

Milestones

Table 1 contains a proposed timeline for completing this project. This work will be split among two developers and there are several modules that can be independently developed. In order to re-architect JSAN, we first need to become familiar with the existing software. While this software will be running as a service, we do not intend to release it to the general public in this time-frame. We will conduct the evaluation with perhaps a very limited subset of other students.

Milestone	Date
Document initial design	October 28
Finish porting JSAN to a web service	November 4
Add the Firebase editro	November 11
Add the third party feature	November 18
Conduct evaluation	November 25
Provide Paper and Presentation	December 2

Table 1: Project Milestones

Future Work

The scope of this project is designed to completed in one Drexel quarter, however we have several ideas for continued work. With a web service, we can offer real-time adversarial stylometry. Assume Alice, who has a blog, wants to anonymize her next blog post. She would connect to our web service, submit the URL for her blog and then write her next post with adversarial stylometry. This real-time data crawling at this scale is simply not feasible with stand alone software.

Now that a third party will be re-writing documents, we feel that it is important to obfuscate the content from the re-writer. One idea, is to mix the document, perhaps with other documents, prior to re-writing. The re-writer would be presented with an otherwise nonsensical document and would focus on re-write each sentence in her native style. The combined document, would contain n author styles, where n is the number of re-writing authors.

Lastly, for long term adversarial projects, we are curious on how often one has to re-train the model. We imagine that a web service could help monitor the change of a stylometric fingerprint over time.