Privacy-Preserving Symptoms-to-Disease Mapping on Smartphones

Michael Holmes[1], Jerald Liu[2], Huanying Gu[3], and Paolo Gasti[3]

[1]Computing Sciences Department, University of Scranton
[2]Department of Computer Science, Johns Hopkins University
[3]Department of Computer Science, New York Institute of Technology

Abstract. Today, several smartphone applications can be used to map user symptoms to the corresponding diseases. However, these applications have several shortcomings, including *limited usability* and *lack of privacy*. In particular, user queries are satisfied by remote servers, which learn all symptoms associated with each request – and therefore, with each user. In this paper we present the design of a privacy-preserving framework and mobile application for mapping symptoms to diseases. The goal of our framework is to suggest a list of diseases, based on user symptoms, without revealing sensitive information to the service provider. Although our work is preliminary, it shows that complex medical applications can be implemented with privacy. In particular, sophisticated functionalities such as NLP and symptoms-to-diseases mapping can be performed without disclosing any information about the user.

Keywords: Mobile Medical Applications, Privacy, SNOMED-CT, OpenNLP

1 Introduction

Smartphones are always-on always-connected general purpose computing devices that allow users to perform a wide variety of tasks. Given their ease of use and portability, they have the ability to meet diverse user needs. Among many applications currently available on mobile platforms, medical apps have become relatively popular in recent years.

Smartphone medical applications allow provide users with a list of diseases related to users' symptoms. Existing applications, such as HealthTap [6] and iTriage [7], are inconvenient to use and do not protect user privacy. The user interfaces of these applications is usually built as a long list of items, among which users must select appropriate symptoms. This is not only cumbersome, but it also requires knowledge of the exact symptom names, and provides limited usability to patients with disabilities. Additionally, since symptoms-to-disease matching is performed on servers owned by the application developer (henceforth, service provider), these applications disclose a large amount of medical information about patients.

A recent study shows that 87% of the United States population can be identified with just three simple pieces of information: zip-code, gender, and birthdate [1]. It is conceivable that a malicious party could extract age, gender, and location information based on the queried symptoms and the IP address of client, thus learning the identity of its users as well as medical conditions they may have.

Privacy can also be a concern for service providers: to limit their liabilities, service providers may be unwilling to process sensitive information from users.

The Food Drug Administration (FDA) does not regulate applications like HealthTap and iTriage as they are categorized as "lifestyle applications". The FDA only enforces privacy guidelines in applications that deal with critical conditions, drug delivery and monitoring [8]. Because of this, developers must be proactive in implementing a secure system in medical mobile applications to safeguard clients' information.

Contributions. In this paper we design a privacy-preserving framework for medical applications. Our framework includes a mobile application and server backend, and addresses privacy and usability issues of current symptom lookup applications. To do this, we rely on Natural Language Processing (NLP) techniques and privacy-preserving protocols to implement secure free-text multi-symptom lookup.

As a proof of concept, we developed an Android application and the corresponding backend. The Android application handles NLP on the client, ensuring that the symptoms do not need to be simplified by an outside service provider.

2 Related Work

In this section we briefly highlight existing related mobile applications, and introduce the tools used in our framework.

Existing Applications. HealthTap [6] and iTriage [7] are prominent examples of medical mobile applications that provide useful health-related information based on user symptoms.

HealthTap is a tablet and smartphone application for iOS and Android that gives a symptom to condition recommendations. It also features a recommendation system for doctors based on diagnosis or disease. However it stores a significant amount of information on its servers with no privacy guarantees [6].

iTriage is another symptom-to-diagnosis tablet and smartphone application for iOS and Android. It does not recommend doctors based on diagnosis but does allow users to search for doctors close to the user's location. Similar to HealthTap, iTriage stores a significant amount of personal information on the developer's servers. Moreover, iTriage's EULA states that the Company has the right to share user information with third parties [7].

Security Primitives. Private Information Retrieval (PIR) is a protocol that allows a client to query a database server without the server learning the content of the user query, or the returned results [2, 3]. For efficiency reasons, we relied on the PIR protocol of Devet, Goldberg and Heninger [2]. The protocol is based on multiple non-colluding servers, which store an identical copy of the database. The protocol is three orders of magnitude faster [2] than previous single-server PIR [3].

To anonymize a client's IP address, all client input is forwarded through the Tor anonymizing network [11] to the group of PIR servers.

Text Processing. OpenNLP is a NLP library, which provides strong stemmer and parts of speech functionality [10]. Although OpenNLP is just a common English processing library, previous work has shown that OpenNLP is a powerful tool that can be used in a domain with very specific terminology [4]. Therefore we have selected OpenNLP to handle the NLP of our pre-processor module.

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms. We rely on WordNet to look up English synonyms before searching the medical database [9].

Medical Ontology. SNOMED-CT is a comprehensive medical terminology database that contains all known medical symptoms and their related diseases. The synonyms for all terms are provided in the SNOMED-CT as well. The SNOMED-CT dataset can easily be imported into any relational database, such as MySQL. We choose to use SNOMED-CT as our primary medical term database because it has a medical concept coverage rate of 98.5% for standardized medical terminology [5].

3 Design

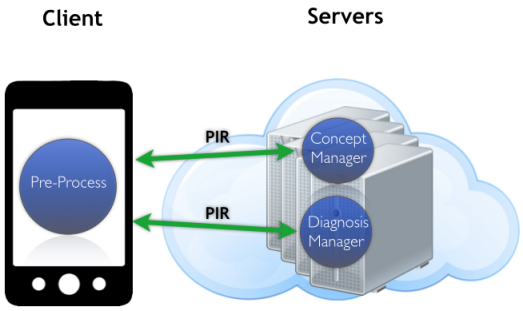

Fig. 1. High-level overview of our framework. Each green arrow represents a separate instance of the PIR protocol.

Our framework takes user symptoms as input in free-text form, and returns a list of likely diseases, corresponding to the symptoms. It prevents the service provider from learning any information about the user, including the list of symptoms and geographical location. Privacy is obtained by allowing client and service provider interaction only through PIR.

When a user submits symptoms as free-text, the smartphone pre-processes the input, removing any excess words generating related terms to broaden the domain of possible queries. The client creates "shares"[2] of pre-processed text to send to all the servers using PIR.

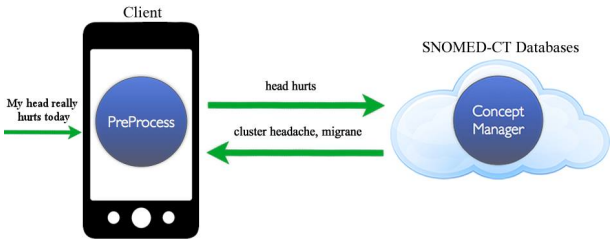

Fig. 2. Client sends shares of pre-processed text to servers where the Concept Manager computes and returns SNOMED-CT concepts

Each server takes its respective share and runs it on its local SNOMED-CT database, resulting in a set of symptoms related to client input. Each server then returns the results to the client. The client reconstructs the query result from the individual shares.

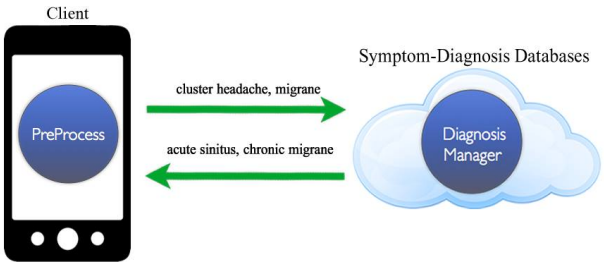

Fig. 3. Client creates and sends shares of the returned SNOMED-CT concepts back to the servers for handling symptom-diagnosis mapping in the Diagnosis Manager

The client sends the Concept Managers' results back to the servers that use Diagnosis Managers in order to map symptoms to diagnoses by using a Symptom-Diagnosis database.

3.1 Pre-Process

Fig. 4. Text pre-processing relies on WordNet and OpenNLP

The Pre-Processor module is designed to filter out irrelevant words from sentences as well as find the stem and possible synonyms for extracted words. The Pre-Processor is based entirely on the client and uses OpenNLP's part-of-speech (POS) tagger to remove unnecessary words from phrases. For example: "Today my throat is very sore" will become "throat sore".

Next the pre-processor can use the stemmer to find the root of each word in the simplified sentence. Once the root has been found, the preprocessor searches for synonyms for each word using the WordNet synonym database. This list of simplified words and their synonyms are then sent to the server diagnosis or SNOMED database. Unlike the medical databases which can be hundreds of megabytes or even gigabytes in size, we are able to use OpenNLP and WordNet on the client because of the small file sizes associated with both frameworks, with the former being approximately 22 MB [10] and the latter being approximately 40 MB [9].

3.2 Diagnosis and concept Managers

The Diagnosis Manager (DiM) module is designed to map symptom to diseases. DiM only needs to query the disease database to return a list of possible diagnoses. If the DiM returns no results, the Concept Manager (CM) module will query the SNOMED-CT database and return a list of synonyms for symptoms.

Both DiM and CM modules rely on PIR for protecting user privacy.

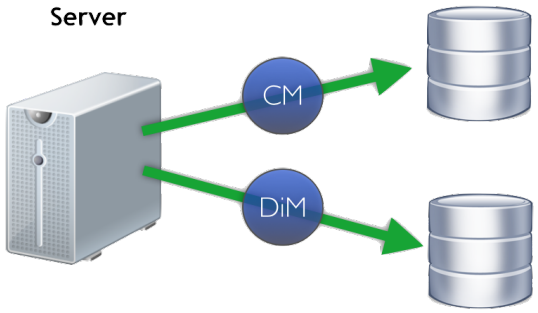

Fig. 5. The Concept Manager (CM) uses SNOMED-CT to map user input to related medical terms. The Diagnosis Manager (DiM) relates symptoms to diagnoses using a symptom-disease database.

4 Conclusion

User privacy and usability are two fundamental requirements for medical applications. Our framework allows users to map symptoms to diseases using on-line services, without revealing any of their symptoms to the service provider. Through the use of Natural Language Processing techniques, our framework allows users to enter their symptoms via free text input.

Our framework does not dictate the use of a specific input device. We are in fact in the process of extending our framework to allow voice input, and determine the impact on usability. Additionally, we are planning to investigate assistive input technique to allow users with disabilities to use our framework.

We are also planning to add a further module that would allow users to look up appropriate medical providers, given a list of symptoms.

Acknowledgment

This project is supported by the National Science Foundation Grant No. 1263283 and New York Institute of Technology.

References

1. Latanya Sweeney. Uniqueness of Simple Demographics in the U.S. Population. LIDAP-WP4 Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA. 2000
2. Casey Devet, Ian Goldberg, Nadia Heninger. Optimally Robust Private Information Retrieval. Security'12 Proceedings of the 21st USENIX Conference on Security Symposium, Berkley, CA. 2012
3. I. Goldberg. Improving the robustness of private information retrieval. In 2007 IEEE Symposium on Security and Privacy, pages 131-148, 2007.
4. Ekaterina Buyko, Joachim Wermter, Michael Poprat, Udo Hahn. Automatically adapting an NLP core engine to the biology domain. In Proceedings of the ISMB 2006 "Joint Linking Literature, Information and Knowledge for Biology and the 9th Bio-Ontologies Meeting, pages 65-68, 2006.
5. Wasserman H, Wang J. An applied evaluation of SNOMED-CT as a clinical vocabulary for the computerized diagnosis and problem list. AMIA Annual Symposium proceedings / AMIA Symposium, pages 699-703, 2003
6. HealthTap. Computer software. Apple App Store. Vers. 4.6. HealthTap Health Corp Inc, 02 July 2013. Web. 5 Aug. 2013.
7. iTriage. Computer software. Apple App Store. Vers. 5.2.Healthagen LLC, 30 July 2013. Web. 5 Aug. 2013.
8. Galen Gruman. The Puzzle of Delivering Medical applications and Devices. http://www.infoworld.com/d/consumerization-of-it/the-puzzle-of-delivering-medical-applications-and-devices-215345, 2013. Accessed August 2013.
9. "WordNet." About WordNet. Princeton University, 27 Dec. 2012. Web. 06 Aug. 2013. http://wordnet.princeton.edu/.
10. "Welcome to Apache OpenNLP." Apache OpenNLP. The Apache Software Foundation, 2010. Web. 06 Aug. 2013. http://opennlp.apache.org/.
11. "The Tor Project." Web. 06 Aug. 2013. http://www.torproject.org/.