Joseph Near
Research Statement

My goal is to build programming systems (programming languages, static analyses, and runtime systems) that enable provable security and privacy. A key component of my research is a focus on real-world needs, which I draw from industrial collaborations and studies of open source projects. These requirements often point out significant barriers to practical use, which represent important open research challenges. My work has made both theoretical and systems contributions to address these challenges and break down the barriers to wider adoption.

Motivation and Overview

Advances in big data and artificial intelligence have brought a range of revolutionary changes, but as recent data breaches [3, 5, 2, 13] and new privacy attacks on both tabular data [24, 17, 22, 9] and machine learning models [11, 25, 23] indicate, both security and privacy remain huge challenges. My research lays foundations for new systems that address these challenges. To accomplish this goal, my research has made contributions in a number of different areas:

- **Differential privacy for database queries:** My research has resulted in a new approximation of a query’s sensitivity and the first available system for enforcing differential privacy for SQL queries [14, 1]. This system employs a novel dataflow analysis for SQL queries to calculate elastic sensitivity.

- **Differential privacy for machine learning:** In ongoing work, I am developing new algorithms and practical systems that can provide differential privacy for machine learning tasks. This work has resulted in the first practical differentially private algorithm for convex optimization, the first broad empirical comparison of algorithms, and the first open source platform for differentially private machine learning.

- **Security for web applications:** My dissertation work addressed the challenge of security bugs in web applications. I proposed a new approach to symbolic execution for dynamic languages [20], and used this approach to build Rubicon [18], a lightweight verifier for web applications; Derailer [19], a visualization-guided static analysis for finding security bugs; and SPACE [21], a nearly automatic tool for detecting access control bugs based on a set of patterns.

**My approach & impact.** A major goal of my research is breaking down the barriers to practical deployment of cutting-edge security and privacy techniques. My approach is therefore motivated in large part by practical, real world requirements. In all of my projects, I have begun by examining the remaining challenges in achieving some goal for a real world use case. The technical contributions of my work typically relate directly to resolving these challenges. I have evaluated the results on real use cases, from enforcing differential privacy on actual SQL queries written by analysts to finding bugs in open source web applications.

The development of elastic sensitivity (technical results described later) provides a perfect example. I began by building a close collaboration with an industrial partner, Uber, to discover the true real world requirements of enforcing differential privacy for database queries. Uber provided real queries and data to enable an empirical study demonstrating these requirements, and I used the results of this study to develop elastic sensitivity. In addition to proving its correctness, I validated elastic sensitivity by an empirical study of its utility on real world queries and data.

The resulting open source system is the first to enforce differential privacy for unmodified SQL queries, running on an unmodified SQL database. As a result, Uber is in the process of adopting it [7] to provide differential privacy in a number of contexts. This work has received wide coverage in the public press, including articles from Wired [12], Gizmodo [8], and more [6, 26], and invitations to speak at many events including Mozilla’s Privacy Lab, a Future of Privacy Forum workshop, the FTC, and Enigma 2018. It has sparked conversations with a number of organizations interested in deploying similar systems, including Jingdong, Mozilla, Stripe and the Winton Group, as well as dialog about the future of differential privacy and privacy policy in general with a number of advocacy groups and policymakers, including the ACLU, the EFF, and the FTC.
Differential Privacy for Data Analytics

Differential privacy provides a rigorous framework for building algorithms that protect the privacy of individuals, and has become the gold standard for privacy in recent years. Despite this success, differential privacy has seen little adoption in practice. In joint work with Noah Johnson, Dawn Song, and Joseph Hellerstein, and leveraging a close collaboration with Uber, I have made several important contributions towards making differential privacy practical for general-purpose analytics.

**Theoretical contribution: elastic sensitivity.** Differential privacy can be enforced by adding random noise to a query’s result. The *sensitivity* \[10\] of the query determines how much noise is required. Elastic sensitivity \[14, 15\] is the first tractable approach to leverage local sensitivity for queries with general equijoins. The key insight of our approach is to model the impact of each join in the query using precomputed metrics about the frequency of join keys in the true database.

**Systems contribution: differential privacy for SQL queries.** To enable the practical use of elastic sensitivity to enforce differential privacy, we built and released a system \[1\] that calculates elastic sensitivity for arbitrary SQL queries and adds the appropriate noise to the results. To the best of our knowledge, this is the first system to enforce differential privacy for an unmodified SQL database. As part of this system, we developed a novel dataflow analysis for SQL queries; we have validated the system on more than 10,000 real queries written at Uber.

**Ongoing work: CHORUS.** Some differential privacy mechanisms (including elastic sensitivity) can be implemented by post processing the query’s results, but many others cannot. In ongoing work, we are extending our existing approach to develop CHORUS, which enables a broad range of differential privacy mechanisms without modifying the underlying SQL database. The key insight of CHORUS is to embed the differential privacy mechanism into the SQL query before execution, so the query automatically enforces differential privacy on its own output.

Differential Privacy for Machine Learning

Differential privacy and machine learning share a fundamental alignment of goals—the ability to *generalize* to properties of a population, and avoid *overfitting* to properties of any one individual. However, as a number of recent attacks demonstrate \[11, 25, 23\], many machine learning models do *not* protect the privacy of individuals in the training set automatically.

Models can be trained with differential privacy to protect individuals in the training set, and theoretical results suggest that machine learning with differential privacy can have minimal impact on the accuracy of the model. In ongoing work with Om Thakkar, Abhradeep Thakurta, Dawn Song, and undergraduate researchers Lun Wang and Roger Iyengar, I have made both algorithmic and empirical contributions in the area of differentially private convex optimization.

**Theoretical contribution: objective perturbation with approximate minima.** Towards the goal of practical differentially private convex optimization, we developed a new variant of the objective perturbation technique. Existing work on objective perturbation only guarantees privacy if the underlying optimizer is able to find the true minima of the perturbed objective—which is generally not possible in practice. Our new variant addresses this limitation with a privacy proof that only requires the optimizer to come close to the minima. Like other variants of objective perturbation, it provides optimal utility bounds and can leverage an off-the-shelf optimizer. In addition, our algorithm can be deployed without the need to tune hyperparameters—an important consideration, since privacy-preserving hyperparameter tuning is a challenging open problem.

**Systems contribution: empirical evaluation and open source platform.** To compare the practical performance of differentially private convex optimization algorithms and quantify the cost of privacy in real-world settings, we performed the first broad empirical evaluation of the available algorithms. As part of this effort, we are developing a platform and benchmark suite for differentially private convex optimization. This platform will provide open source implementations of differentially private algorithms from all major approaches, plus a set of benchmarks for reproducing our empirical results and comparing new algorithms. For many algorithms, our platform will provide the first implementation available *anywhere.*
Detecting Security Bugs in Web Applications

The web is fast becoming the most popular platform for application programming, but web applications continue to be prone to security bugs. As part of my dissertation work, I built lightweight verification tools for detecting security bugs in the server-side code of these applications.

**Theoretical contribution: language-based symbolic execution.** Web applications are typically built from small, independent actions, and they have fewer conditionals and loops than typical code, reducing the number of program paths. However, the dynamic languages often used in web application programming pose additional challenges for static analysis. To enable symbolic execution in this setting, I built the first symbolic evaluator as a library for Ruby on Rails applications. My approach uses the standard Ruby interpreter to perform symbolic execution [20], ensuring perfect compatibility and minimizing complexity—the library is just 1000 lines of code. By integrating tightly the standard interpreter and leveraging domain-specific properties of web applications, my approach scales extremely well, analyzing large open source applications in just a few seconds.

**Systems contributions: tools for finding security bugs in web applications.** Leveraging my approach to symbolic execution, I explored three different approaches to detecting security bugs in real applications. First, I built Rubicon [18], a lightweight verifier for Ruby on Rails applications. Rubicon automatically checks verification conditions against a specification written by the developer using the Alloy Analyzer, an automatic bounded verifier. Rubicon generates verification conditions using symbolic execution of the Rails application. I used Rubicon to specify and check properties of five open-source Rails applications, and found a previously unknown security bug in the largest (Fat Free CRM).

Second, I developed Derailer [19], a visualization-guided static analysis for finding security bugs. Derailer provides a visual display of exposures, which succinctly represent data exposed by the application along with the conditions that cause its exposure. Derailer makes it easy to quickly discover exposures missing some checks, which tend to represent bugs. I used Derailer to find dozens of bugs in student projects intended to demonstrate access control.

Third, I formalized the most common patterns of access control policies and built SPACE [21], a nearly automatic tool for finding access control bugs. SPACE uses symbolic execution to generate verification conditions, and checks them against this formal pattern library using the Alloy Analyzer. Using SPACE, I found 23 previously unknown security bugs in 50 open-source Rails applications, with just 10 false positives.

**Future Work**

With the increasing prevalence of data collection, concerns about security and privacy are only growing. Building on my previous work, I plan to pursue a research agenda towards approaches that empower developers to deploy practical security and privacy protections.

**Protecting privacy in big data analytics and machine learning.** A number of important open challenges remain in developing broadly applicable solutions to differential privacy. Key among these is the impact of the privacy budget, some of which is used to answer each query; when it is depleted, no more queries may be answered. A number of different approaches have been proposed for maximizing the privacy budget; I plan to study the real-world impact of these different approaches and develop improvements, based on my collaborative relationship with Uber and new collaborations with other partners.

I would also like to extend CHORUS to support combining mechanisms to answer a single query, and to support multi-query mechanisms like the matrix mechanism [16], which can provide significantly better utility for large query workloads. Unlike current systems, CHORUS will be able to support both extensions without requiring modifications to the database engine.

I plan to develop a complete platform for privacy-preserving machine learning, providing comprehensive and practical implementations of many different machine learning tasks in a manner similar to scikit-learn [4]. To this end, I plan to also explore the non-convex setting, including deep learning, recommender systems, decision trees, and others. For many of these, differentially private algorithms are known, but challenges remain in achieving good utility, and practical systems are not available.

In addition to containing algorithm implementations for practitioners to use, this platform will provide a
base for building new algorithms and a standardized benchmark for comparing them to existing approaches,
accelerating the pace of research on differential privacy in machine learning.

**Automatic security & privacy with optimal utility.** Most work on differential privacy has focused
on designing algorithms for specific tasks, with pen-and-paper proofs of privacy. My work considers more
general solutions, at the cost of (sometimes) sub-optimal utility. A key challenge for future deployments
will be achieving both at once—building frameworks that enable developers who are not privacy experts to
achieve both privacy and optimal utility.

I plan to build a new framework that allows privacy non-experts to develop domain-specific systems for
enforcing security and privacy guarantees. By providing a set of building blocks with known properties for
the developer to assemble, this framework will automatically provide proofs of security and privacy properties
of the resulting system. Building blocks will include differentially private algorithms for machine learning,
analytics, and query workloads, allowing developers to tailor the assembled system to achieve optimal utility
for their specific tasks.

### References


