Research Statement

Roxana Geambasu, Columbia University

I am a software systems researcher with particular interest in security and privacy. At Columbia, I have focused on developing a *new model for privacy* in our data-driven web world. Today's web, a complex ecosystem, is largely driven by the collection, mining, and monetization of personal data. Many web services, mobile applications, and third-party trackers collect and use our personal data for varied purposes, e.g., to target advertisements, personalize recommendations, and fine-tune prices. Some of these uses offer benefits to us (e.g., recommendations on Netflix or Pandora), but others may impose serious personal costs. At present, we have no window into how our data is being managed and used, and service providers have limited accountability, raising the risk for deceptive, unfair, or insecure practices.

I seek a new model for how we address such personal privacy issues. I envision a web environment where users can become more aware of the privacy consequences of their online actions and make more informed decisions about the services they choose or reject. In my model, services and applications are explicitly constructed to protect user privacy, and transparency enables accountability for inappropriate data use. To forge this new environment, I design, build, and evaluate the following systems, tools and abstractions, which I describe in detail in subsequent sections:

1. *New web transparency systems* to increase users' and society's oversight regarding how applications use personal data in order to detect and deter unfair and deceptive practices.
2. *New testing and certification tools* to assist developers in creating data-driven applications that are more privacy-preserving, fair, and robust *by design*.
3. *New protection abstractions* that facilitate and promote a more rigorous and selective approach to data access, sharing, and retention in today's data-driven companies.

My privacy work has garnered the following recognition: two best paper awards (EuroSys'11, USENIX Security'09); technology transfer to a New York City-based startup; extensions of my work by top teams from CMU, Duke, and Facebook AI Research; invitations to address the Federal Trade Commission (FTC), members of the Cybersecurity Caucus of the U.S. House of Representatives, and members of the U.S. Intelligence Community; ongoing collaborations with the FTC, the U.S. Food and Drug Administration, and the New York Presbyterian hospital; media coverage (e.g., *The New York Times*, National Public Radio, and *The Economist*); and faculty awards including Sloan Foundation fellowship, NSF CAREER award, Microsoft Faculty Fellowship, Google Research awards, and Popular Science Brilliant 10 award, 2014.

1 Web Transparency Systems

Are third-party web trackers watching our children's online activities and targeting them with ads for inappropriate products? Do shopping, loan, or housing websites tailor their prices based on what they know about us? Is our data being shared with unexpected parties and, if so, how are those parties using it? Today, we lack verifiable answers to even simple questions about what black-box web services and third-party groups are doing with personal data they collect. We must rely on explanations or privacy statements that companies make, which are often vague, incomplete, or (as our research found) inaccurate [10]. And *privacy watchdogs* – such as FTC investigators and investigative journalists – lack the tools needed to track online personal data flows to discover deceptive or unfair practices.

To address this situation, my collaborators and I are building the first scalable, generic, and interpretable systems that detect data flows within and across web services. We call these *transparency systems*, and their purpose is to help end-users and privacy watchdogs better understand web service use of personal data. Thus far, we published two transparency systems – *XRay* [7] and *Sunlight* [10] – that detect data use for targeting and personalization based on causal inference from randomized experiments with synthetic accounts.

The key insight in the first system, *XRay* [7], is that we can infer targeting by correlating user inputs (such as searches, emails, or locations) to service outputs (such as ads, recommendations, or prices) based on observations obtained from synthetic accounts populated with randomized subsets of the inputs. XRay pioneered this methodology to detect online targeting in a generic way; after its introduction in 2014, other systems with similar goals became available (e.g., CMU's AdFisher [5]). A unique property of XRay remains its scalability, enabling large-scale studies of targeting on a large number of inputs. Indeed, XRay's algorithms need a number of synthetic accounts that grows only logarithmically with the number of inputs to track, a property that my collaborators and I proved both experimentally and theoretically. However, the lack of tests of statistical validity for XRay's predictions make its conclusions difficult to interpret.

The second system, *Sunlight* [10], offers statistical justification and a clean causal interpretation for its inferences while preserving the same scaling properties as XRay. To date, Sunlight remains the most scalable system for detecting the causal effects of online targeting. It is based on three insights. First, it decouples formulation from statistical testing of targeting hypotheses. This lets us use scalable machine learning (ML) models to formulate hypotheses on a training set and assess statistical significance on a testing set. Second, statistical methods to control for multiple hypothesis testing heavily penalize incorrect assumptions. Uniquely, Sunlight favors high-precision algorithms to formulate hypotheses, which, counter-intuitively, results in higher recall after statistical testing. Third, the correlations Sunlight detects have a causal interpretation because input values are randomly assigned synthetic accounts independent of other factors.

Using Sunlight, my team and I ran the *largest-scale studies of ad targeting* on Gmail and on the web [10]. We found solid evidence that *contradicted two of Google's own statements* related to targeting on sensitive topics. For example, in an FAQ about Gmail ad targeting, Google formerly stated that it "will not target ads based on sensitive information, such as race, religion, sexual orientation, health, or sensitive financial categories." We created 300 Gmail accounts that exchanged emails containing keywords related to each of these topics. We collected 20M unique ads that Gmail delivered to those accounts over a period of 30 days in September-October 2014. We then used Sunlight to reverse the ads' targeting against the emails and kept only the statistically significant results (corrected p-values < 0.05). We found numerous ads that targeted *each and every sensitive topic* that Google committed to prohibit. Some of the targeting was quite disturbing. Numerous ads for payday and subprime loans targeted users who expressed financial concerns in their private communications (e.g., used "unemployed" or "bankrupt" keywords in their emails). Ads for shamanic healing and spouse surveillance services targeted users who expressed health concerns (e.g., used "cancer," "depressed," or "sad" keywords in their emails). While we do not claim that this was prompted by our work, the Gmail ad service we studied was discontinued in late 2014, and the FAQ was removed.

These results attracted attention not only from the research community and the media but also from the FTC, who expressed interest both in our data and in scalable systems to automate their largely manual studies of deceptive targeted advertising. In terms of data sharing, we shared the data we collected from both the 30-day Gmail study and a 4-month Facebook study that we executed with their input. FTC investigations are confidential, so we do not know whether/how these results were used. In terms of system sharing, we are working on a transparency system that replaces randomized experiments on synthetic accounts, with observational data supplied by volunteer users to study targeting effects at scale. The observational setting is more attractive to the FTC because synthetic accounts are expensive to create and maintain at scale. However, inferring causality from observational data is also notoriously challenging. Our new system combines the latest methods from causal inference on observational data with automatic experiment generation that confirms causation of a small number of promising hypotheses using small randomized trials.

Since I started my work on web transparency, the topic has gained substantial traction, with numerous papers verifying web service use of personal data and discovering irregularities every year. ACM recently introduced a new conference dedicated to topics of fairness, accountability, and transparency of data-driven systems (FAT*), on which I served as systems/measurement track chair last year. Despite substantial activity in this space, I believe that my work remains one of the few that develops scalable, reusable, systems infrastructure for auditing online use of personal data. My ultimate goal is to build a system akin to a scalable web crawler. Instead of finding and indexing online information for convenient access by the public, it would find uses of personal data and index their causes for convenient inspection by privacy watchdogs.

2 Testing and Certification Tools for Programmers

By constructing transparency tools and placing them into the hands of privacy watchdogs, I hope to increase society's oversight on web services' activities and interest developers in constructing services that better protect users' privacy. This has led to the second thrust of my privacy research: creating development tools that facilitate the construction of applications that are by design more privacy-preserving, fair, and robust. Today, creating such applications is difficult, particularly in the context of ML, which can exhibit unexpected behaviors, such as biased or offensive predictions, or predictions that change unnaturally upon slight changes to the input. Such behaviors can instill mistrust in ML, prevent its adoption in legally constrained domains, and open new attack vectors, such as adversarial examples.

My collaborators and I are developing tools for systematic and rigorous testing of fairness, robustness, and privacy properties of ML models by the developers who create them. We have published two tools thus far: *FairTest* [15], which detects violations of desirable fairness properties in ML models; and *PixelDP* [6], which provides a way to construct ML models with a guaranteed level of robustness against adversarial examples. These two properties are related, and we are now working on a single framework for formally verifying fairness, robustness, and privacy properties of ML models.

FairTest [15] lets developers systematically check for discriminatory effects of an ML model on slices of their user population. A programmer submits: the trained model (such as a model to predict hospital readmission based on patient medical history); a test set which includes not only user features that the model takes as input for inference but also protected attributes (such as race, age, or gender), on which the programmer does not want the model to discriminate. The programmer also supplies the fairness criteria from a variety of definitions that FairTest supports. FairTest evaluates the model on the test set and searches for interpretable contexts of the user population (such as users with some pre-existing conditions) where the model's outputs violate the fairness criteria. When I started work on FairTest, significant definitional work existed in the algorithmic fairness literature. However, the space was fragmented, with no tools for systematic detection of violations, and with custom-built detection techniques for specific definitions. FairTest makes two contributions. First, it unifies and rationalizes fairness definitions from 13 papers and supports them with a common, association-based abstraction. Second, it provides an efficient algorithm to systematically detect violations of specified fairness criteria not only at full population level, but also within smaller contexts where discriminatory effects are stronger. This lets programmers systematically explore and debug the effects of ML programs on target user populations.

My team and I used FairTest to study a model for hospital readmission prediction that won a Heritage Health Competition [15]. While the model showed high accuracy overall, its errors were unevenly concentrated on elderly patients, and the discriminatory effect was particularly strong in populations with certain pre-existing conditions, where the error could reach 45%. Therefore, an insurance company using this algorithm to tune insurance premiums might involuntarily discriminate against these elderly patients. Also using FairTest, we discovered that when conditioning on prediction confidence, the bias disappeared. Thus, to address the imbalance, an insurance company might adjust premiums only from high-confidence predictions.

PixelDP [6] lets programmers construct ML models that guarantee a level of robustness against adversarial examples. In this attack, an adversary finds a small perturbation to a correctly classified input, which results in a misclassification. For example, an adversary wearing makeup could fool a face recognition model into classifying them as someone else, enabling access to a building if the model's predictions are used for entry authorization. Numerous defenses have been proposed, but most were best effort and broken by subsequent attack instantiations. Recently, emerging *certified defenses* guarantee levels of robustness against arbitrary attack instantiations. However, they do not scale to large models or support all relevant model structures. PixelDP is the first certified defense that applies to large models and is agnostic to model structure. It is built on *differential privacy* (DP), a theory from the privacy domain. The expected output of a DP mechanism can be shown to be bounded under small changes in its input. PixelDP transforms a given ML model into a DP mechanism and uses the bound on its expected output to assess whether any attack below a given norm-size can change the prediction on a given input. If it cannot, the prediction is deemed certifiably robust against attacks up to that size. This *robustness certificate* for an individual prediction can serve two purposes. First, a building authentication system can use it to decide whether a prediction is sufficiently robust to rely on the face recognition model to make an automated decision, or whether a human should be consulted. Second, a model designer can use robustness certificates for predictions on a test set to assess a lower bound on accuracy under attack. This bound holds for arbitrary norm-bounded attacks, so there is no danger breaking it with future attack instantiations.

Using PixelDP, my team and I produced the first version of Google's Inception deep neural network for ImageNet that has non-trivial guaranteed accuracy under arbitrary, norm-bounded adversarial examples [6]. This network is many orders of magnitude larger than what previous certified defenses handled. Its guaranteed accuracy is reasonable for small attacks invisible to a human eye (e.g., 60% for 2-norm attacks of size 0.1). For state-of-the-art contemporary attacks, PixelDP outperforms the guarantee's predictions: it maintains an accuracy above 60% for attacks that are four times as large (though still invisible to humans) and performs better than other certified and best-effort defenses. Moreover, if one is willing to act only on predictions that PixelDP certifies as robust (as in the preceding building authentication scenario), then accuracy jumps close to the undefended and unattacked network's accuracy for the approximately 70% of the predictions that PixelDP deems robust. Using a recent Google Research award, we are now working to make the guarantee practical for larger attacks (hopefully visible to humans) by leveraging variational inference techniques to account for and eliminate the negative effects of DP noise while preserving the guarantees afforded by it. Other teams, from CMU [4], Duke [11], and Facebook AI Research [12], are also extending our approach to improve the bounds at a given noise level [4, 11] or use other noise distributions [12].

FairTest and PixelDP exemplify my broader vision of developing a framework for systematic testing and certification of important properties for ML systems, such as fairness, robustness, and privacy. Perhaps surprisingly, these three classes of properties are related. For example, individual fairness requires that whenever two users are "similar" (according to some distance function), an ML model's predictions should also be similar. This resembles a distance-based definition of robustness against adversarial examples. And, as our use of differential privacy to guarantee robustness shows, robustness and privacy are also related. This suggests that we may be able to build a unified and comprehensive framework for certifying many types of desirable properties for ML models, and perhaps leverage techniques from certifying robustness (including PixelDP) to certify fairness. Doing so would significantly advance the algorithmic fairness literature, which has thus far been dominated by best-effort, empirical assessments on limited test sets.

3 Protection Abstractions for Data-Driven Ecosystems

The third thrust of my privacy research revisits decades' old protection abstractions from operating systems (OS), which have become unfit for emerging data-driven workloads. Traditional protection units –

such as files, directories, or database tables – fail to support the data access and sharing patterns common in ML ecosystems. This leads some companies to adopt either too loose, wide-access data policies (e.g., all engineers and processes within the company get access to all user data), or too restrictive, siloing policies (e.g., the data from service X is beyond reach for any engineer or process outside X's scope). Neither extreme is preferable: the former can result in wide exposure to hackers or snooping employees; the latter can disable potentially valuable uses of the data.

I believe that the most appropriate protection abstraction for ML is not files/directories/tables. Rather, it is *ML models*, particularly *feature models*. Most ML predictive models are not built directly on the raw data, but on features that encode, aggregate, and otherwise transform the raw data to make learning on it more efficient. These features, such as embeddings, covariates, and various statistical aggregates, are often shared across teams in big companies through company-wide feature stores. I thus developed the notion of a *private feature model*, a new abstraction for protected data access and sharing for ML ecosystems. A private feature model is one that is *learned incrementally* over historical data, made *differentially private* to bound leakage of the data through its parameters, and made *broadly accessible* within the company so engineers can use it to improve their service, ideally in lieu of accessing, and thus exposing, historical raw data.

My team and I developed, evaluated, and published *Pyramid* [8, 9], a system that implements a special case of this abstraction based on a feature model called *count featurization*. Count featurization, a frequently used method for scaling ML to very large datasets, replaces the features of an example with the probability of its label, conditioned on feature values. For instance, for a movie rating, *userId* becomes $P(rating|userId)$. Because the new features are low dimension, predictive models trained on them require much less data to fit. Pyramid constructs count tables to compute these probabilities over historical data and stores them with DP in company-wide feature stores. From there, they can be accessed by all engineers and incorporated as part of predictive models to reduce the amount and timeframe of the raw data to which the models need access. Using multiple workloads, including a production workload from Microsoft, we show that predictive models need access to 2-3 orders of magnitude less raw data when training with DP counts versus when training without the counts, while sacrificing 2-4% in accuracy. This reduces exposure of raw data – *and* improves training performance and resource consumption – by a corresponding 2-3 orders of magnitude.

I am now enhancing Pyramid to support arbitrary feature models, including embeddings, latent variable models, and sketches. Doing so raises substantial technical challenges because each feature model exposes a different privacy-accuracy tradeoff when made DP, complicating the design of a uniform abstraction and system. I am addressing these challenges using a combination of new DP theory extensions and systems techniques for resource allocation to manage the privacy resource judiciously and enforce both a user-level global privacy guarantee and accuracy service-level agreements (SLAs) for all private feature models. Upon completion, I will evaluate Pyramid not only on workloads from or relevant to the tech industry, but also through a collaboration with Columbia's Medical School, the New York Presbyterian hospital (NYP), and the U.S. Food and Drug Administration (FDA). These entities have engaged me to develop a system that continuously trains and releases DP latent variable models over NYP's streams of clinical patient records to enhance biomedical research on certain public datasets, such as FDA's drug adverse reaction database, which currently lacks deconfounding information and hence cannot support causal studies of drug side effects. Our preliminary evaluation with NYP data suggests that DP latent variable models offer a rich source of deconfounding information. This will be a great example of the power of using differentially private feature models as unit for protected data sharing.

Beyond ML-driven workloads, I have also worked on new protection abstractions for mobile workloads [13, 14]. A similar mismatch between traditional protection abstractions and the needs of emerging workloads occurs in mobile devices. While traditional OSes provide low-level storage abstractions – files and directories – modern OSes embed higher-level abstractions, such as relational databases or object-relational models. Despite the change in abstraction, many crucial protection systems, such as encryption or deniable systems, continue to operate at the former file level, which renders them ineffective since files are both too fine- and too coarse-grained for effective protection. My collaborators and I developed two systems – *CleanOS* [14] and *Pebbles* [13] – to introduce a new and more meaningful abstraction for protecting data in mobile OSes: *logical data objects*. These correspond to user-relevant objects, such as emails, documents, bank accounts, and pictures. CleanOS opens an API for mobile app programmers to define logical data objects. Pebbles constructs these objects without programmer input based on pieces stored in various database and blob files. The two systems provide logical data objects as abstractions to let protection tools operate at a level relevant to user, such as encrypting, hiding, or properly deleting individual emails, documents, or bank accounts along with all pieces of related data.

4 Beyond Privacy

My primary research focuses on privacy, whose problematic state in today's data-driven world drives my efforts. But my curiosity and aptitude span other aspects of computer systems, as well, especially distributed and operating systems. As one example, my collaborators and I developed *Synapse* [16], a system that replicates data cleanly across heterogeneous databases, including Oracle, MySQL, Cassandra, MongoDB, ElasticSearch, Neo4J and others. The key idea is to perform the replication at the level of object-relational mappers, a common abstraction that web developers often use when interacting with databases. This way, many of the details of each database are abstracted away; for differences that remain, it turns out that simple interfaces can enable programmers to resolve the inconsistencies. We deployed Synapse at a NYC-based social media startup in 2013, which has been using it in production for at least two years with 650K users.

As another example, I observed that today's applications running on modern operating systems (OSes) – such as Android, iOS, and OSX – require very different abstractions from those offered by traditional OS standards, such as POSIX. The new abstractions, which are implemented atop POSIX, are offered as part of user-space libraries. How well are the new abstractions supported by the traditional ones? To find out, my collaborators and I measured POSIX use by Android, iOS, and OSX applications [3]. We found that many of the new OS abstractions rely upon POSIX's unstructured extension interfaces (such as `ioctl`) to implement their functionality, suggesting serious mismatches between traditional and modern OS abstractions. The mismatches we observed to the file system POSIX API motivated and fueled the design of Pebbles (§3).

Finally, on the topic of live virtual machine migration, my collaborators and I developed ways to leverage past state access histories to enable the streaming of large virtual machines over low-bandwidth cellular networks with limited loss of interactivity [1, 2].

5 Summary

Across these various research areas, my work provides a vision of how emerging computing technologies – such as big data, cloud computing, and mobile devices – can enrich our lives without imposing unknowable personal privacy costs. Much of my Columbia research focuses on a new model for privacy that is suited to our emerging data- and ML-driven world. My model involves building and deploying a scalable external transparency infrastructure for the web that will increase society's oversight on web services' use of personal data while providing the critical missing tools needed by the services to themselves safeguard this data. Unlike other privacy models, which rely on protection or prevention of data access from the web services, my model puts forward transparency and accountability as key new components of designing, overseeing, and choosing a more private data-driven world.

References

[1] Yoshihisa Abe, Roxana Geambasu, Kaustubh Joshi, H. Andres Lagar-Cavilla, and Mahadev Satyanarayanan. vTube: Efficient streaming of virtual appliances over last-mile networks. In *Proceedings of the Symposium on Cloud Computing (SoCC)*, 2013.

[2] Yoshihisa Abe, Roxana Geambasu, Kaustubh Joshi, and Mahadev Satyanarayanan. Urgent virtual machine migration with enlightened post-copy. In *Proceedings of the Conference of Virtual Execution Environments (VEE)*, 2016.

[3] Vaggelis Atlidakis, Jeremy Andrus, Roxana Geambasu, Dimitris Mitropoulos, and Jason Nieh. POSIX abstractions in modern operating systems: The old, the new, and the missing. In *Proceedings of the IEEE European Conference on Computer Systems (EuroSys)*, 2016.

[4] Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv:1902.02918*, 2019.

[5] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *Proceedings of Privacy Enhancing Technologies Symposium (PETS)*, 2015.

[6] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *Proceedings of the IEEE Security and Privacy Symposium (IEEE S&P)*, 2019.

[7] Mathias Lecuyer, Guillaume Ducoffe, Francis Lan, Andrei Papancea, Theofilos Petsios, Riley Spahn, Augustin Chaintreau, and Roxana Geambasu. XRay: Increasing the web's transparency with differential correlation. In *Proceedings of USENIX Security*, 2014.

[8] Mathias Lecuyer, Riley Spahn, Roxana Geambasu, Tzu-Kuo Huang, and Siddhartha Sen. Pyramid: Enhancing selectivity in big data protection with count featurization. In *Proceedings of the IEEE Security and Privacy Symposium (IEEE S&P)*, 2017.

[9] Mathias Lecuyer, Riley Spahn, Roxana Geambasu, Tzu-Kuo Huang, and Siddhartha Sen. Enhancing selectivity in big data. In *IEEE Security and Privacy Magazine*, 2018.

[10] Mathias Lecuyer, Riley Spahn, Yannis Spiliopoulos, Augustin Chaintreau, Roxana Geambasu, and Daniel Hsu. Sunlight: Fine-grained targeting detection at scale with statistical confidence. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2015.

[11] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness. *arXiv:1809.03113*, 2018.

[12] Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cedric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization: the case of the exponential family. *arXiv:1902.01148*, 2019.

[13] Riley Spahn, Jonathan Bell, Michael Lee, Sravan Bhamidipati, Roxana Geambasu, and Gail Kaiser. Pebbles: Fine-grained data management abstractions for modern operating systems. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2014.

[14] Yang Tang, Phillip Ames, Sravan Bhamidipati, Ashish Bijlani, Roxana Geambasu, and Nikhil Sarda. CleanOS: Mobile OS abstractions for managing sensitive data. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2012.

[15] Florian Tramer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. Fairtest: Discovering unwarranted associations in data-driven applications. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017.

[16] Nicolas Viennot, Mathias Lecuyer, Jonathan Bell, Roxana Geambasu, and Jason Nieh. Synapse: New data integration abstractions for agile web application development. In *Proceedings of the ACM European Conference on Computer Systems (EuroSys)*, 2015.