# Research Statement

Yitao Duan
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720, USA
duan@cs.berkeley.edu

My primary research interest is in practical privacy and security technologies for a variety of applications, such as ubiquitous computing, collaborative and distributed computation, data mining, etc., and applied cryptography. The goal of my research is to develop protocols and schemes, with provably strong privacy and security, that are practically realizable on today's commodity hardware. Furthermore, I am not satisfied with merely "finding the perfect technology". I would like to see my work make positive impact in the real world. I strive to make my solutions not only practical in terms of performance, but also socially acceptable so that they will be adopted.

Based on my training and research philosophy, I situate myself at the intersection of three major research areas in computer science: cryptography, systems, and human-computer interaction (HCI). Cryptography provides powerful primitives, paradigms, and theories that can be used to build systems with properties that are provable, either unconditionally or under some reasonable assumptions. The importance of cryptography has been demonstrated by its numerous applications in our daily life, from secure online banking to password protection. Modern cryptography is concerned with more than just encryption. It also provides tools such as zero-knowledge proof (ZKP) and secure multiparty computation (MPC) that offer mechanisms for achieving and verifying security properties without violating data privacy. However, a solution can only make a difference when it is implemented and running in the real world. And there are a lot of system issues when one tries to build a real cryptographic system. For example, many theoretical approaches stop at asymptotic complexity analysis, and the actual cost is often hidden behind the notation. System considerations, on the other hand, expose the real operation costs and identify bottlenecks. By thinking beyond asymptotic complexity and considering system as well as theoretical issues, I strive to build real, practical systems with provably strong security and privacy. The P4P project, which is a practical framework with open source toolkits for privacy-preserving distributed computation, is an effort to move cryptographic private computation into the realm of the practical for many Internet and intranet applications. Finally, technology does not exist in vacuum. HCI provides a human-centered perspective and stresses the usability, acceptance, social effects, etc., of technology, which are vital for developing viable technological solutions. Many non-technical factors such as legal, economic, and social issues etc., have been shaping how technology is conceived, developed and especially how it is adopted. While these factors place constraints on the design of technology, they also provide leverage that we can use to boost system performance and optimize a protocol. My exposure to the HCI principles (through classes and collaboration with colleagues in the Berkeley Institute of Design, an interdisciplinary lab consisting of faculty, students and researchers in areas such as HCI, mechanical design, education, etc.) allows me to pay attention to these factors while designing secure algorithms. My work not only carefully considers such factors to maximize adoption, but also draws upon them for its security and efficiency.

## Research Projects

**Practical Privacy-Preserving Distributed Computation** Modern networked applications and services, such as e-voting, e-commerce, data mining etc., often call for aggregation of user data. Privacy concerns have become a major obstacle hindering the acceptance/success of such applications. Existing solutions such as secure multiparty computation suffer from prohibitive cost and are not practical. I strive to improve the situation. My work draws on the insights offered by studies on privacy from legal, economic and social science perspectives. Two of the most important observations are power imbalance and lack of incentive. That is, the party seeking information (e.g. the big online vendor, the employer, the government) is usually much more powerful than the individuals who are trying to protect information. The individuals have limited resources and influence to convince a powerful actor otherwise. The powerful actors are reluctant to adopt and participate in privacy-preserving schemes because they perceive that such schemes will decrease the amount of information they obtain, increase their operational cost, and/or reduce the precision of the computation. These phenomena are also prevalent in today's computing reality: a large number of existing systems are server-based where the owner of the server possesses all the information and controls who has access to it. The goal of my work is to build privacy technologies that acknowledge these factors and work with them, using novel private computation approaches. Private computation provides aggregation of user data (means, sums, statistical analyses etc.) but provides no other information (in a cryptographic or information-theoretic sense) than the final aggregate.

Peers for Privacy (P4P) is a framework for carrying out private computation with adequate performance on today's commodity hardware at reasonable, realistically large scale. P4P leverages the natural incentives on the user side. Most users want better privacy, but don't understand the risks and generally are not willing to pay much for better protection. On the other hand, user communities always contain a fraction of altruistic users who provide resources to the rest of the community – this is how most peer-to-peer computing actually works. Our approach uses resources from a few of these users which allows a very simple, efficient solution that places little or no additional demands on either the server or the other users.

The P4P framework features a unique hybrid architecture combining P2P and existing client-server paradigm, and takes advantage of players' heterogeneity that exists in real world systems. The bulk of the computation and storage are still based on the server, but a (small) subset of users, denoted Privacy Peers, participate in the computation, removing data control from the server. The privacy peers do not gain control of the data, and need not be trusted - they only provide compute cycles. This arrangement allows us to take advantage of the different resources and protections offered by different players. In terms of security and privacy, P4P relies on the server, which is typically protected behind firewalls and maintained by professional administrators, for defending against outside attacks and uses the privacy peers, which are mostly client machines maintained by regular users, to protect user privacy against a curious server. Performance-wise, this architecture allows us to use a verifiable secret sharing (VSS) paradigm for computation. The advantage of a VSS private computation scheme is that it works with any sized field. And arithmetic operations with small field elements are extremely efficient, when an element fits in a single memory cell (in contrast, arithmetic operations with cryptographic field numbers, as is required by almost all existing MPC protocols and ZKPs, are typically $10^6$ times slower). This means private computation in P4P is almost as efficient as the centralized implementation if the computations are composed mostly of additions. In a sense, for these applications, P4P can provide privacy almost for free (in terms of server's

computation overhead). Addition-based algorithms are more general than would appear, and include non-linear gradient approaches such as Singular-Value Decomposition and many data mining algorithms based on EM (Expectation Maximization), etc. Thus P4P provides efficient solutions for a large number of real world applications.

The P4P framework also supports a very efficient user data validation protocol that verifies, in zero-knowledge, that the L2 norm of the vector a user inputs into the computation is bounded by a predefined limit. This prevents a malicious user from exerting too much influence on the computation, which is a serious threat in any realistic application but generally lacks scalable solutions. The protocol uses only a logarithmic number of large field (1024 bits or more) cryptographic operations. In experiments with our implementation, it has been shown that verification of a million-element vector (e.g. during an SVD calculation) takes a few seconds of server or client time on commodity PCs (in contrast, using standard techniques takes hours). Overall, the protocol is dominated by the (linear) time to do small field operations that one has to pay even when the computation is done directly on user data without privacy. This makes privacy protection almost free from the vendor's point of view, which is essential for wide adoption. In addition, P4P can also deal with malicious privacy peers who participate in the computation. However, this is done without resorting to expensive ZKPs or homomorphism. Instead, we introduce a new VSS that takes advantage of the existence of honest majority among the players and relies on consensus for identifying correct behavior. The resulting scheme preserves the feature of "keeping the number of large integer operations small".

P4P has been implemented (in Java) and tested with real data. This includes the vector verification protocol and all its necessary cryptographic components which can also be used independently. I am adding more "middle tier" components to support more concrete applications including some common statistical aggregates such as ANOVA, correlation, and sparse factor analysis. The P4P code will be released to general public for free. I believe it will be a valuable tool for developers in areas such as data mining and others to build privacy preserving real-world applications.

**Data Protection in Ubiquitous Computing** An ubiquitous computing environment is typically envisioned as a physical space containing a large number of invisible, collaborating computers, sensors and actuators interacting with user-worn gadgets. Data about individuals who are in the environment is constantly being generated, transmitted and stored. There are many challenges facing data protection in such setting, including dynamic and unfamiliar environment, ubiquitous and high-speed data generation, lack of central point of control, etc. Traditional access control-based solutions are inadequate in this setting in that they rely on active code to guard the data and oftentimes it is difficult or even impossible to deploy such protection because data can be accessed through multiple mechanisms. Data are not protected if the authorization agent is down or faulty, or bypassed. And such situations can be quite common in ubiquitous computing environments (e.g. mobile devices are much easier to lose). In this project I proposed the Data Discretion Principle which states that users should always have access to, and control of (recorded or live) information that would be available to them in "real-world" situations. They should not have direct access in other situations. I provide a technical scheme to enforce this principle. The method is a key embedding scheme that embeds access rights in the data while the data are being generated. The security of our scheme does not rely on a centralized access control system. Rather, it uses cryptographic techniques to manage access through metadata. This approach facilitates sharing of data among legitimate users and allows for trustworthy verification of an ubicomp system's privacy policy.

**Multicast Encryption and Secure Group Communication** Unlike point-to-point communication, multicast allows multiple recipients to access the message transmitted from a single sender. Multicast encryption is concerned with protecting the secrecy of the data such that only the intended parties can read the data. In this project I introduced a general framework for constructing efficient multicast cryptosystems with provable security and show that a line of previous work on multicast encryption are all special cases of this general approach. The new framework makes novel use of threshold decryption and provides new methods for building such cryptosystems with various levels of security (e.g., IND-CPA, IND-CCA2). The results enable the construction of a whole class of new multicast schemes with guaranteed security using a broader range of common primitives such as OAEP. Moreover, I show that multicast cryptosystems with high level of security (e.g. IND-CCA2) can be based upon public key cryptosystems with weaker (e.g. CPA) security as long as the decryption can be securely and efficiently "shared". The constructions feature truly constant-size decryption keys whereas the lengths of both the encryption key and ciphertext are independent of group size.

Later this work was extended to support secure bidirectional group communication which is important in many applications such as sensor network, intrusion detection etc., but lacks scalable solutions. This work considers both one-to-many (multicast) and many-to-one communication. For the 2nd case (many-to-one) the challenge is to provide data authenticity since there are multiple data sources. The new scheme makes novel use of the key structure of our multicast encryption and provides a scalable secure group communication scheme where both the message size and the client key length are independent of the group size.

## Future Work

In the near future, I plan to continue the exploration of the P4P since I believe it has strong potential. P4P already supports efficient private addition (over small fields). There is ongoing work on a new multiplication protocol. The idea comes from the observation that the two major paradigms for private arithmetic, secret sharing and threshold homomorphic encryption, are inherently connected and it is easy to switch from one to the other at some point of a protocol execution. I am working on a hybrid protocol where additions are done with secret sharing (over small field) and multiplications are performed using threshold homomorphic encryption. The advantage of this approach is that private additions are as efficient as regular ones and the only large integer operations are for multiplication that takes place in the threshold homomorphic encryption paradigm which has complexity linear in the number of players. There is also some work to future develop and maintain the P4P code (protocols and primitives) so that they can be a useful tool for building privacy-preserving applications.

In the long run, as computing touches more and more aspects of our daily life, security and privacy risks are becoming increasingly important. I envision cryptography and private computation will play increasingly important roles in emerging applications such as location-based services and ubiquitous computing, as well as traditional ones such as data mining, information retrieval, database, etc. My ultimate goal is to make such computations private and secure at reasonable cost so that people can enjoy the services while preserving their right to privacy. As research on privacy technology has inherently been interdisciplinary, I believe my work, by adding privacy to their applications, can make contributions to other areas of computing (e.g. HCI) as well and benefit the society as a whole.