Privacy-Preserving Randomized Controlled Trials: A Protocol for Industry Scale Deployment (Extended Abstract)

Mahnush Movahedi* Benjamin M. Case Andrew Knox Li Li

Yiming Paul Li Sanjay Saravanan Shubho Sengupta Erik Taubeneck

Facebook Inc.

ABSTRACT

We outline a way to deploy a privacy-preserving protocol for multiparty Randomized Controlled Trials on the scale of 500 million rows of data and more than a billion gates. Randomized Controlled Trials (RCTs) are widely used to improve business and policy decisions. A Randomized Controlled Trial is a scientifically rigorous method to measure the effectiveness of a treatment. This is accomplished by randomly allocating subjects to two or more groups, treating them differently, and then comparing the outcomes across groups. In many scenarios, multiple parties hold different parts of the data for conducting and analyzing RCTs. Given privacy requirements and expectations of each of these parties, it is often challenging to have a centralized store of data to conduct and analyze RCTs.

We accomplish this by a three-stage solution. The first stage uses the Private Secret Share Set Intersection (PS^{3}I) [3] solution to create a joined set and establish secret shares without revealing membership, while discarding individuals who were placed into more than one group. The second stage runs multiple instances of a general purpose MPC over a sharded database to aggregate statistics about each experimental group while discarding individuals who took an action before they received treatment. The third stage adds distributed and calibrated Differential Privacy (DP) noise to the aggregate statistics and uncertainty measures, providing formal two-sided privacy guarantees.

We also evaluate the performance of multiple open source general purpose MPC libraries for this task. We additionally demonstrate how we have used this to create a working ads effectiveness measurement product capable of measuring hundreds of millions of individuals per experiment.

1 INTRODUCTION

Randomized Controlled Trials (RCT) are considered to be one of the most reliable forms of scientific evidence since it reduces spurious causality and bias. United States Preventive Services Task Force has recognized *"evidence obtained from at least one properly randomized controlled trial with good internal validity"* as the highest quality evidence [7]. It is not uncommon in RCT that multiple parties hold the data needed for analysis. For example, one party knows how subjects are randomized into treatment and control groups, and the other party holds the *outcomes* measures that the treament is meant to affect. They wish to compare aggregate statistics of the treatment and control groups, without revealing their input data to the other party.

We design and implement Private RCT; a practical and scalable secure two-party computation system for calculating RCT results. Designing a scalable privacy-preserving solution is technically challenging, especially when using one cryptographic primitive in isolation, such as either Garbled Circuit (GC) or Homomorphic Encryption (HE). Our protocol combines a suite of underlying cryptographic primitives such as GC, HE, Private Set Intersection (PSI) and Secret Sharing, and incorporates them in a way that is more efficient than using any one secure computation technique by itself. Moreover, we demonstrate how to add distributed computing scaling techniques such as sharding without compromising privacy. Finally, we use differential privacy to add noise to the output of the computation and prevent leakage of the input data based on the revealed output.

1.1 Private Randomized Controlled Trials

Privacy-preserving computation (secure computation) is a cryptographic method that enables parties to jointly compute a function on each of their secret inputs while preserving the privacy of the inputs. The technology guarantees that the parties will only learn the designated output of the function and they cannot access or derive each other's inputs (also known as input privacy), any intermediate values, or statistical results.

In the context of RCTs, privacy-preserving computation is highly desirable as it enables statistical measurements without giving access to the raw/un-encrypted input databases to external parties. This creates stronger guarantees to enforce the individual expectation of privacy.

Take clinical trials for example. Medical researchers randomly assign subjects into treatment and control groups and

Conference on Digital Experimentation (CODE), 2020, MIT

*presenting

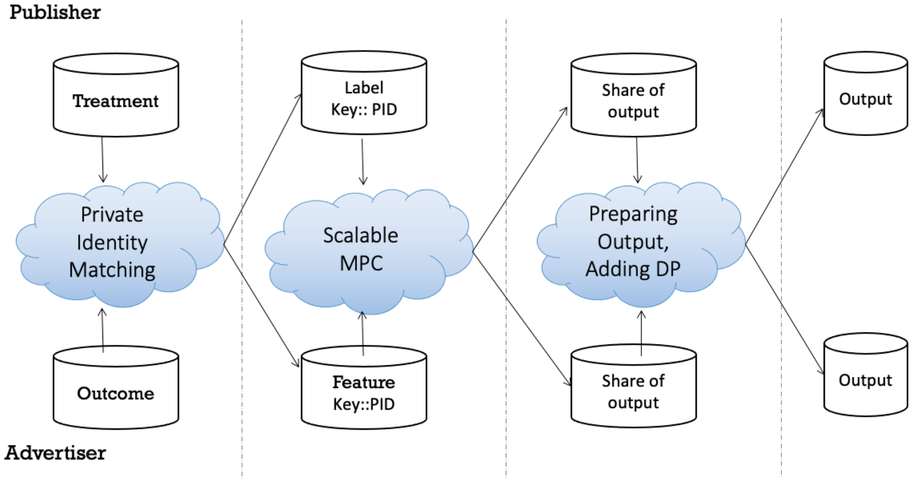

Figure 1: Private RCT, high level design

measure a few pre-determined health outcomes. While researchers can measure some demographic variables by themselves, much richer but usually unavailable demographic and behavioral data such as living conditions, mental status, social interactions, socio-economic status can be immensely valuable to improve statistical power, cut costs, and extract more credible insights from typically small-scale clinical trials. Unfortunately such auxiliary data is usually held by third parties such as hospitals, physicians, and government agencies. Private RCT solution makes it easier for these independent parties to collaborate in analyzing RCTs.

1.2 High-level Design

Private RCT has three main stages. The protocol is designed to make real-world deployment easy by modularizing the protocol and address their challenges individually. Figure 1 shows the overall design.

(1) *Private Identity Matching.* The first stage is to determine the union/intersection of users we want to compute on. Private Identity Matching also aligns the rows which the two parties have in common. We have proposed a couple of protocols for handling this identity intersection: PID and PS^{3}I [3]. The former computes a full outer join of the identities using a new Decisional Diffie-Hellman (DDH) based construction that is only twice as computationally expensive as standard DDH-based Private Set Intersection (PSI). PS^{3}I extends this to generating secret shares of attached data for only those identities in the intersection and uses DDH techniques and additive Homomorphic Encryption (HE). We also have ongoing work to extend PS^{3}I to handle identity matching in many-to-many scenarios.

(2) *Scalable Secure Computation.* In the second stage, we calculate a pre-defined linear circuit on the output of the previous step. Since we envision different use cases, which differ in their nature of calculation, the computation step should be able to compute any circuit on the joined data. We evaluated multiple general purpose MPC frameworks for this step and concluded to build our solution based on EMP-toolkit. We used sharding mechanism to scale it from handling 2 million rows of input data to 500 million rows. We use an XOR secret sharing mechanism to hide any intermediary result in this step.

(3) *Differentially Private RCT Output.* The last step is to prepare the output. Since the output is going to be revealed to both parties, it is important to ensure it does not break the privacy guarantees that the system promises. We need to check for access control, ensure rate limits, and finally add DP noise to the output. We calibrate DP noises to ensure two-sided privacy guarantee: participant A gets only a differentially private view of B's secret input, and vice versa. To provide differentially private confidence intervals of two-sample difference-in-means estimators, we examine several algorithms and compare their coverage, tightness, and computational cost within MPC. We also propose methods to generate DP noises distributively in the presence of semi-honest or malicious adversaries.

1.3 Our Contributions

By combining Decisional Diffie-Hellman (DDH), Homomorphic Encryption (HE), Secure Multi-Party Computation (MPC), and Differential Privacy (DP), we provide a private yet performant solution to enable two parties to collaborate in RCTs. Starting from the correct underlying cryptographic primitives, we show how to use them, scale to large data sets, and handle the challenges in practice. In addition to designing and implementing the Private RCT, we have the following contributions which can be of independent interest to the community:

(1) *Evaluation Framework for MPC.* We looked into multiple services that enable secure computation to decide on the best framework that meets the requirements of Private RCT. This systematization of knowledge on MPC protocols can be of independent value for researchers and industry cryptographers. We started with a comparison across more than twenty secure computation services — across secret sharing, garbled circuits, homomorphic encryption-based protocols. We chose ABY, EzPC, EMP-toolkit, Fancy Garbling, Scale-Mamba, Obliv-C and MPyC for a more detailed literature review — and from them we did performance testing on ABY, EMP-toolkit, and Scale-Mamba.
(2) *Scaling via Private Sharding.* Handling large volumes of data is important in practice. Unfortunately, most current secure computation platforms cannot handle large enough data sets at the required scale and speed imposed by our application. In this work, we designed and implemented a privacy-preserving sharding technique to make EMP scale to 500M rows. Our protocol does not reveal any intermediary information to the involved parties. ie., the intermediary result of computation on each shard will remain private. We will discuss the details of this protocol in section 2.3.
(3) *Private Conversion Lift.* To test the efficiency and scalability of our design, we implemented Conversion Lift as an application that uses Private RCT in the backbone. Conversion Lift compares the actions of users in randomized test and control groups to measure the additional business driven by the advertisement.

2 PRIVATE RCT PROTOCOL

In the following sub-sections, we provide detailed design for each step of the protocol.

2.1 Assumptions and Model

We consider two parties, the most common in real world scenarios. We assume each party has their own separate infrastructures that jointly participate in the 2PC protocol and there is no trusted third party involved. We assume a Semi-Honest (Passive) adversary: a corrupted party will memorize and use any information it can learn from the protocol, but will not deviate from the protocol specification. We do consider that an adversary may craft their inputs to maximize the information they can learn passively, as discussed in 2.4.

2.2 Private Identity Matching

Before computation can be done for Private RTCs the two data sets must be joined by some identities of the users. There are three matching cases we consider:

(1) **Single unique identifier.** The simplest situation is that each user has one identifier (e.g., email) which is unique to them in one of the party's input sets. We want to compute on users who have the same identifier in both party's input sets.
(2) **External identifier.** In this situation, there is a common identifier shared between the two parties and each party has a many-to-many relationship between their own user ids and the external ids.
(3) **Many identifiers.** Each user may have multiple identifiers (e.g., email and phone). This case also results in a many-to-many matching between users from both parties.

The paper [3] introduces two protocols for handling the case of a single unique identifier. Both protocols can compose with general-purpose MPC to enable arbitrary computation on the joined data. The first variant which we call Private-ID (PID), allows the parties to privately compute a set of pseudorandom universal identifiers (UID) corresponding to the records in the union of their sets, where each party additionally learns which UIDs correspond to which items in its set but not if they belong to the intersection or not. This new formulation enables the parties to independently sort their UIDs and the associated records and feed them to any general-purpose MPC that ignores the non-matching records and computes on the matching ones. The Private-ID protocol has the advantage that it only needs the identifiers from the records as input to produce the UIDs and hence for each application, parties can assemble a possibly new set of features/labels per identifier for the downstream computation without re-executing the protocol.

The second protocol from [3] called Private Secret Shared Set Intersection (PS^{3}I), is a natural extension of PSI where instead of learning the plaintext matched records, parties only learn additive shares of those records which they can feed to any general-purpose MPC to execute the desired computation on. The construction is based on efficiently extending existing DDH-based PSI using any additive homomorphic encryption scheme. The advantage of PS^{3}I over PID is that its output size and hence the complexity of the subsequent MPC is proportional to the size of intersection which in some cases is much smaller than size of union of the two original datasets. Its disadvantage, similar to prior work, is that full records and not just the identifiers need to be ready at the time of execution, and it requires a rerun when associated records change for the same identifiers. We have further generalized these protocols to work in matching scenarios (2) and (3) where the mapping is many-to-many. There are two ways to resolve these many-to-many mappings:

(1) **Resolve to a single match.** In the matching case where each user has many identifiers, we can define rules of priorities or weights so as to choose the best possible match. Both Private-ID and PS^{3}I can be naturally generalized to do this. In cases where the many-to-many mapping results from a shared external identifier, it is generally the case that each occurrence of the identifier is of equal value in defining a match, so the next collecting approach may be better.

(2) **Collect a many-to-one match.** In some applications of Private RCT it is actually best not to resolve to a single match but rather to collect all the matches on one side of a many-to-many matching. The PS^{3}I protocol generalizes more naturally to do this than Private-ID, and we call this variant Collecting PS3I.

One implication of a many-to-many identity matching is that it affects the RCT validity. If one party partitions users into Test and Control groups, it is possible that one user from the other party's set will match with users from both the Test and Control groups. We call such users contaminated and drop them from the study by running the collecting version of PS^{3}I on both the test and control groups simultaneously and looking for overlap in the encrypted intersections.

2.3 Private RCT Computation

We estimate that a large RCT, as is seen in advertising Conversion Lift, may routinely have up to 500M rows; thus handling a large volumes of data is crucial in practice. Unfortunately, EMP-toolkit and similar MPC platforms were not able to handle such a large volume of data at once in the required scale and speed imposed by the application. EMP-toolkit was limited to 2M rows of integer data when we tested on the largest available AWS Fargate option (30GB), although it was able to handle 4M rows on a larger EC2 instance (64GB).

In this section we go over our design for computing a RCT game privately across multiple containers via our privacy-preserving sharding mechanism. In our protocol, data from the private id match is deterministically partitioned into separate container shards. Each shard then performs a 2PC with the corresponding shard controlled by the other party. The intermediary output of each shard remains private, i.e. the garbled values will not be open to the parties. Instead, each party will learn an XOR share of the value. Consequently, the aggregation step happens after the data is reconstructed from the XOR-shares in a garbled circuit and only the final result will be revealed.

Figures 2 show the high-level design of the Protocol, as it is replicated on both the Garbler and Evaluator's sides. Both Garbler and Evaluator each have one Coordinator, multiple Workers and one Aggregator. The Coordinator partitions the input database into shards in a round robin method, and assigns each shard to a Worker. Workers will evaluate the sub-circuit on the input shard and send the intermediary result to the Aggregator who evaluates the final output and adds DP noise to it. For each Worker on the Garbler side, there is a corresponding Worker on the Evaluator side that have point-to-point connection with each other and work together to evaluate the garbled circuit. Similarly, the Aggregator on the Garbler side has network connection and computes the aggregation jointly with the Aggregator on the evaluator side.

2.3.1 *Private Partitioning and Aggregation.*

To ensure no party learns the output of computation on one shard, we do not reveal the intermediary output at the end of their game. Instead, The server will choose a random number as a new input and XOR the result of the computation with that random number in the garbled circuit. At the end of the game, Evaluator knows the result of computation XORed with that random number and Garbler knows the random number. Jointly, they can reconstruct the result but individually, they do not have any information about it due to the security of one-time pads.

2.4 Differential Privacy in MPC

Deterministic MPC output can leak input information. Typical privacy attacks on aggregated statistics include re-identification attacks, database reconstruction, and membership inference [6]. To further enhance privacy of the protocol, we use Differential Privacy (DP) to add randomness to the output. See [5] for common DP algorithms. We address three specific considerations to implement DP for Private RCT below.

2.4.1 *DP Confidence Intervals (CIs).*

Valid statistical inference with RCTs requires valid measures of uncertainty (e.g. Confidence Intervals). Valid CIs consider both the sampling variation and the randomness added by the DP noise. We consider three criteria for CIs. 1) CIs should be differentially private. 2) CIs should have the correct coverage probability. 3) CIs should be narrow, given the DP guarantee and the correct coverage.

There are several additional challenges to construct DP confidence intervals in Private RCTs. 1) Most proposed methods for DP confidence intervals aim at estimating unknown parameters from known parametric distributions, e.g. releasing the DP mean estimate and its DP standard error from a Gaussian distribution [1, 4, 8]. In RCTs we estimate the difference-in-means of two samples from unknown distributions. 2) Nonparametric methods for confidence intervals are typically based on resampling techniques such as bootstrapping, which can be computationally expensive in MPC [2].

We compare existing parametric and nonparametric approaches to DP confidence intervals, and found nonparametric bootstrapping methods are generally more robust to our application, due to wide variations across RCTs and vastly different expected treatment effect sizes.

2.4.2 *Two-side Privacy Guarantees.*

In Private RCTs, each party's input data is masked from the other party, thus DP noises need to be calibrated to ensure two-side privacy guarantee: Party A gets only a differentially private view of Party B's secret input, and vice versa [9]. Recall that data parties hold different parts of the data, for example, party A has the treatment and party B holds the outcome. The sensitivity of the calculation with respect to the treatment variable is larger than the sensitivity with respect to the outcome variable. It suggests that to protect party A's input data, more noise needs to be added to the final output exposed to party B.

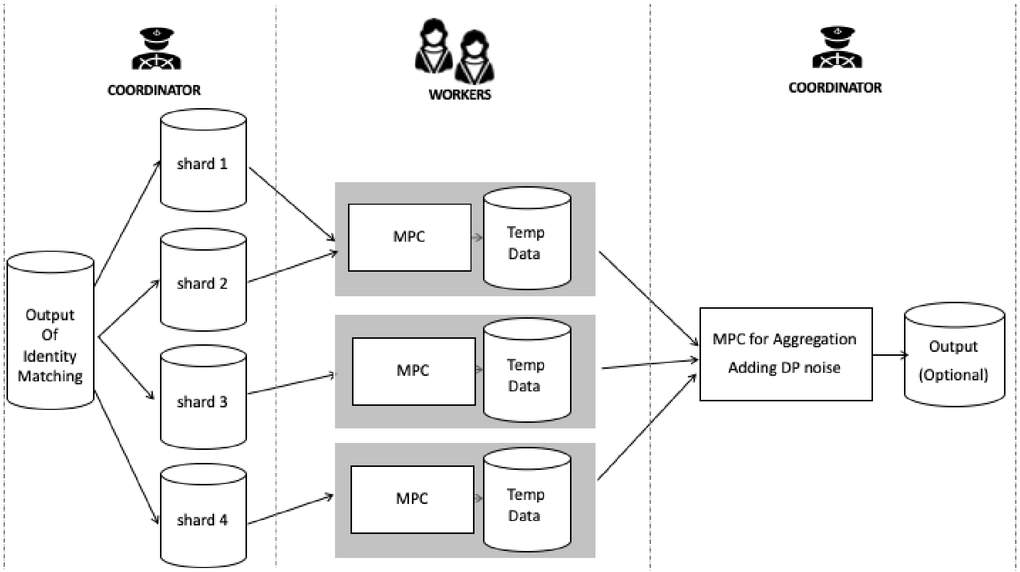

Figure 2: Private RCT design. Garbler and Evaluator have the same overall architecture.

2.4.3 Distributed DP Noise Generation. If any party knows the DP noise added to the final output, they can back out the non-private deterministic output. One way to avoid that is to add party A-generated random noise to the final output and send the sum to party B, and add party B-generated random noise to the output exposed to party A. This way, neither party sees the true output, and both parties have incentives to protect their noise values.

However, a malicious adversary may intentionally choose a huge noise to destroy the other party's data utility. To address this concern, we use the cut and choose method. Each party generates a list of random values as noise, drawn from respective Laplace or Gaussian distributions scaled proportional to sensitivities. They commit to these values and put them as input to the garbled circuit. Each party also chooses a random index which will be used to select one of the other party's noise inputs and includes it as input to the garbled circuit. The computation circuit picks the noise values at the two indices and adds them for DP noise and reveals the rest of random values. This way, both parties can verify that the noises come from the claimed distribution.

One caveat with our system is that the private identity match already leaks the size of the match, which is not differentially private. One method to prevent this problem is adding synthetic data to the set in a way that it hides the size of the match. However, addressing this issue is an open problem for future.

REFERENCES

[1] Jordan Awan and Aleksandra Slavkovic. 2019. Differentially Private Inference for Binomial Data. *Journal of Privacy and Confidentiality* 10, 1 (mar 2019). arXiv:1904.00459

[2] Thomas Brawner and James Honaker. 2018. *Bootstrap Inference and Differential Privacy : Standard Errors for Free.* Technical Report. 1–17 pages.

[3] Prasad Buddhavarapu, Andrew Knox, Payman Mohassel, Shubho Sengupta, Erik Taubeneck, and Vlad Vlaskin. Private Matching for Compute. Cryptology ePrint Archive, Report 2020/599. (2020). https://eprint.iacr.org/2020/599.

[4] Wenxin Du, Andrew Bray, and Adam Groce. 2020. *Differentially Private Confidence Intervals.* Technical Report. arXiv:2001.02285v1

[5] Cynthia Dwork and Aaron Roth. 2013. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2013), 211–487. https://doi.org/10.1561/0400000042

[6] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. 2017. Exposed! A Survey of Attacks on Private Data. *Annual Review of Statistics and Its Application (2017)* (2017).

[7] Russell P Harris, Mark Helfand, Steven H Woolf, Kathleen N Lohr, Cynthia D Mulrow, Steven M Teutsch, David Atkins, Methods Work Group Third US Preventive, and Services Task Force. 2001. Current methods of the US Preventive Services Task Force: a review of the process. *American journal of preventive medicine* 20, 3 (2001), 21–35.

[8] Vishesh Karwa and Salil Vadhan. 2017. *Finite sample differentially private confidence intervals.* Technical Report.

[9] Andrew Mcgregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil Vadhan. 2011. The Limits of Two-Party Differential Privacy. (2011).