Preventing Data Leakage during Web Service Accesses

by

Frank Yi-Fei Wang

B.S., Stanford University (2012)
S.M., Massachusetts Institute of Technology (2016)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2018

Author .....
Department of Electrical Engineering and Computer Science
May 21, 2018

Certified by .....
Nickolai Zeldovich
Professor
Thesis Supervisor

Certified by .....
James Mickens
Associate Professor
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Preventing Data Leakage during Web Service Accesses

by

Frank Yi-Fei Wang

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

Web services, like Google, Facebook, and Dropbox, are a regular part of users' lives. As a form of payment, these services collect, store, and analyze user data. Even accessing these web services can leak a substantial amount of data.

This dissertation presents two practical, secure systems, Veil and Splinter, that prevents some of this data leakage. Veil protects against client-side information leakage from the browser by allowing web page developers to enforce stronger private browsing semantics without browser support. Splinter protects sensitive information present in a users' query on cleartext datasets in a practical manner by leveraging a recent cryptographic primitive, Function Secret Sharing (FSS).

Thesis Supervisor: Nickolai Zeldovich
Title: Professor

Thesis Supervisor: James Mickens
Title: Associate Professor

Acknowledgments

More acknowledgments here.

⋆ ⋆ ⋆

The dissertation incorporates and extends work published in the following papers:

Frank Wang, James Mickens, and Nickolai Zeldovich. Veil: Private Browsing Semantics Without Browser-side Assistance. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2018.

Frank Wang, Catherine Yun, Shafi Goldwasser, Vinod Vaikuntanathan, and Matei Zaharia. Splinter: Practical Private Queries on Public Data. In *Proceedings of Networked Systems Design and Implementation (NSDI)*, 2017.

Contents

Figures and tables

ONE

Introduction

This dissertation presents two practical, secure systems, Veil and Splinter, which protect against certain data leakage that happens when a user accesses a web service. The rest of this chapter will motivate this problem and briefly describe Veil and Splinter.

1.1 Motivation

1.2 Our systems

To address the problems above, we develop two systems, Veil and Splinter. Veil prevents client-side browser leakage, and Splinter prevents leakage of sensitive user information contained in queries to web applications.

1.2.1 Veil: Private Browsing Semantics without Browser-side Assistance

All popular web browsers offer a "private browsing mode." After a private session terminates, the browser is supposed to remove client-side evidence that the session occurred. Unfortunately, browsers still leak information through the file system, the browser cache, the DNS cache, and on-disk reflections of RAM such as the swap file.

Veil is a new deployment framework that allows web developers to prevent these information leaks, or at least reduce their likelihood. Veil leverages the fact that, even though developers do not control the client-side browser implementation, developers do control 1) the content that is sent to those browsers, and 2) the servers which deliver that content. Veil web sites collectively store their content on Veil's *blinding servers* instead of on individual, site-specific servers. To publish a new page, developers pass their HTML, CSS, and JavaScript files to Veil's compiler; the compiler transforms the URLs in the content so that, when the page loads on a user's browser, URLs are derived from a secret user key. The blinding service and the Veil page exchange encrypted data that is also protected by the user's key. The result is that Veil pages can safely store encrypted content in the browser cache; furthermore, the URLs exposed to system interfaces like

the DNS cache are unintelligible to attackers who do not possess the user's key. To protect against post-session inspection of swap file artifacts, Veil uses heap walking (which minimizes the likelihood that secret data is paged out), content mutation (which garbles in-memory artifacts if they do get swapped out), and DOM hiding (which prevents the browser from learning site-specific HTML, CSS, and JavaScript content in the first place). Veil pages load on unmodified commodity browsers, allowing developers to provide stronger semantics for private browsing without forcing users to install or reconfigure their machines. Veil provides these guarantees even if the user does not visit a page using a browser's native privacy mode; indeed, Veil's protections are *stronger* than what the browser alone can provide.

1.2.2 Splinter: Practical, Private Web Application Queries

Many online services let users query datasets such as maps, flight prices, patents, and medical information. The datasets themselves do not contain sensitive information, but unfortunately, users' queries on these datasets reveal highly sensitive information that can compromise users' privacy. This paper presents Splinter, a system that protects users' queries and scales to realistic applications. A user splits her query into multiple parts and sends each part to a different provider that holds a copy of the data. As long as any one of the providers is honest and does not collude with the others, the providers cannot determine the query. Splinter uses and extends a new cryptographic primitive called Function Secret Sharing (FSS) that makes it up to an order of magnitude more efficient than prior systems based on Private Information Retrieval and garbled circuits. We develop protocols extending FSS to new types of queries, such as MAX and TOPK queries. We also provide an optimized implementation of FSS using AES-NI instructions and multicores. Splinter achieves end-to-end latencies below 1.6 seconds for realistic workloads including a Yelp clone, flight search, and map routing.

1.2.3 Dissertation Roadmap

The dissertation will be organization like the following: Chapter 2 will motivate and describe Veil. Chapter 3 will present Splinter. Finally, Chapter 4 will describe future work.

Two

Veil: Private Browsing Semantics Without Browser-side Assistance

2.1 Motivation

THREE

Splinter: Practical, Private Web Application Queries

3.1 Motivation

FOUR

Conclusion and Future Work

Bibliography