From Usability to Secure Computing and Back Again

Lucy Qin, Andrei Lapets, Frederick Jansen, Peter Flockhart, Kinan Dak Albab, and Ira Globus-Harris, *Boston University;* Shannon Roberts, *University of Massachusetts Amherst;* Mayank Varia, *Boston University*

https://www.usenix.org/conference/soups2019/presentation/qin

This paper is included in the Proceedings of the
Fifteenth Symposium on Usable Privacy and Security.

August 12–13, 2019 • Santa Clara, CA, USA

ISBN 978-1-939133-05-2

From Usability to Secure Computing and Back Again

Lucy Qin[1], Andrei Lapets[1], Frederick Jansen[1], Peter Flockhart[1], Kinan Dak Albab[1], Ira Globus-Harris[1],
Shannon Roberts[2], and Mayank Varia[1]

[1]Boston University, MA, USA
[2]University of Massachusetts Amherst, MA, USA

Abstract

Secure multi-party computation (MPC) allows multiple parties to jointly compute the output of a function while preserving the privacy of any individual party's inputs to that function. As MPC protocols transition from research prototypes to real-world applications, the usability of MPC-enabled applications is increasingly critical to their successful deployment and widespread adoption. Our Web-MPC platform, designed with a focus on usability, has been deployed for privacy-preserving data aggregation initiatives with the City of Boston and the Greater Boston Chamber of Commerce. After building and deploying an initial version of the platform, we conducted a heuristic evaluation to identify usability improvements and implemented corresponding application enhancements. However, it is difficult to gauge the effectiveness of these changes within the context of real-world deployments using traditional web analytics tools without compromising the security guarantees of the platform. This work consists of two contributions that address this challenge: (1) the Web-MPC platform has been extended with the capability to collect web analytics using existing MPC protocols, and (2) as a test of this feature and a way to inform future work, this capability has been leveraged to conduct a usability study comparing the two versions of Web-MPC. While many efforts have focused on ways to enhance the usability of privacy-preserving technologies, this study serves as a model for using a privacy-preserving data-driven approach to evaluate and enhance the usability of privacy-preserving websites and applications deployed in real-world scenarios. Data collected in this study yields insights into the relationship between usability and security; these can help inform future implementations of MPC solutions.

USENIX Symposium on Usable Privacy and Security (SOUPS) 2019.
August 11–13, 2019, Santa Clara, CA, USA.

1 Introduction

Companies, educational institutions, government agencies, and other modern organizations have been collecting and analyzing data pertaining to their internal operations for some time with great effect to evaluate performance, to improve efficiency, and to test hypotheses. While each organization's own data sets have internal value, combining data from multiple organizations and analyzing it as a single corpus is likely to provide even more value to the organizations themselves, to policymakers, or to society at large. Unfortunately, each organization's internal data sets are often proprietary and confidential, and their release may be potentially deleterious to the organization's interests. Organizations may be able to release sensitive data selectively to specific agents entrusted with its analysis. This is often costly, requires that the organizations strongly trust the agent, and presents a security risk if the data sets are improperly handled.

A cryptographic primitive called *secure multi-party computation* (MPC) resolves this tension: aggregate data may be computed and released while preserving the confidentiality of each organization's internal data. This has significant potential social benefits: MPC enables groups of organizations to leverage collective data aggregation and analysis techniques in contexts where data sharing is constrained or prevented by legal and corporate policy restrictions.

As the focus of MPC research moves from the underlying theory to the issues that affect real-world use cases, friendly user interfaces and effective communication to non-expert users is crucial for effective deployment. First, interactions with users must build trust and create buy-in to the idea of secure multi-party computation. Second, an effective user interface is especially critical for MPC applications, as it is computationally expensive (or impossible) to verify the correctness of a user's inputs after they have been encrypted. Input validation poses a problem for computations involving many parties; one party's incorrect input, whether submitted maliciously or by mistake, could skew the results of the entire computation. Multiple studies have shown that poorly designed user interfaces hinder the success of security measures [45, 51]. A clearly designed and effective user interface can both maximize the rate of participation and minimize the chance of human errors leading to incorrect and misleading results.

1.1 Our contributions

This paper examines the connections between usability and cryptographically secure computing via MPC. We showcase the importance of usability within secure computing, as well as the value of secure computing in applications that calculate usability metrics in a privacy-preserving manner. We detail our experience designing and implementing a usable web-based framework for secure computing, which we call Web-MPC. We successfully deployed Web-MPC in two scenarios:

- Evaluating pay equity using the sensitive salary data of 166,705 employees from the city of Boston, MA (about 16% of the region's workforce) [14, 15].
- Measuring the rate at which member organizations within the Greater Boston Chamber of Commerce subcontract work to minority-owned businesses [42].

Usability challenges of secure computing The design of our framework is influenced by the interconnected usability, security, and legal requirements of our two applications, which we believe may be generalized to other scenarios. The primary benefit of our framework over prior deployments of MPC is an emphasis on the application's *usability* to drive participation within the target user community. CIOs, CTOs, human resources personnel, and lawyers from key participating organizations (along with social scientists and members of the city council that commissioned the studies) must all be able to learn and comprehend both the application itself and its underlying cryptographic properties.

We describe general usability challenges that MPC applications face in Section 3, and focus on usability challenges and solutions specific to our application in Section 4. We evaluate the effectiveness of our approach in Sections 5 and 8.

Employing secure computing to improve usability In the second half of this work, we describe our implementation of privacy-preserving web analytics on top of our existing Web-MPC framework. This enables the analysis of user behavior within the application without compromising any privacy goals or guarantees, and demonstrates that it is possible to measure and improve the usability of secure web applications with no privacy cost.

We discuss the importance of privacy-preserving usability data collection in Section 6, describe our implementation in Section 7, and analyze the collected results in Section 8.

1.2 Related Work

Usability research within the broader field of security is widespread. Countless "Why Johnny Can't Encrypt" papers have established that unless encryption is so seamless that the user cannot tell that they are encrypting, they will not bother to do so [45, 51]. Usability studies and work to incentivize use of secure platforms are common [21, 30, 46, 52]. However, these studies do not specifically tackle the difficulties in adopting MPC for a broader audience, nor do they use MPC to collect usability data.

Usability of MPC software There exist dozens of MPC software frameworks [27] and several successful deployments of MPC over the past decade [11, 13, 22]. These frameworks span a wide range of design choices that have usability implications for both developers and data contributors:

- Proprietary [12, 33] vs. open-source [2, 10, 17, 24, 28, 36].
- Access to low-level cryptographic primitives [8, 9, 23, 26] vs. use of programming language abstractions like data structures and formal type systems [7, 38, 44].
- Function specification in domain-specific languages [12, 44] vs. existing general-purpose programming languages [7, 47, 50].
- Whether data contributors run web servers for communication or leverage a web-based service for improved accessibility [29, 48].

Available frameworks also vary in software maturity, security guarantees, and programming APIs. Despite the widespread use of MPC, to our knowledge there has not yet been a public usability study of any application that employs MPC.

Security and privacy of usability analytics Within the cryptography community, previous work has been done to create privacy-preserving web analytics, with some focus on usability. However, these solutions have focused on using differential privacy [3, 18, 43], which provides fundamentally orthogonal (though compatible and complementary) privacy and security guarantees when compared to MPC.

2 Application Context

Our Web-MPC application was initially developed to aid a study through the Boston Women's Workforce Council. It has also been used for another initiative with the Greater Boston Chamber of Commerce. The design and implementation of Web-MPC was informed by nearly two years' worth of discussions with personnel (including CIOs, CTOs, HR executives, and lawyers from key participation organizations), social scientists, and members of the city council that commissioned the original Boston Women's Workforce Council study.

100% Talent Compact The Boston Women's Workforce Council (BWWC) is an initiative established in 2013 by Mayor Thomas Menino's office to measure and eliminate gender-based pay gaps [19]. In order to assess the wage gap, companies in the Greater Boston Area signed a compact promising to contribute their highly sensitive wage data across gender, race, and job categories. However, their effort was stalled due to privacy concerns over the collection of sensitive data. Employers were not willing to reveal their payrolls to a "trusted" third party and, conversely, no employer was willing to take on the role of a trusted third party due to the risk of storing or leaking such sensitive data.

As a result, we built a platform using secure multi-party computation to provide aggregate-level data statistics without collecting any company's individual data set. It has been successfully deployed in 2015, 2016, and 2017 with results and detailed analysis captured by reports through the Boston Women's Workforce Council. In the most recent 2017 deployment, the system aggregated data from 114 companies, representing 166,705 employees. This comprises over 16% of the Greater Boston Area workforce and almost $15 billion in collective annual compensation [20].

Since the application is used by HR professionals and other employees who do not have a background in cryptography, it is important for the user interface to be as intuitive as possible and to require no knowledge of the underpinning cryptographic technologies. Our application interface resembles the format of form EEO-1, which the U.S. Equal Employment Opportunity Commission requires companies to file annually. By using the familiar EEO-1 format, we aimed to improve the learnability and ease of use of our application, and to minimize errors in data submission.

Pacesetters Initiative The Greater Boston Chamber of Commerce (GBBC) launched the Pacesetters Initiative in January 2018 [41]. This initiative aims to enhance economic opportunities for minority-owned businesses by leveraging the purchasing capacity of medium and large businesses in the Greater Boston area [42]. A cohort of participating companies track and report metrics on their spending with Minority Business Enterprises (MBE), contrasted with their general spending across all subcontractors. The first data analysis occurred in March 2018, with the second one following a year later in February 2019. As a longitudinal study, the effort allows the GBCC to validate what effect their initiative has on equitable spending with MBEs. Given the data's sensitive nature, we partnered with the GBCC to use Web-MPC to securely and privately compute aggregate results.

2.1 Roles

Generalizing from our two application scenarios, we consider three types of roles in secure multi-party computation.

- An *analyst* (BWWC and GBCC in our settings) who specifies the analytics, handles some of the computational burden of calculating it, and receives its output.
- Several *contributors* who permit their private data to be used within the analytic's calculation. The number of contributors is unbounded and may be unknown in advance. In both our settings, Boston-area employers agreed to serve as contributors.
- An automated, publicly-accessible *service provider* that connects all other participants without requiring them to maintain servers or even to be online simultaneously, and that handles most of the computational burden in calculating the analytics. In our deployments, we (Boston University) configured a web server to act in this role.

2.2 Selection of MPC Protocol

In general, MPC assures a contributor that the analyst and service provider may only learn her data by pooling the information they receive. We rely upon passive (i.e., semi-honest) security, which informally states that if parties agree to adhere to the protocol and not collude together, then any passive attempt to glean information along the way is futile [25]. The service provider and analyst lack any clear incentive to falsify the results of the aggregation or collude to learn private input data. On the contrary, completing the data collection successfully is directly beneficial to the BWWC and GBCC (as the initiators of their respective initiatives) as well as to us as the service provider (who is incentivized to maintain a good reputation in order to deploy the application again in the future). These security protections also extend to external attackers who compromise the service provider. In more detail, MPC guarantees that read attacks against the service provider yield no private input data. Our implementation and MPC protocol also guarantees that inputs cannot be linked to the original contributor, parties that did or did not submit cannot be identified, and that the number of users does not need to be determined in advance.

We surveyed existing MPC implementations and their designs at the beginning of our effort [32]. None of the existing implementations at the time sufficed for our purposes. Some of them required the analyst to configure a public-facing web server, whereas others failed to provide the accessibility, asynchrony, auditability, resubmission, or other usability requirements listed in Section 4.2.

Instead of using existing frameworks, we opted to create a simple MPC protocol that was easy to implement without errors, straightforward to explain to users who are not domain experts, and adaptive enough to handle an *a priori* unknown number of participants. The protocol uses a variant of additive secret sharing in which random masks are added to each company's private inputs, as shown in Figure 1. The service provider (Boston University) then computes the aggregate sum of the masked inputs while the analyst (BWWC) receives the masks. The analyst is then able to subtract the aggregate of the masks from the aggregate of the masked values to get the aggregate data. This protocol is detailed in Appendix C.

2.3 Application Versions

Our Web-MPC application has gone through multiple iterations over time. There are two versions we will consider in our discussions, and we will refer to them as V1 and V2. V1 refers to first iteration of the application that uses additive secret sharing. The V1 data submission page is displayed in Figure 4. V2 refers to the current iteration of the application, after changes were made to the user interface based on the heuristics evaluation detailed in Section 5.1. Instead of additive secret sharing, V2 uses Shamir's Secret Sharing to support richer analysis (as discussed in Section 4.1). The most recent version also enables the creation of smaller participant subsets, called cohorts, within a session. For the 100% Talent Compact, these cohorts will consist of companies grouped by industry; the Pacesetters Initiative divides participants into cohorts based on prior participation. Aspects of the V2 user interface are captured in Figures 6, 8, 7, and 9. The differences between the V1 and V2 iterations of the platform are discussed in greater detail in Appendix B.

2.4 Deployment

The protocol and software application described in this section were deployed successfully by the BWWC [6, 35] and by the GBCC. We split each deployment into two phases: (1) a dry run on innocuous data and (2) a live analysis over sensitive data during which our team remained on-call to ensure any potential technical issues or usage questions were addressed.

Training Sessions The application's learnability, familiarity, and ease of use are critical to minimizing user error and to achieving a successful secure deployment. To preserve privacy, we could not help participants enter data into the user interface or allow them to ask us questions dependent on their data during actual deployments. This concerned us: we felt that we only had one chance to introduce MPC to participants. If even one participant entered erroneous data and the output was unsatisfactory, they might blame the technology (and switch to something less secure or not participate at all). To reduce this risk, we conducted dry runs involving fictitious data (provided to participants) in which they could ask us questions, and become familiar with the application interface and submission flow.

A dry run of the deployment served the purpose of familiarizing participants with the protocol, process, interface and requirements. We distributed an Excel spreadsheet that exactly matched the browser data entry interface, and ensured interoperability between the two. Contributors could use this spreadsheet to prepare their data and to verify that their browser allowed them to copy and paste data directly into the web application. The entire workflow was demonstrated via a live WebEx session that all participants could join. We initiated a mock collection, shared the session key, and encouraged all contributors to submit random data. This WebEx session was recorded, uploaded to YouTube, and shared with all participants so they could review it at their own pace. The training sessions provided the contributors with opportunities to ask questions, and allowed us to discover technical issues contributors might encounter (*e.g.*, using an outdated browser) without the risk of leaking information about their inputs.

Live Deployments In both deployments, the analyst was able to perform MPC jointly with our server to privately compute the aggregate across all contributing parties, and to delete their private key (in effect erasing the input data).

2.5 Mechanical Turk Usability Study

Prior to recent deployments, we conducted a usability study using Amazon's Mechanical Turk (MTurk) Platform to evaluate the success of different iterations of our application. We utilized MTurk users, rather than data contributors involved in either the BWWC or the Pacesetters deployments, in order to obtain data from a larger audience that is not already familiar with versions of the interface. Current data contributors have either already seen previous iterations of the platform or have received training during which they had the opportunity to ask questions. By using MTurk, we could assess whether our tool is usable with limited instructions and no prior training. During the study, MTurk users used our application to submit data before completing a System Usability Scale (SUS) questionnaire [16]. MTurks users were split into three groups. The first interacted with V1; the second and third interacted with V2, but were asked to enter data manually or via a spreadsheet. We describe the SUS and its results in Section 5.2. Additionally, during this study we collected usability data securely via MPC in the background as a proof of concept.

3 Usability Challenges in Deploying MPC

MPC introduces unique usability challenges. Target users are not domain experts and are unfamiliar with this technology; their willingness to use an application depends on their confidence that MPC protects their sensitive data and guarantees compliance with data sharing requirements. Also, the inherent privacy-preserving properties of MPC make it difficult or impossible to identify spurious or erroneous contributions that might compromise the overall analysis. Thus, the application's learnability, familiarity, and ease of use are also critical to minimize errors.

3.1 Inspiring Trust in MPC

Unlike more popular cryptographic primitives (*e.g.*, end-to-end encryption), MPC is not yet ubiquitously used in practice. MPC's guarantees and terminology are not widely circulated within non-technical or semi-technical contexts. As a result, HR personel, lawyers, CEOs, and end users are less likely to be aware of MPC guarantees or to have high confidence in it. This puts an additional burden on MPC application designers to communicate the guarantees of MPC to the relevant stakeholders and inspire confidence in its security.

This cannot be achieved solely by relying on mathematical and cryptographic proofs. Such proofs are not accessible to the wider population. Furthermore, they may not be convincing to someone that does understand what a proof entails. In our experience, analogies, examples, and concrete demonstrations of MPC play a large role in this endeavor.

Some contributors still require that the various compute parties within an MPC application sign a non-disclosure agreement governing data submitted by the contributors. This can be attributed to a lack of understanding or confidence in MPC guarantees, as well as familiarity with NDAs and their use to mitigate liability. However, due to the way that MPC works, we believe participants would benefit more from the creation of a different legal construct: a "non-collusion agreement" with enforceable civil penalties.

3.2 Correctness and Participation Trade-offs

When building usable MPC platforms, designers must negotiate an inherent trade-off between the participation rate, the correctness of the aggregate output data, and the security of the input data. With an increase in contributors, the chance that at least one contributor provides incorrect data increases; in other words, increased participation adversely impacts correctness. Simultaneously, participants are more likely to participate if they have confidence that the computation will be correct (and, thus, useful).

Error Sensitivity Unlike traditional computation platforms, the nature of dealing with private inputs under MPC makes error recovery tedious (if not impossible). MPC does not allow any single party to look at input data, or to analyze it manually. This makes it difficult to detect and correct invalid input data and to remove outliers. It is also difficult to use contributor-specific context that could normally inform an analyst of potentially incorrect data. For example, a publicly traded multinational company that submits an input indicating it employs only five individuals is likely to represent a mistake, but this cannot be determined easily without seeing the individual inputs. Detection and correction logic can be encoded to run under MPC. However, this increases performance overhead and may require that all such logic be formalized and written down without prior knowledge of the input data's characteristics.

All these issues make it critical for the application to detect errors *before* contributors submit erroneous data and taint the aggregate results. This, in turn, necessitates that the application be easy to use and to learn, and that it allows contributors to review, correct and resubmit any erroneous data. Contributors cannot contact application maintainers during deployment, as this may inadvertently reveal information (for example, certain errors may only occur when the input meets certain conditions). We attempt to increase the contributors' familiarity with the application and to decrease their need for support during its deployment by holding training sessions (as described in Section 2.4).

Benefits of Participation We also note that output privacy (informally speaking) increases with the number of contributors, as the output is an aggregate of the inputs (e.g., in the pathological case of a single contributor providing input, that input is necessarily leaked by the output). Thus, the simplicity and accessibility of the framework, as well as the comprehensibility of the underlying cryptographic tools indirectly contribute to the overall security of the protocol by encouraging participation.

4 Usability Challenges within our Use Cases

Our application requirements and deployment scenario posed specific usability challenges, in addition to the general usability challenges described in Section 3. We describe our solutions to these challenges throughout this section.

4.1 Communicating MPC

In order to convey MPC's security guarantees to non-experts, we devised various analogies for additive secret sharing that constitute a possible workflow they can replicate on their own. While our evidence is purely anecdotal, we surmised that some explanations worked better than others. In an early example, we attempted to demonstrate the process of splitting values into shares, which was visually represented by dividing bars into smaller bars and then reassembling them with the pieces of others. This limited us to using only positive values (negative space is difficult to represent), and in turn falsely gave some participants the impression that we would leak the lower bound of their data. In another example, and partly as a response to the lower bound question, we used clocks with the summation of a random value to explain the concept of finite fields or modulo. This in turn raised questions about the process of joining multiple clocks together, and how the actual data would not get lost in the process.

One analogy that did appear to resonates with our target audiences, and could be explained outside of a presentation, is describing the process as *lying about your salary* to one party, and letting another party know *by how much you lied* (*i.e.*, an offset); neither number reveals your actual salary to either party. But if multiple contributors give their lies and offsets to the two parties, each party can tally what they have (either lies or offsets) and then subtract one sum from the other to obtain a total of the original values.

The analogies are not meant to explain intricate details of MPC, but to give the audience confidence that it is possible to compute a function without revealing private inputs to those performing the computation. We estimate that about 500 people viewed our presentations, with attendees present at training sessions, conference talks, academic events, corporate conference calls, and other venues. While we cannot claim to know the total number of people whose minds were changed as the direct result of our presentations, both (1) personal feedback after the sessions and (2) a marked uptick in BWWC Compact Signers who indicated their willingness to contribute data indicate the effectiveness of our communication efforts.

We accompany our explanations with a diagram of our protocol that includes the public-key cryptography required to allow one enabling party to handle all communications. Figure 1 accurately reflects the MPC protocol used in the first iteration of the platform, as additive secret sharing met the basic analytic needs (*i.e.*, averages) and was straightforward to explain. Our implementation was open-source and available for anyone to audit. In later deployments, analysts specified more complex analytics (*e.g.*, deviations and longitudinal analysis of specific cohorts) that required a general-purpose MPC library that relies on Shamir's Secret Sharing [49].

Shamir's Secret Sharing is more complex than the additive secret sharing scheme described initially (*e.g.*, it relies on properties of polynomials over a finite field). We found that non-experienced personnel were willing to have more confidence in this scheme after being exposed and familiarized with simpler variants like additive secret sharing. They were more receptive to the idea that any function can be computed on private data once they understood how *some* functions could be computed. This appreciation increased the willingness of several participants to contribute sensitive data despite an initial unwillingness to do so.

4.2 Usability Requirements

Our application scenarios involve individuals with a wide range of technical backgrounds utilizing computing resources that are outside our control and governed by a variety of organizational constraints. Thus, application usability is critical. For a starting point grounded in the literature, we referred to the five usability components defined in seminal work by Jakob Nielsen and other human factors practitioners [39]. However, we found that these five components were insufficient to fully and faithfully characterize our usability requirements. This is due to the distinct usability challenges associated with deploying MPC both in general and in our specific application. Instead, we present a novel categorization of usability requirements that we believe is more suitable to our MPC application. Where possible, we indicate how our category relates to those defined by Nielsen. We recognize that our requirements may not be well-suited for evaluating the usability of privacy-preserving software in general; that is a broader issue outside of the scope of this paper.

Error Minimization Errors are particularly problematic in our setting, as MPC's encoding of data prevents us from detecting or sanitizing bad inputs. In addition to allowing users to copy and paste the data (rather than use a more error-prone manual entry process), we proactively provide feedback to warn users about missing or spurious (*e.g.*, out of range) data prior to submission (see Figures 4 and 8). We also compel users to consciously confirm that their submissions do not have errors by requiring that they click a checkbox before submission attesting to this fact. Additionally, we permit contributors to resubmit their data if they discover that a previous submission was corrupted due to human error or software failure. In general, supporting re-submission influences the design of the underlying MPC protocols, since not all MPC protocols can support it without modification.

Asynchronous Participation In traditional usability, this can be viewed as a component of *subjective satisfaction* as defined by Neilsen. However, it is particularly important for our MPC application. We used software support for asynchronous participation in order to satisfy the logistical challenge of scheduling a data analysis effort that involves numerous enterprises of various sizes. This requirement has significant implications when employing MPC because it dictates which MPC protocols can be used. In fact, this requirement was a factor in our decision to design a new MPC software framework. When using our software, contributors only need to be online while entering their data, and the analyst only needs to be online at two points in the protocol: to start the process and to compute the analytic. The analyst additionally needs to store and safeguard one piece of information (an RSA private key) between these two points in time. Finally, the analyst has the ability to see how many submissions have taken place on their interface; this feedback reassures them that the application is being used successfully by contributors.

Ease of Use This umbrella component includes the *learnability*, *efficiency of use*, and *memorability* requirements as defined by Neilsen. These requirements have significance in MPC use cases only in that addressing them can reduce submission errors; their significance otherwise is primarily determined by their importance in informing the design of *any* data entry system. Our application meets these requirements: it relies on familiar web interfaces, and requires (on the part of the contributor and of the analyst) no setup process,

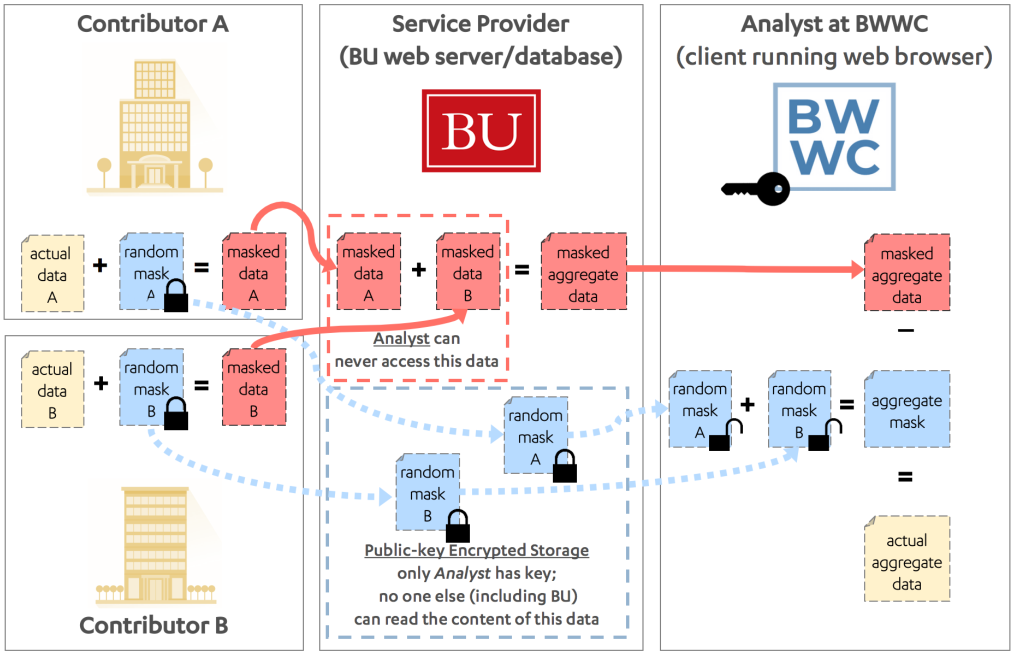


Figure 1: Slide detailing the MPC protocol. This was used in explanations to HR employees, lawyers, CEOs, and so on.

no specialized software or hardware, and no management of public internet addresses. Additionally, we provide users with an Excel spreadsheet to complete ahead of time; this sheet requires the same information as the web application, in the same format. Users can simply copy and paste the information from one source to another, or import the spreadsheet directly into our interface. Even when the UI is improved, the spreadsheet template and design remain fixed between deployments, improving the memorability of our system (especially because companies use it to recurringly provide their salary data).

Comprehension and Trust MPC technology can be difficult to explain to non-experts. It is critical to ensure contributors understand and trust the security guarantees of MPC, as this can increase the rate of participation. This challenge is unique to privacy preserving technologies as described in Section 3.1, and is not emphasized in Neilsen's requirements. Section 4.1 describes our approach: we initially chose a protocol that is easier to explain, accompanied by analogies and visualizations to improve comprehensibility for non-experts. Finally, our implementation is open-source to inspire trust.

4.3 Workflow Design

Given that the purpose of this application is to allow a group of non-domain expert participants to execute a session of a secret sharing scheme, we had to make several design choices that increase the usability of the system without compromising its security.

Experience of Contributors The application automates almost all portions of the protocol. The analyst must distribute the session identifier and participation codes to the participants (*e.g.*, via email). This ensures that the participants' experience is simple and error-resilient: they can simply click on a link (without entering participation codes or long identifiers manually). Because the analyst sends these links via an external channel, the service provider cannot see any correlation between participation codes and individual contributors. The service provider does not allow the analyst to see which participation codes were used to submit and which were not; only a count of contributions and their submission times are shown.

Participants must enter data either through manual entry or by pre-populating a formatted spreadsheet that has been provided to them and importing the file into their web browser. Since realistic scenarios involve not one but a collection or table of labeled numeric quantities from each participant, the software application actually implements the protocol in parallel on multiple labeled fields within a table.

Experience of Analysts The analyst starts a session by clicking a single button in the analyst interface and saving the 2048-bit private RSA key (unique to that session) to their hard drive. This interface also enables the creation of cohorts and the generation of participation links for contributors. Finally, it includes a session tracker that displays an anonymous participant submission history, as depicted in Figure 6. Their second interface, the final unmasking page, computes the final aggregate data upon successful submission of the analyst's private RSA key associated with the session.

Due to the role of the analyst in our system, the usability of their interface is not as critical as the usability of the contributors' interface. This is partly due to the fact that the analyst does not provide numerical inputs, and thus is not responsible for any error-prone input tasks that can affect the quality of the computation. Additionally, analysts had constant interactions with us as we developed the application, and thus have more experience navigating and using it.

If too few participants have submitted their data (the minimum number of participants can be configured) the service provider will not allow the analyst to compute the final aggregate data. Once the final aggregate is computed, it is displayed in the same familiar table format as the input table presented to individual participants.[1]

5 Usability Evaluation

We conducted two kinds of common usability evaluations, a Heuristics Evaluation and a System Usability Scale (SUS). We used both to evaluate our application's flow and interface. The results of the Heuristics Evaluation informed the subsequent redesign of our V1 interface into the V2 version. We discuss these evaluations and their results throughout this section.

Figure 4 illustrates the original web interface (prior to changes made based on the heuristics study) used for the BWWC study[2] as it appears within a web browser to each contributor. The participant interface provides a familiar spreadsheet table that an end-user can fill with data either manually or by pasting the data from another application. The email address is hashed on the client-side and this hashed value is used only as an index into the server database, allowing each participant to submit more than once in a session (overwriting their previous submissions).[3]

5.1 Heuristics Evaluation

After deploying our web application twice for the BWWC, a heuristic evaluation was conducted on V1 (displayed in Figure 4) to serve as an iterative design tool for addressing usability concerns. Heuristic evaluations involve having a set of evaluators examine the web application to judge its compliance with recognized usability principles (known as "heuristics"). More information can be found in Appendix A.

Results A total of 32 issues were identified. Each issue, along with its categorization and average severity rating, are shown in Tables 4 and 5. Two issues ("There is no indication as to whether the email address is valid" and "There is no email confirmation indicating that data was submitted") directly highlight the trade-offs between usability, correctness, and security discussed in Section 4.2. The security requirements preclude any server-side validation of the identity of a data contributor (because we impose the requirement that no individual participant can be identified by any part of the application) or any communications from the server to the client via another channel that requires their identity (such as email messages). At the same time, an error rate of zero is required, since any data entry errors lead to incorrect aggregate results. Thus, it is still necessary to allow users to resubmit data if they notice they made a mistake. The compromise was to create a unique identifier corresponding to the clients' submission.

Ten issues (2.1, 2.2, 2.3, 2.5, 4.1, 4.2, 4.3, 4.4, 4.5, and 6.2 in Table 4) are difficult to correct because the layout and format of the web application is based on the Equal Employment Opportunity Commission's (EEOC) Employer Information Report EEO-1 [1], which large-employer organizations are required to complete and file annually. Changing the web application to address these ten issues would lead to a mismatch with the EEOC Report, potentially confusing users and reducing memorability.

Some issues (1.5, 4.2, 4.3, 4.4, 4.5, 6.1, and 6.2) were addressed during in-person training and/or within training materials . Other issues were deemed out of scope for this project (3.1, 7.1), or no longer relevant with the introduction of new workflows.

Interface Redesign We used feedback from the heuristics evaluation to redesign our application's data submission interface prior to the 2017 deployment. The redesign divided the application into four distinct steps, each visually and logically separated by a card layout that clearly delineates varying steps in the submission process, and addressed the remaining issues from the evaluation. More details can be found in Appendix B.

5.2 System Usability Scale

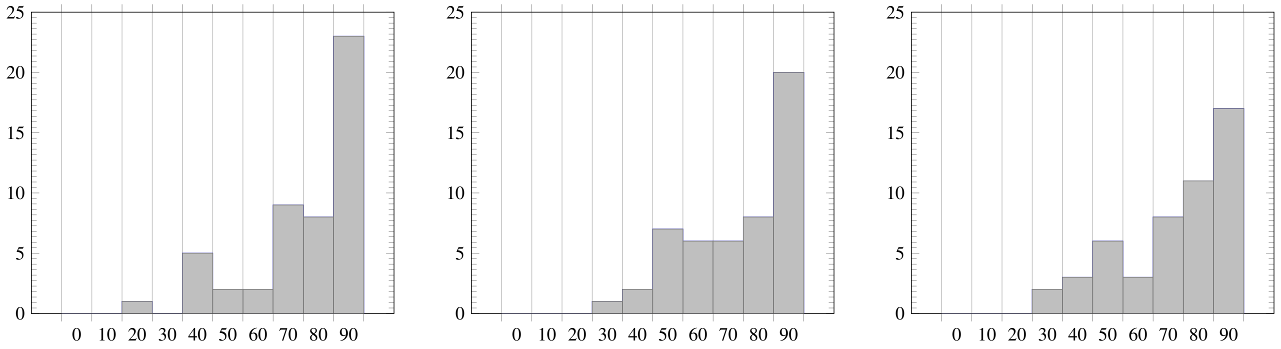
Prior to our recent deployments, we conducted a usability study on Amazon's Mechanical Turk (MTurk) Platform. The users were all directed to the Pacesetters Initiative data collection platform and asked to fill in the nine data fields with numerical data. Afterwards, users completed the System Usability Scale (SUS) questionnaire [16], comprised of 10 questions evaluating the application. Using the SUS allows us to measure our application's perceived usability with a relatively small sample size. It is also advantageous because it is quick for the participants to complete.

---

[1] It is the responsibility of the analyst to destroy their local copy of the private key after retrieving the result if this is the agreed-upon procedure. That is: assuming secure erasures, we achieve forward secrecy when the protocol is composed.

[2] The client application can be viewed at `https://100talent.org`.

[3] After submission, data remains visible in cleartext in the participant's browser so that any errors can be identified and a fixed set of inputs can be resubmitted. We relied upon briefings to inform participants about the post-submission encryption process.



(a) Group 1 SUS Scores (b) Group 2 SUS Results (c) Group 3 SUS Results

Figure 2: Histograms of SUS results, binned with width 10 on the x-axes and number of respondents on the y-axes.

Study Methodology To successfully submit data on the web page, users were required to:

1. Load the page
2. Fill out the nine fields of the form with numerical values of their choosing (either manually or by uploading a spreadsheet, depending on the experimental group)
3. Check a box to indicate the submission is correct
4. Correct any errors indicated by the user interface after the verification box is checked
5. Press the button labeled "Submit"

Unlike participants for the BWWC and GBCC initiatives, Mechanical Turk participants did not receive any prior training on the platform or an explanation of MPC.

Experimental groups Data sets were collected from three experimental groups:

- ***Group 1***: Users who interacted with V1 of the web page; V1 has the original appearance and interface of the platform as first deployed in 2015, configured with the data fields from the Pacesetters initiative as displayed in Figure 5. The users had to manually enter data into each of the cells as there was no import option.

- ***Group 2***: Users who interacted with the current design, V2, of the web page. These users were instructed to manually enter data into each of the cells. The drag-and-drop import box, as depicted in Figure 7, was removed to avoid confusion.

- ***Group 3***: Users who interacted with the current design, V2, of the web page, and were asked to fill out a spreadsheet on their local machine and upload it to the page via either the drag-and-drop or the file import feature.

For each group, 50 participants were recruited on the Amazon Turk Platform. Participants were paid $2.50 for completing the task, which at the beginning of the study was estimated to take 10 minutes. Previous studies have noted demographic differences between MTurk workers and the general population [34], specifically that MTurk workers tend to be younger and have lower incomes than the general population. Users are likely more familiar with common website interfaces, which may skew the SUS scores collected slightly higher than they might be for users of the BWWC and GBCC applications.

Results The results of the SUS Survey are intended to be a general indicator of a system's usability, rather than a diagnostic of specific system successes or failures. Histograms of the SUS scores by group are displayed in Figure 2. The mean and median SUS scores are displayed in Table 1. Group 1 scored highest, and all scores were above average compared to an analysis by Bangor et al. in 2009, which reports a mean SUS score of 68.2 for the over 3400 web pages surveyed [5]. However, one might speculate this comparison to Bangor et al. is limited due to improvements in web technology and design standards, as well as increased familiarity with web interfaces among the general population.

	Group 1	Group 2	Group 3
Mean	80.4	78.3	76.9
Median	85.00	83.75	82.50

Table 1: The mean and median of SUS scores for each experimental group.

In order to determine if there was a significant difference between the SUS scores per group, a Kruskal-Wallis one-way analysis of variance was performed on the SUS scores in each group [31]. The resulting p-value of 0.474 is not sufficient to reject the null hypothesis that there is no difference. Thus, although more recent user interfaces that we tested (Groups 2 and 3) scored slightly lower on the SUS scale, the scores do not indicate a statistically significant decrease in overall usability.

6 MPC for Improving Usability

Collecting user data on websites is an accepted best practice for understanding and improving their design and usability. Web analytics platforms such as Google Analytics and Mixpanel provide valuable feedback on application usage and facilitate data driven methods for improving usability features, accessibility, and design efficacy. These platforms are compatible with almost all web pages and provide valuable usage data, but also collect a multitude of identifying factors about users that expose information to both the site administrator and third-party platform providers [4]. While collecting information about how people interact with an application is helpful for interface enhancements, it is naturally at odds with the goals of an application designed to protect user privacy.

Our goal was to introduce a collection method for web usability metrics and usage data that did not compromise the privacy goals of the original application. Previous deployments of Web-MPC lacked a method to analyze how users are interacting with the application, as we believed collecting this data could undermine the privacy of the users and their trust in the system.

7 Collection of Usability Data via MPC

Secure web analytics data collection via MPC was integrated into the implementation of Web-MPC. Client-side analytics were collected through the existing MPC protocol, meaning that no new cryptographic protocols or libraries were required to collect and analyze the data. The usability data sets were masked in the same manner as the application data (salary data for BWWC and spending data for GBCC). The masked values are also sent to the *service provider* while the masks were encrypted and sent to the *analyst*. The metrics are revealed in aggregate form to the analyst in the same way as the application data sets that are contributed, meaning that no piece of information can be tied to a specific user. No modifications were made to the submission workflow itself or the underlying protocol detailed in Appendix C.

7.1 Usability Metrics

The following metrics were measured on the client side and submitted via MPC. Metrics are captured either on entry time (when a user completes a field) or on submission (when a user clicks the submit button); all metrics are only communicated to the server upon successful submission.

Time spent measures the number of milliseconds a user spends on various areas of the user interface. It is captured for each card of the layout: session area (cf. Fig. 7, table area (cf. Fig. 8, submission area (cf. Fig. 9). In the current UI, time spent is also tracked for the entire table area and the review area.

Data Prefill is a metric that measures if a user filled out the data using a spreadsheet and then submitted it by either importing or dragging and dropping the spreadsheet.

Validation errors refer to any error encountered by the participant while interacting with the user interface as enumerated below.

- ***Session info error*** occurs when a user enters an invalid session key or participation code.
- ***Empty cell error*** refers to any occurrence in which the user leaves a table cell empty during manual entry.
- ***Invalid cell error*** refers to any occurrence in which the user does not enter integers into the table cell during manual entry.
- ***Submission cell error*** measures whether there are any remaining empty cell or invalid cell errors when a user clicks the submit button. It does not count the number of errors remaining. This metric increments by 1 each time a user attempts to submit with remaining errors.
- ***Unchecked error*** occurs when a user attempts to submit data without first verifying that they have double-checked the values they have entered.
- ***Server error*** is captured if a user attempts to submit but is unsuccessful and encounters a status of 0 or 500, indicating a server-side error.
- ***Generic submission error*** is captured if a participant attempts to submit but is unsuccessful and receives a status of anything other than 200, 0, or 500.

8 Results and Analysis

The usability study ran over a three-day period and had 150 participants from Mechanical Turk. Of those 150 participants, 143 were able to successfully submit data with a total number of 200 submissions made (including re-submissions). Although 7 participants did not successfully submit data, all 150 participants completed the SUS survey (discussed in Section 5.2).

We emphasize that the results and success of the user study serve as a proof-of-concept for MPC-enabled web-analytics collection, and demonstrates the general feasibility of MPC as a tool to inform and improve application usability.

8.1 Usability Metrics Results

The usability metrics collected during use of the application are presented in Table 3, which contains the breakdown of errors encountered by users, and Figure 3, which presents the average time spent on the page by the different user groups.

Group	# Participants	# Submissions	# Re-Submissions
1	50	72	22
2	49	69	20
3	44	59	15

Table 2: Total number of successful participants and submissions by group.

Validation Error	Group 1	Group 2	Group 3
Empty cell	0.56 (28)	0.20 (10)	0.07 (3)
Invalid cell	0.04 (2)	0.02 (1)	0.00 (0)
Submission cell	0.08 (4)	0.16 (8)	0.05 (2)
Unchecked	0.02 (1)	0.04 (2)	0.00 (0)
Session info	0.00 (0)	0.00 (0)	0.00 (0)
Server	0.00 (0)	0.00 (0)	0.00 (0)
Generic submission	0.00 (0)	0.00 (0)	0.00 (0)

Table 3: Average errors per user for each experimental group. The number of participants was 50, 49, 44 for Groups 1, 2, 3, respectively. The absolute number of total errors encountered by group is listed in the parenthesis.

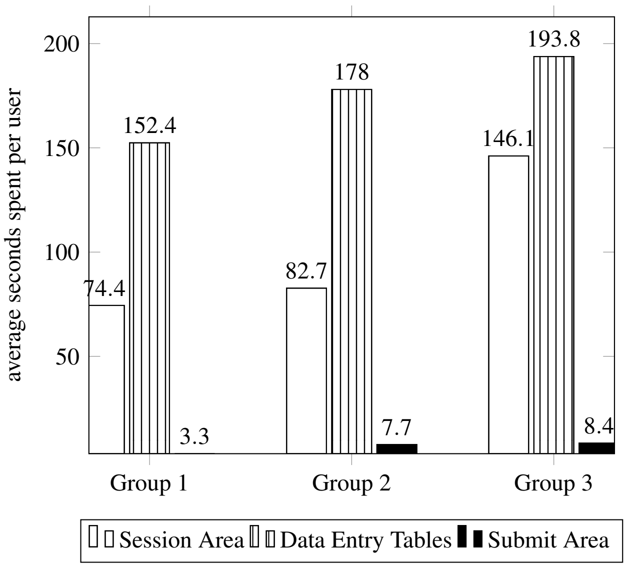


Figure 3: Average time spent per participant in seconds on the individual data entry tables by group.

8.2 Analysis

First, the results of the study confirm the feasibility of using MPC to understand and improve application usability. This could be achieved by introducing an MPC library to an existing web application in order to collect data in a privacy preserving way, or, more reasonably, by adding usability data analysis features to applications that already employ MPC.

With regard to Web-MPC, the results of the study suggest that the usability enhancements made to the application led to fewer data entry errors, even though some participants experienced difficulties with importing pre-filled spreadsheets. Only 21 of the 50 participants in this cohort chose to upload their pre-filled spreadsheets, which may account for the lower SUS results. It is expected that there would be fewer submission cell errors for Group 3, as their data would have been already filled out ahead of time. This should reduce the number of errors due to an empty cell or an invalid cell, as well. Even though only 21 participants imported spreadsheet files, there were only 2 total submission cell errors; this implies that fewer users encountered difficulties at submission time.

Although participants in Group 1 had a slightly higher SUS score, they encountered more errors in almost all categories compared to both Group 2 and Group 3. The version of the interface they used highlighted cells with errors in red but did not have tooltips explaining what the error was, which may have made it more difficult for participants to understand how to fix their errors. As shown in Figure 8, the V2 submission page provides additional instructions for participants compared to the V1 page. The additional instructions explaining what red and yellow cells indicate may have assisted these users in fixing their errors and could have prevented them from making the same mistake in subsequent cells.

8.3 Limitations

A key limitation of the process for gathering usability metrics is that information is only collected upon submission, when the secret sharing scheme is initiated. Thus, we cannot obtain metrics from participants who may have had too much trouble with the platform to successfully submit any data. In Group 1, all 50 participants successfully submitted data and metrics were captured for all of them. For Group 3, 6 participants were unable to complete the data submission process. Their contributions were therefore omitted from the final aggregate results. To account for this, the SUS survey was completed by all participants regardless of their ability to successfully contribute data.

Another limitation is that we are not able to link usability metrics back to individual users (as per the security guarantees of the MPC protocol). While this ensures user privacy, it limits the ability to gauge variance between the results. It is possible that all errors are produced by a single outlier participant in each group.

For each of the experimental groups, there were multiple resubmissions of data. Each time a re-submission occurs, the data corresponding to the participant is overwritten. This includes all of the data capturing the participant's submission errors and usage data. Thus, we only have the usability metrics from each participant's most recent submission, omitting their initial behavior with the platform. This may understate the number of errors encountered because we cannot analyze client-side usability metrics from previous data submissions.

As empty and invalid cell errors are captured upon manual entry, the results for Group 3 emerge either from participants who manually submitted or from those who attempted to fix cells, adding errors after importing a spreadsheet.

9 Future Work

The integration of usability metrics into a MPC protocol allows us to inform ongoing iterations of the user interface design using data from both usability studies and real-world deployments. We plan to collect usability data from participants in future deployments of the application with the BWWC and the GBCC. We also plan to continue expanding the number of usability metrics we collect.

As previously discussed, client-side usability metrics are only secret-shared upon successful data submission. We would like to gather usability data from all submission attempts. This method for gathering usability metrics has helped us mitigate the tension between the need to gather data on user behavior and the need to ensure the privacy of user activity and inputs. We intend to generalize this solution into a standalone plugin that can be used to allow other web services to obtain web analytics in a privacy-preserving way.

10 Lessons Learned

In order for privacy-preserving applications to be deployed and used to analyze sensitive data, the relevant stakeholders need to be convinced of the privacy guarantees of the underlying protocols while being insulated from the nuances of their implementation. Although there are a broad array of MPC protocols available, our initial use of additive secret sharing enabled relatively straightforward explanations of MPC. Demonstrating how MPC works using a simple, concrete function increased confidence among potential users that more complicated functions may be computed securely.

The user interface was designed in a manner that was familiar for participants, which also aided their ability to participate. Although participants may not have initially been familiar with the cryptographic techniques used, they were familiar with the task of filling out cells in a spreadsheet. The rest of the details of the protocol were abstracted away.

As our experimental results demonstrate, incorporating the improvements identified via the heuristic evaluation reduced the amount of human error during data entry. This shows that deployments such as these may be well-served if they are executed as joint collaboration between security engineers (who can design a technically sound framework) and human factors experts (who can improve UI/UX features before, during, and after deployment). Such collaboration between application developers and human factors experts is particularly important for privacy-preserving applications with which users have limited opportunities to interact: the applications must be usable even when users encounter them for the first time.

Gathering user behavior data in privacy-preserving applications may leak information, or create side channels for identifying or linking inputs to clients. However, MPC itself can be used to gather these metrics while still maintaining the guarantee that individual inputs are not revealed. The results from our usability study show that a variety of client-side metrics can be collected at the aggregate level to guide future usability improvements. This also demonstrates the relative ease with which this can be done: no new cryptographic tools or protocols (beyond those already necessary for the application) were built for this usability study.

Data sanitization, error detection, and error recovery cannot be achieved through manual inspection of secret-shared input data, and one single erroneous input may corrupt the entire result of an aggregate computation. Recovery from such errors is difficult and at worst impossible: it may require the execution of expensive MPC recovery protocols over the secret-shared data. This is further complicated by the fact that contributors must consent to the execution of such additional protocols. Thus, it is important that extensive error checking be performed on the client side within the application before data sets are contributed. Even when the contributed data set consists of analytics or usability metrics, such cleaning cannot be done ad hoc after the computation and must be encoded into the MPC protocol before deployment occurs.

The prohibitive cost of sanitizing the data after submission also makes it critical that contributors can resubmit (or withdraw) their data. In our last deployment, we received an email from a participant inquiring if resubmission is possible, as they had submitted random data to try the application out. Without resubmission, the output would have been corrupted in its entirety.

Finally, we have found that our experience echoes and confirms thoughts expressed by other researchers in the community on the development of real-world MPC applications [53]: "choose an application, starting from a very real business need, and build the solution from the problem itself choosing the right tools, tuning protocol ideas into a reasonable solution, balancing security and privacy needs vs. other constraints: legal, system setting, etc."

Acknowledgments

This work is partially supported by the NSF (under Grants #1430145, #1414119, #1718135, and #1739000) and the Honda Research Institutes. We are grateful to the Boston Women's Workforce Council and the Greater Boston Chamber of Commerce for their continued partnership in deploying MPC. We would like to acknowledge Rose Kelly for her work on the heuristics evaluation. We thank our shepherd, Nina Taft, for the exceptional care and thought that went into her advice, and our reviewers for their detailed and insightful feedback.

Availability

The entire codebase is open-source and available at: https://github.com/multiparty/web-mpc

References

[1] U.S. Equal Employment Opportunity Commission EEO-1 Survey. https://www.eeoc.gov/employers/eeo1survey/index.cfm. [Accessed: March 7, 2017].

[2] VIFF, the Virtual Ideal Functionality Framework. http://viff.dk/. [Accessed: August 15, 2015].

[3] Istemi Ekin Akkus, Ruichuan Chen, Michaela Hardt, Paul Francis, and Johannes Gehrke. Non-tracking web analytics. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 687–698, New York, NY, USA, 2012. ACM.

[4] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: User activity tracking for website usability evaluation and implicit interaction. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 203–212, New York, NY, USA, 2006. ACM.

[5] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *J. Usability Studies*, 4(3):114–123, May 2009.

[6] Rich Barlow. Computational Thinking Breaks a Logjam. http://www.bu.edu/today/2015/computational-thinking-breaks-a-logjam/. [Accessed: August 15, 2015].

[7] Johes Bater, Gregory Elliott, Craig Eggen, Satyender Goel, Abel N. Kho, and Jennie Rogers. SMCQL: secure query processing for private data networks. *PVLDB*, 10(6):673–684, 2017.

[8] Mihir Bellare, Viet Tung Hoang, Sriram Keelveedhi, and Phillip Rogaway. Efficient garbling from a fixed-key blockcipher. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 478–492, 2013.

[9] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 784–796, 2012.

[10] Assaf Ben-David, Noam Nisan, and Benny Pinkas. Fairplaymp: a system for secure multi-party computation. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 257–266. ACM, 2008.

[11] Dan Bogdanov, Liina Kamm, Baldur Kubo, Reimo Rebane, Ville Sokk, and Riivo Talviste. Students and Taxes: a Privacy-Preserving Study Using Secure Computation. *PoPETs*, 2016(3):117?135, 2016.

[12] Dan Bogdanov, Sven Laur, and Jan Willemson. Sharemind: A Framework for Fast Privacy-Preserving Computations. In Sushil Jajodia and Javier Lopez, editors, *Proceedings of the 13th European Symposium on Research in Computer Security - ESORICS'08*, volume 5283 of *Lecture Notes in Computer Science*, pages 192–206. Springer Berlin / Heidelberg, 2008.

[13] Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael Schwartzbach, and Tomas Toft. Financial cryptography and data security. chapter Secure Multiparty Computation Goes Live, pages 325–343. Springer-Verlag, Berlin, Heidelberg, 2009.

[14] Boston Women's Workforce Council. Boston women's workforce council report 2016. January 2017.

[15] Boston Women's Workforce Council. Boston women's workforce council report 2017. January 2018.

[16] John Brooke. Sus: a retrospective. *Journal of usability studies*, 8(2):29–40, 2013.

[17] Martin Burkhart, Mario Strasser, Dilip Many, and Xenofontas Dimitropoulos. Sepia: Privacy-preserving aggregation of multi-domain network events and statistics. In *USENIX SECURITY SYMPOSIUM*. USENIX, 2010.

[18] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explor. Newsl.*, 4(2):28–34, December 2002.

[19] Boston Women's Workforce Council. What We Do. accessed 10/10/2018.

[20] Boston Women's Workforce Council. Boston women's workforce council 2017, Januaray 2018.

[21] L.F. Cranor and S. Garfinkel. *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly Media, 2005.

[22] Ivan Damgård, Kasper Damgård, Kurt Nielsen, Peter Sebastian Nordholt, and Tomas Toft. Confidential benchmarking based on multiparty computation. Cryptology ePrint Archive, Report 2015/1006, 2015. http://eprint.iacr.org/.

[23] Daniel Demmler, Thomas Schneider, and Michael Zohner. Aby-a framework for efficient mixed-protocol secure two-party computation. In *NDSS*, 2015.

[24] Yael Ejgenberg, Moriya Farbstein, Meital Levy, and Yehuda Lindell. Scapi: The secure computation application programming interface. iacr cryptology eprint archive, 2012.

[25] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[26] Adam Groce, Alex Ledger, Alex J. Malozemoff, and Arkady Yerukhimovich. Compgc: Efficient offline/online semi-honest two-party computation. *IACR Cryptology ePrint Archive*, 2016:458, 2016.

[27] Marcella Hastings, Brett Hemenway, Daniel Noble, and Steve Zdancewic. Sok: General purpose compilers for secure multi-party computation. In *SoK: General Purpose Compilers for Secure Multi-Party Computation*, page 0. IEEE, 2019.

[28] Wilko Henecka, Stefan Kögl, Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. TASTY: tool for automating secure two-party computations. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 451–462, 2010.

[29] Ayman Jarrous and Benny Pinkas. Canon-mpc, a system for casual non-interactive secure multi-party computation using native client. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, pages 155–166, New York, NY, USA, 2013. ACM.

[30] Ronald Kainda, Ivan Flechais, and AW Roscoe. Security and usability: Analysis and evaluation. In *2010 International Conference on Availability, Reliability and Security*, pages 275–282. IEEE, 2010.

[31] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

[32] Andrei Lapets, Nikolaj Volgushev, Azer Bestavros, Frederick Jansen, and Mayank Varia. Secure multi-party computation for analytics deployed as a lightweight web application. Technical report, Computer Science Department, Boston University, 2016.

[33] John Launchbury, Iavor S. Diatchki, Thomas DuBuisson, and Andy Adams-Moran. Efficient lookup-table protocol in secure multiparty computation. In *Proceedings of the 17th ACM SIGPLAN International Conference on Functional Programming*, ICFP '12, pages 189–200, New York, NY, USA, 2012. ACM.

[34] Kevin E. Levay, Jeremy Freese, and James N. Druckman. The demographic and political composition of mechanical turk samples. *SAGE Open*, 6(1):2158244016636433, 2016.

[35] Joanne Lipman. Let's Expose the Gender Pay Gap. `http://www.nytimes.com/2015/08/13/opinion/lets-expose-the-gender-pay-gap.html`. [Accessed: August 15, 2015].

[36] Chang Liu, Xiao Shaun Wang, Kartik Nayak, Yan Huang, and Elaine Shi. Oblivm: A programming framework for secure computation. In *IEEE S & P*, 2015.

[37] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996.

[38] Kartik Nayak, Xiao Shaun Wang, Stratis Ioannidis, Udi Weinsberg, Nina Taft, and Elaine Shi. Graphsc: Parallel secure computation made easy. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 377–394, 2015.

[39] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, San Francisco, CA, 1993.

[40] Jakob Nielsen and Robert L. Mack. *Usability Inspection Methods*. John Wiley & Sons, Inc., New York, 1994.

[41] Greater Boston Chamber of Commerce. Gbcc launches pacesetters initiative aimed at uniting the business community's response to economic inclusion, January 2018. Retrieved February 26, 2019 from `http://bostonchamber.com/about-us/media-center/gbcc-launches-pacesetters-initiative`.

[42] Greater Boston Chamber of Commerce. Pacesetters initiative, January 2018. Retrieved February 26, 2019 from `http://bostonchamber.com/programs-events/pacesetters`.

[43] Do Le Quoc, Martin Beck, Pramod Bhatotia, Ruichuan Chen, Christof Fetzer, and Thorsten Strufe. Privapprox: Privacy-preserving stream analytics. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pages 659–672, Santa Clara, CA, 2017. USENIX Association.

[44] Aseem Rastogi, Matthew A. Hammer, and Michael Hicks. Wysteria: A programming language for generic, mixed-mode multiparty computations. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, SP '14, pages 655–670, Washington, DC, USA, 2014. IEEE Computer Society.

[45] Scott Ruoti, Jeff Andersen, Daniel Zappala, and Kent E. Seamons. Why johnny still, still can't encrypt: Evaluating the usability of a modern PGP client. *CoRR*, abs/1510.08555, 2015.

[46] Martina Angela Sasse, Sacha Brostoff, and Dirk Weirich. Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security. *BT technology journal*, 19(3):122–131, 2001.

[47] Berry Schoenmakers. Mpyc: Secure multiparty computation in python, 2018. https://www.win.tue.nl/berry/mpyc/.

[48] Axel Schroepfer and Florian Kerschbaum. Demo: Secure computation in javascript. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pages 849–852, New York, NY, USA, 2011. ACM.

[49] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

[50] Nikolaj Volgushev, Malte Schwarzkopf, Ben Getchell, Mayank Varia, Andrei Lapets, and Azer Bestavros. Conclave: secure multi-party computation on big data. 2019.

[51] Alma Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, SSYM'99, pages 14–14, Berkeley, CA, USA, 1999. USENIX Association.

[52] Ka-Ping Yee. Aligning security and usability. *IEEE Security & Privacy*, 2(5):48–55, 2004.

[53] Moti Yung. From mental poker to core business: Why and how to deploy secure computation protocols? In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 1–2, New York, NY, USA, 2015. ACM.

A Heuristic Evaluation

Two evaluators, both of whom were familiar with general human factors and usability principles, conducted the heuristic evaluation. The evaluators were familiar with the web application, but had not been involved in its development or deployment. Each independently examined the web application and listed general usability concerns. Next, they emulated users and attempted to submit data. While emulating the user tasks, they documented any issues they found. Then, they independently categorized the issues into the ten heuristic categories listed in Tables 4 and 5 [39,40].

After each issue was placed into a category, each evaluator gave each issue a severity rating according to the frequency, impact, and persistence of the issue. The severity ratings ranged from 1 to 5, with 1 representing the lowest severity and 5 representing the highest severity. Next, the evaluators compiled a list of potential solutions. Lastly, the evaluators met with the web application designers to discuss the feasibility of implementing the solutions.

B User Interface Redesign

The redesign divides the application into four distinct steps, each visually and logically separated by a *card layout*. Cards adhere to a pattern of *help text*, followed by a separator and the *action item*. The first card contains three input elements: session ID, participation ID, and spreadsheet (cf. Figure 7). Both the session and participation ID are validated when the user shifts the application focus from the input box. The final input element, the spreadsheet, triggers a browser-based parser for Excel files. This allows participants to simply drag-and-drop the pre-filled template into the web application, and reduce the possibility of copy/paste mistakes.

The second card contains the various tables for data entry, as displayed in Figure 8. It shows only the help text by default. To encourage participants to upload data rather than entering it manually, the card only expands after the spreadsheet has been provided. The tables were originally split by gender, and contained entries for financial information, length of service, and employee counts. User testing showed this caused confusion, so our new interface reverses the grouping. This resolves issues.

Validation of entries was enhanced for the 2017 collection. First, "semantic validation" across tables was added: if the employee count for a category in the first table is greater than zero, we also expect a matching salary and length of service greater than zero for that field in other tables. Clicking on any red cells shows a pop-up describing the problem. This resolves issues. The same validation is also present in the Excel spreadsheet, so participants get the chance to address any issues early on.

With the third card we introduce a new feature to collect anonymous user feedback about the data collection process. The responses will inform improvements for future data sessions. Answering questions is mandatory in this iteration, and unanswered questions turn red when attempting to submit.

The fourth and final card handles the actual data submission. In this card, shown in Figure 9, we show a summary of the employee count entered, a list of all errors messages that prevent a successful submission, and the history of both successful and failed submissions. A pop-up appears on submission to highlight the submission outcome.

C Secure Aggregation Protocol

In this section, we describe in full our protocol for secure aggregation within a finite additive group G such as $\mathbb{Z}/2^{64}\mathbb{Z}$. Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme that provides IND-CPA security; concretely, we use RSA in our implementation [37]. We restrict our attention to the case with a single analyst (as with the pay equity scenario). A single execution or *session* of the protocol proceeds in the following way:

Table 4: Listing of usability issues and average severity ratings categorized by heuristics 1 through 6; for the severity ratings, 1 is the lowest severity and 5 is the highest severity. Issues not fixed in the redesign are marked.

#	Usability Issue	Avg.	Fixed
1	**Visibility of system status: the web application should always keep users informed, through appropriate feedback within reasonable time.**		
1.1	There is no indication as how much of the table has been or is yet to be completed.	4.5	
1.2	There is no indication as to whether the session key is valid.	3.5	
1.3	There is no indication as to whether the email address is valid.	1.5	
1.4	After submission, user sees messages saying “loading” and then a confirmation window, which is confusing.	3	
1.5	After submission, there is no information indicating that data can be resubmitted.	3.5	
1.6	There is no email confirmation indicating that data was submitted.	2	
2	**Match between system and the real world: the web application should speak the user’s language and present information in a logical order.**		
2.1	The column and row headings do not use real-world terms that Human Resources (HR) uses, e.g., sum instead of total, workforce instead of employees, and mos. instead of months.	4.5	No
2.2	The tables are separated by gender, irrespective of whether HR data is usually separate by gender, e.g., if it is separated by ethnicity, it will be difficult for them to enter data separated by gender.	4.5	No
2.3	The table require a summation instead of an average, irrespective of whether HR data is given via averages or summations.	4	No
2.4	The columns requiring summary data (i.e., sum) are visually the same and not separated from data on raw numbers or monetary values.	2	
2.5	The sum cells require one to calculate totals by hand.	3.5	No
2.6	When you drag to select the same value for multiple cells, the cells are highlighted in red, implying an error.	3.5	
3	**User control and freedom: users will make mistakes and should be able to fix their errors easily. The web application should support undo, redo, and process cancellation.**		
3.1	A cell is highlighted in red if a user clicks there, does not input a number, then clicks somewhere else.	5	
3.2	Ctrl + Z (undo) is functional, but it always results in the previous cell being highlighted in red.	4	
3.3	The meaning of the red cell is unclear.	5	
3.4	Decimal points are not allowed in any cell.	4	
4	**Consistency and standards: users should not have to wonder whether different words, situations, or actions mean the same thing. The web application should follow platform conventions.**		
4.1	The terms #, $, and mos. are used in the row headings, but not the column headings.	3.5	No
4.2	The difference between multiple employee groups is unclear, e.g., executive versus mid-level.	5	No
4.3	There is no option for “other” employee, i.e., if they don’t fall into one of the employee groups.	5	No
4.4	While there is an option for 2+ races, not including Hispanic/Latino, there is not an option for 2+ races, including Hispanic/Latino.	5	No
4.5	Some employee types end with the word worker, but others do not.	2.5	No
5	**Error prevention: Reduce errors by reconfirming actions before they are carried out.**		
	No errors were detected.		
6	**The user should not have to remember information from one part of the dialogue to another. Instructions for use of the web application should be visible or easily retrievable whenever appropriate.**		
6.1	There is no objective or set of instructions indicating what and where information is to be entered as well as where users can find the session key or appropriate email address.	5	
6.2	There are no definitions of the terms, e.g., executive, mid-level, and annual compensation	5	No

Table 5: Listing of usability issues and average severity ratings categorized by heuristics 7 through 10; for the severity ratings, 1 is the lowest severity and 5 is the highest severity. Issues not fixed in the redesign are marked.

7	**Flexibility and efficiency of use: The web application should cater to both inexperienced and experienced users, allow users to tailor frequent actions, and provide defaults.**		
7.1	Though copy and paste works, if an empty cell is copied, the pasted cell will be highlighted in red, indicating that the copy and paste procedure did not work.	4	No
7.2	There is no way to enter functions into the cells, e.g., C2 = A2 + B2.	2	
7.3	You can only drag cells to copy values either horizontally or vertically, not both.	2.5	
8	**Aesthetic and minimalist design: dialogues should not contain information which is irrelevant or rarely needed.**		
8.1	Contrast between column and row fillings and text may be inadequate, i.e., black text on grey background may not be visible for some users.	4	
8.2	Red cells are inappropriate for those who are color blind, which is 8 percent of all males.	5	
9	**Help users recognize, diagnose, and recover from errors: error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.**		
9.1	There are no messages associated with the red cells.	5	
9.2	There are no messages associated with the grey cells.	3.5	
9.3	There is not a list of errors near the submit button that would indicate what needs to be fixed before submission is possible.	4	
10	**Help documentation: Documentation should be easy to search, be focused on the user's task, list concrete steps to be carried out, and be minimalist.**		
10.1	There is no help page, documentation, or instructions.	5	

1. The analyst initiates the process by generating a secret and public RSA key pair (s, p) and a unique session identifier $\mathsf{id} \in \mathbb{N}$, submitting p to the service provider, and sending id to all the contributors;[4]

2. Each of the n contributors possesses a secret *data* value $d_i \in G$ and does the following at least once[5]:
   (a) Generate a secret *random mask* $m_i \in G$ and calculate the *masked data* $r_i = d_i + m_i$,
   (b) Receive p from the service provider.
   (c) Send r_i and $c_i = \text{Enc}_p(m_i)$ to the service provider.

3. The service provider computes the sum of the masked data values to obtain the aggregate masked data quantity $R = \sum_{i=1}^{n} r_i$;

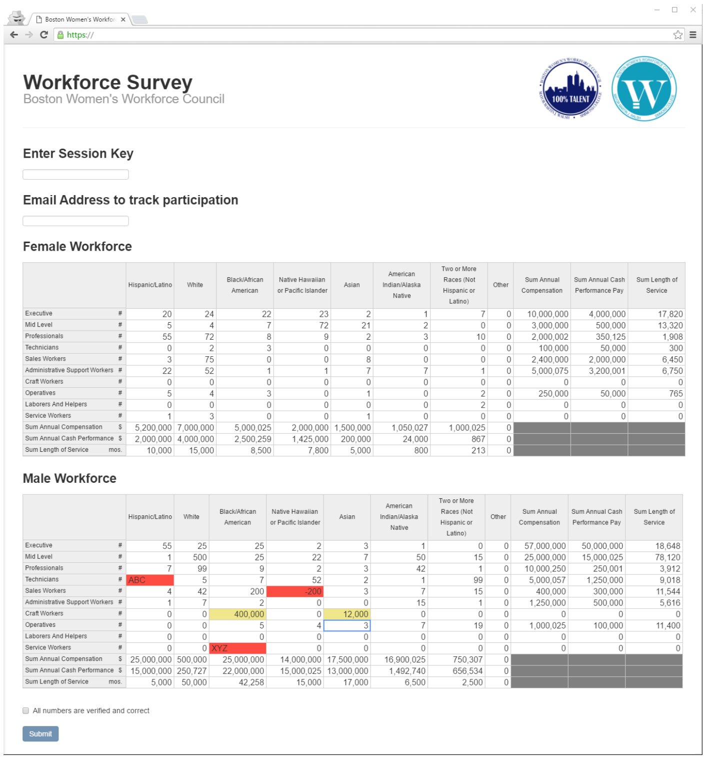
4. The analyst then retrieves R and all the $c_1, \ldots, c_n$ from the service provider, computes $m_i = \text{Dec}_s(c_i)$ for all i, computes $M = \sum_{i=1}^{n} m_i$, and obtains the final result $R - M = \sum_{i=1}^{n} d_i$. No other party receives any output.

Figure 1 illustrates an example deployment of the protocol with two contributors. Intuitively, this protocol is secure because the service provider's view of the random masks is protected using the analyst's public key, and the analyst never sees the individual masked data values unless it violates its promise not to collude with the service provider.

---

[4] The session identifier is only to allow distinct sessions, but it can serve another purpose: if no malicious agent possesses the session identifier, any data submitted by malicious agents will be ignored during the computation of the result.

[5] Each contributor can perform step (2) as many times as they wish before step (3) occurs; the operation they perform is idempotent if they always submit the same data.



Workforce Survey
Boston Women's Workforce Council

Enter Session Key

Email Address to track participation

Female Workforce

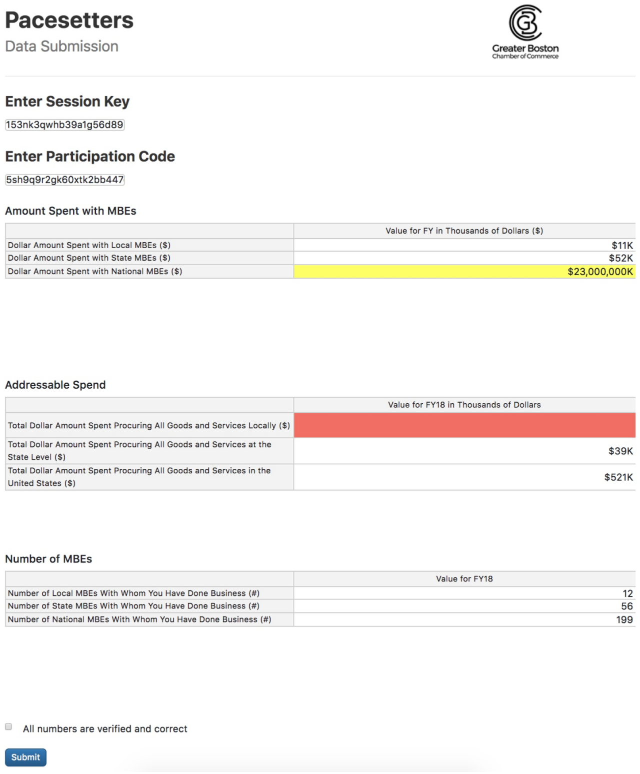
		Hispanic/Latino	White	Black/African American	Native Hawaiian or Pacific Islander	Asian	American Indian/Alaska Native	Two or More Races (Not Hispanic or Latino)	Other	Sum Annual Compensation	Sum Annual Cash Performance Pay	Sum Length of Service
Executive	#	20	24	22	23	2	1	7	0	10,000,000	4,000,000	17,820
Mid Level	#	5	4	7	72	21	2	0	0	3,000,000	500,000	13,320
Professionals	#	55	72	8	9	2	3	10	0	2,000,002	350,125	1,908
Technicians	#	0	2	3	0	0	0	0	0	100,000	50,000	300
Sales Workers	#	3	75	0	0	8	0	0	0	2,400,000	2,000,000	6,450
Administrative Support Workers	#	22	52	1	1	7	7	1	0	5,000,075	3,200,001	6,750
Craft Workers	#	0	0	0	0	0	0	0	0	0	0	0
Operatives	#	5	4	3	0	1	0	2	0	250,000	50,000	765
Laborers And Helpers	#	0	0	0	0	0	0	2	0	0	0	0
Service Workers	#	1	3	0	0	1	0	0	0	0	0	0
Sum Annual Compensation	$	5,200,000	7,000,000	5,000,025	2,000,000	1,500,000	1,050,027	1,000,025	0			
Sum Annual Cash Performance	$	2,000,000	4,000,000	2,500,259	1,425,000	200,000	24,000	867	0			
Sum Length of Service	mos.	10,000	15,000	8,500	7,800	5,000	800	213	0			

Male Workforce

		Hispanic/Latino	White	Black/African American	Native Hawaiian or Pacific Islander	Asian	American Indian/Alaska Native	Two or More Races (Not Hispanic or Latino)	Other	Sum Annual Compensation	Sum Annual Cash Performance Pay	Sum Length of Service
Executive	#	55	25	25	2	3	1	0	0	57,000,000	50,000,000	18,648
Mid Level	#	1	500	25	22	7	50	15	0	25,000,000	15,000,025	78,120
Professionals	#	7	99	9	2	3	42	1	0	10,000,250	250,001	3,912
Technicians	#	ABC	5	7	52	2	1	99	0	5,000,057	1,250,000	9,018
Sales Workers	#	4	42	200	-200	3	7	15	0	400,000	300,000	11,544
Administrative Support Workers	#	1	7	2	0	0	15	1	0	1,250,000	500,000	5,616
Craft Workers	#	0	0	400,000	0	12,000	0	0	0	0	0	0
Operatives	#	0	0	5	4	3	7	19	0	1,000,025	100,000	11,400
Laborers And Helpers	#	0	0	0	0	0	0	0	0	0	0	0
Service Workers	#	0	0	XYZ	0	0	0	0	0	0	0	0
Sum Annual Compensation	$	25,000,000	500,000	25,000,000	14,000,000	17,500,000	16,900,025	750,307	0			
Sum Annual Cash Performance	$	15,000,000	250,727	22,000,000	15,000,025	13,000,000	1,492,740	656,534	0			
Sum Length of Service	mos.	5,000	50,000	42,258	15,000	17,000	6,500	2,500	0			

☐ All numbers are verified and correct

Figure 4: The original contributor web interface used for the BWWC study at `https://100talent.org` as it appears within a web browser with some user errors highlighted.



Pacesetters
Data Submission

Enter Session Key

153nk3qwhb39a1g56d89

Enter Participation Code

5sh9q9r2gk60xtk2bb447

Amount Spent with MBEs

	Value for FY in Thousands of Dollars ($)
Dollar Amount Spent with Local MBEs ($)	$11K
Dollar Amount Spent with State MBEs ($)	$52K
Dollar Amount Spent with National MBEs ($)	$23,000,000K

Addressable Spend

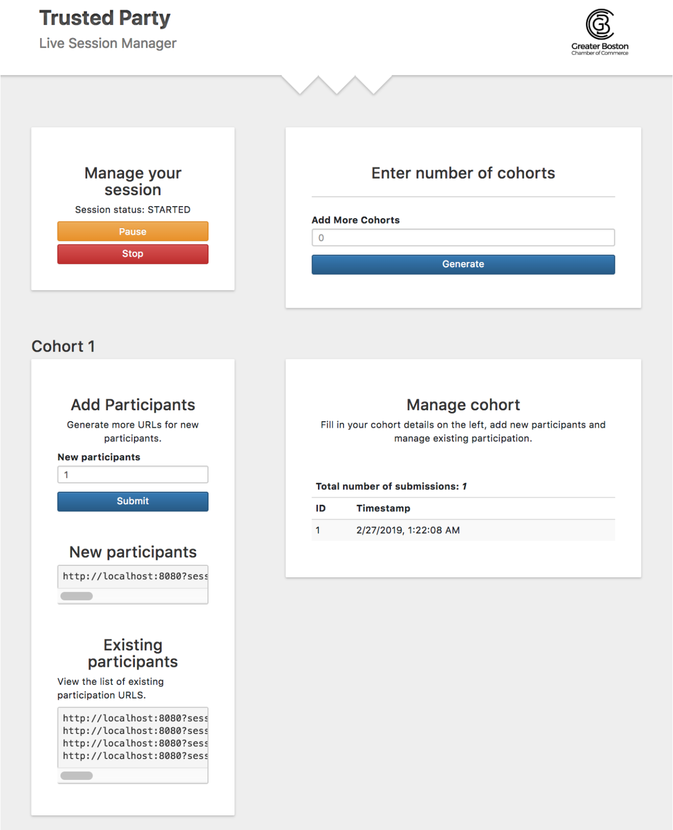
	Value for FY18 in Thousands of Dollars
Total Dollar Amount Spent Procuring All Goods and Services Locally ($)	
Total Dollar Amount Spent Procuring All Goods and Services at the State Level ($)	$39K
Total Dollar Amount Spent Procuring All Goods and Services in the United States ($)	$521K

Number of MBEs

	Value for FY18
Number of Local MBEs With Whom You Have Done Business (#)	12
Number of State MBEs With Whom You Have Done Business (#)	56
Number of National MBEs With Whom You Have Done Business (#)	199

☐ All numbers are verified and correct

Figure 5: The original interface modified for the Pacesetters Initiative. This was used by Group 1 in the usability study.



Manage your session

Session status: STARTED

Enter number of cohorts

Add More Cohorts: 0

Cohort 1

Add Participants

Generate more URLs for new participants.

New participants: 1

New participants

```
http://localhost:8080?sess
```

Existing participants

View the list of existing participation URLS.

```
http://localhost:8080?sess
http://localhost:8080?sess
http://localhost:8080?sess
http://localhost:8080?sess
```

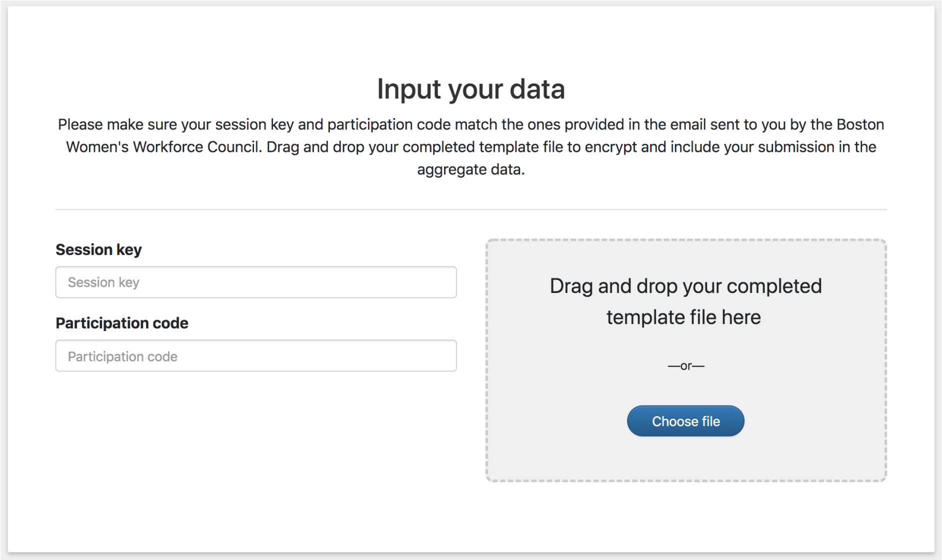
Manage cohort

Fill in your cohort details on the left, add new participants and manage existing participation.

Total number of submissions: *1*

ID	Timestamp
1	2/27/2019, 1:22:08 AM

Figure 6: Page for analyst to create cohorts, generation participation codes, and track anonymous submission history

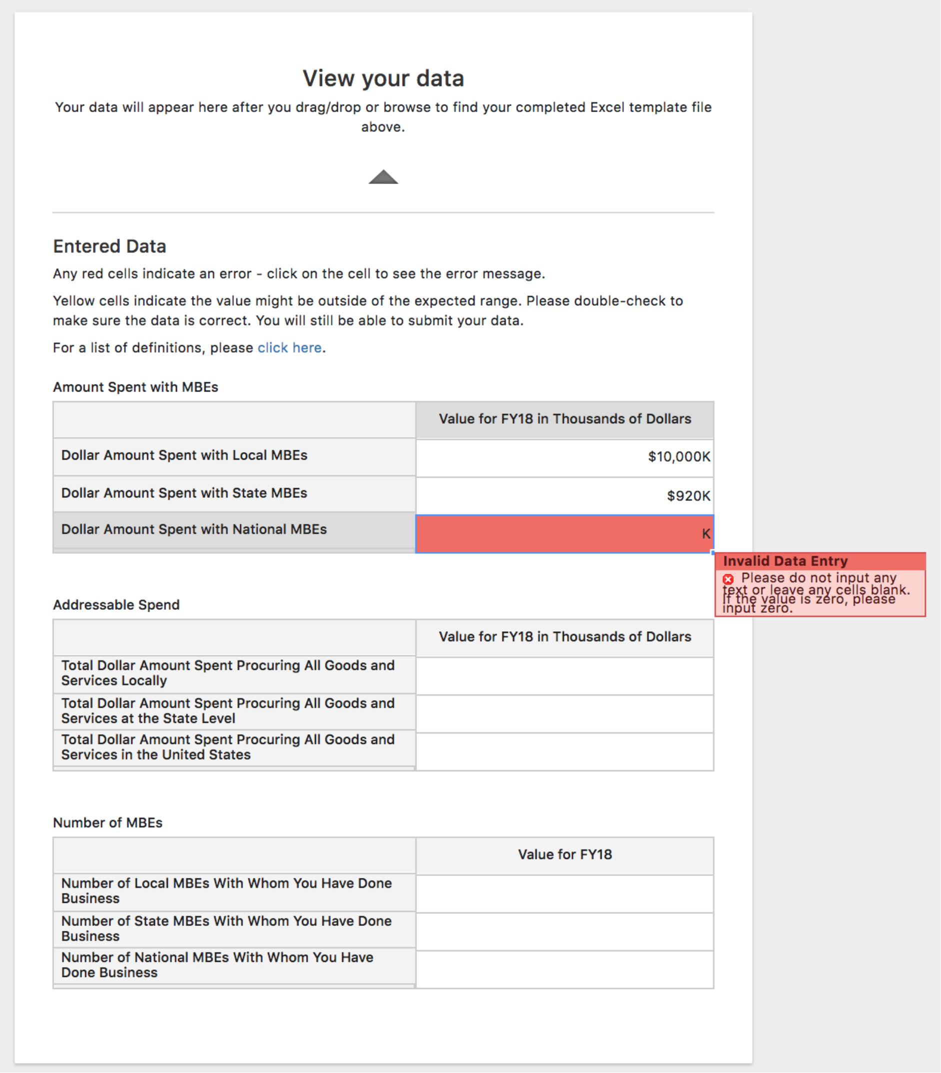


Input your data

Please make sure your session key and participation code match the ones provided in the email sent to you by the Boston Women's Workforce Council. Drag and drop your completed template file to encrypt and include your submission in the aggregate data.

Drag and drop your completed template file here

Figure 7: First card of the 2017 interface. To encourage drag-and-drop upload from an Excel file, this is the only card shown by default.



View your data

Your data will appear here after you drag/drop or browse to find your completed Excel template file above.

Entered Data

Any red cells indicate an error - click on the cell to see the error message.

Yellow cells indicate the value might be outside of the expected range. Please double-check to make sure the data is correct. You will still be able to submit your data.

For a list of definitions, please click here.

Amount Spent with MBEs

	Value for FY18 in Thousands of Dollars
Dollar Amount Spent with Local MBEs	$10,000K
Dollar Amount Spent with State MBEs	$920K
Dollar Amount Spent with National MBEs	K

Invalid Data Entry

Please do not input any text or leave any cells blank. If the value is zero, please input zero.

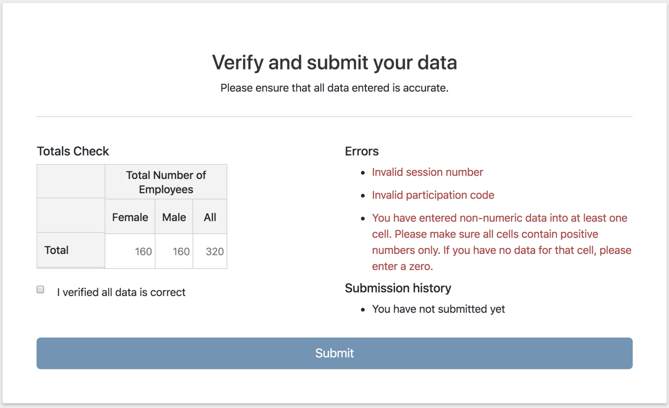
Addressable Spend

	Value for FY18 in Thousands of Dollars
Total Dollar Amount Spent Procuring All Goods and Services Locally	
Total Dollar Amount Spent Procuring All Goods and Services at the State Level	
Total Dollar Amount Spent Procuring All Goods and Services in the United States	

Number of MBEs

	Value for FY18
Number of Local MBEs With Whom You Have Done Business	
Number of State MBEs With Whom You Have Done Business	
Number of National MBEs With Whom You Have Done Business	

Figure 8: Submission card within the V2 interface. This example displays an empty cell and the corresponding tooltip.



Verify and submit your data

Please ensure that all data entered is accurate.

Totals Check

	Total Number of Employees		
	Female	Male	All
Total	160	160	320

☐ I verified all data is correct

Errors

- Invalid session number
- Invalid participation code
- You have entered non-numeric data into at least one cell. Please make sure all cells contain positive numbers only. If you have no data for that cell, please enter a zero.

Submission history

- You have not submitted yet

Figure 9: Final card within the V2 at `https://100talent.org`. In this example, the verification check has failed. Text boxes are still highlighted just as they were in the old interface (cf. Figure 4). Now, the full list of errors is co-located in this card.