Poster: PrivForm — A Privacy-Preserving Framework for Online Forms

Andreas Brüggemann
Technical University of Darmstadt
Germany
brueggemann@encrypto.cs.tu-darmstadt.de

Tamino Goldan
Technical University of Darmstadt
Germany
tamino.goldan@outlook.com

Thomas Schneider
Technical University of Darmstadt
Germany
schneider@encrypto.cs.tu-darmstadt.de

Abstract

Online forms are used in diverse contexts such as surveys, scheduling, or political polls. However, form response data is usually stored in plaintext to enable easy analysis of aggregated answers. Hence, current online form services put potentially sensitive user data at risk while giving users no control over how their data is processed. We introduce PrivForm, a privacy-preserving framework for online forms, that utilizes multiple existing non-privacy-preserving form services by secret sharing form inputs between them, hiding these inputs from individual service providers while seamlessly integrating with existing platforms. Then, Secure Multi-Party Computation (MPC) is used for privacy-preserving analysis of the collected data while providing transparency on how the data is being processed.

1 Introduction

Online forms are a popular web service that often is invaluable, e.g., between friends and family to schedule a meetup, for researchers to conduct research studies, for companies for market research purposes or customer feedback, and for many more applications. A variety of companies such as Google, Microsoft, or Jotform offer tools where anyone can create and submit forms, and respondents can submit their answers, often by simply clicking on a link to access the poll. While this offers excellent functionality to the users, it insufficiently protects their privacy as the respective form service provider usually has plaintext access to all given responses. The provider might decide to analyze given answers while infringing on the respondents' privacy for its own benefit. Moreover, there is the risk of a data breach that involuntarily leaks responses to third parties.

Such risks can be mitigated by only storing form responses in encrypted form [11]. For instance, Jotform [7] and SurveyCTO [10] deploy asymmetric encryption s.t. only the creator of the form can decrypt all responses and analyze them. This approach does not fully protect privacy because still, the form creator has access to all responses whereas in an ideal world, they should have access only to the aggregated statistics over this data, but nothing more. In addition, such an approach limits the practicability as it requires key management on the form creator's side, custom infrastructure for storing answers, and as it demands all respondents to use a specific service which might reduce the form's outreach due to missing acceptance of some users regarding the used service.

To tackle these issues, we introduce PrivForm, a new framework for online forms which provides more privacy than prior services while maintaining good usability similar to that of non-privacy-preserving online form services. Instead of using asymmetric encryption to keep responses private, we secret share [6, 9] them among different existing online form services, keeping them private as long as there is no malicious collusion between all service providers. This also enables easy use of Secure Multi-Party Computation (MPC), taking as input the secret shared responses and revealing only some agreed on statistics/data-aggregation and nothing more. While there are works that utilize secret sharing for questionnaires or aggregation usable for such, e.g., [1, 2, 5, 12], they do not integrate with existing online form services.

Instead, we generalize format-preserving encryption [3], enabling PrivForm to generate shares of form responses that themselves are valid responses and then store them in multiple existing non-privacy-preserving online form services. Hence, we build on top of existing storage infrastructure to ease the deployment of our privacy-preserving service. PrivForm includes a new application PrivFormApp providing users with a similar user experience to that of using the underlying online form services directly. To increase the outreach of forms, PrivForm also leaves the choice whether to use secret sharing or submitting responses in a non-private way to the user which also allows users to ordinarily use a single of the underlying form services directly, e.g., using Google Forms. Hence, our approach gives users complete freedom to decide to submit forms using the services they are used to or secret sharing their responses between multiple such services using our new app PrivFormApp.

2 Secret Sharing Form Data

While simply secret sharing [6, 9] user inputs is straightforward and has been done, e.g., for sharing email contents for privacy-preserving email services [4], utilizing existing online form infrastructure imposes an additional challenge. This is due to the specific input ranges of form inputs and potential form validation which additionally may prevent specific inputs. This does not only need to be taken into consideration for the plain inputs, but also for the individual shares that are to be stored in forms with the same format requirements again.

As an example, we consider a multiple choice question with three possible options A, B, and C. An additional requirement may be that at least one option has to be selected. Simply secret sharing one bit per option does not work as the shares sent to one fixed server could all be zero while not selecting any option locally violates the requirement of at least one option being selected. Disabling this requirement, on the other hand, would also allow people who directly access the poll on this server in a non-privacy-preserving fashion to submit invalid choices.

Instead, we use a single arithmetic and additive secret sharing to encode the entire answer as an element of the ring $\mathbb{Z}_m$, where m is the size of the set of legal answers $\mathcal{A}$. Therefore, we fix some bijection $\mathcal{F}_{\text{MAP}} : \mathcal{A} \to \mathbb{Z}_m$ and encode an answer $a \in \mathcal{A}$ as $\mathcal{F}_{\text{MAP}}(a) \in \mathbb{Z}_m$ locally. The result x is then additively shared

over $\mathbb{Z}_m$ as $x = x_1 + \cdots + x_n \in \mathbb{Z}_m$ for n parties and finally, each share x_i is converted back to $\mathcal{A}$ using $\mathcal{F}_{\text{MAP}}^{-1}$ before storing it to the respective online form service. An example for this process is given in Figure 1. Note that the internally used additive sharing over $\mathbb{Z}_m$ follows standard techniques [6] and that the overall process including the use of $\mathcal{F}_{\text{MAP}}$ is lightweight on the clients and servers. Furthermore reversing the sharing process for reconstructing an input will always result in a valid answer. Hence, all possible sharings represent valid answers and additional validation to protect against wrongly formatted data is not required.

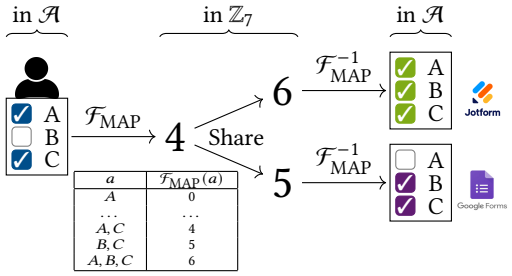
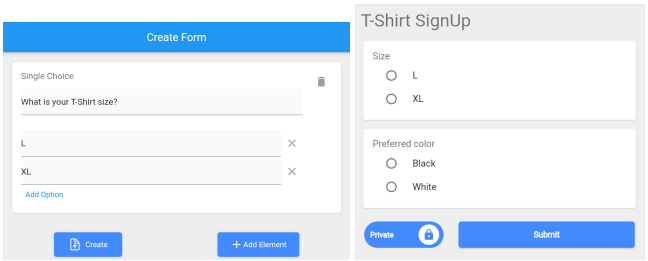


Figure 1: Sharing of an answer for a multiple choice question with three choices and no empty input allowed (set of legal answers $\mathcal{A}$). Answers are mapped to $\mathbb{Z}_7$, additively shared and then mapped back to valid answers in $\mathcal{A}$.

We use similar approaches for other possible question formats such as single choice or number inputs. E.g., for number inputs $\mathcal{F}_{\text{MAP}}$ can be defined not only according to some valid input range, but also to, e.g., exclude certain values from an input range and more. The only exception from arithmetic sharing are text inputs which we share using Boolean secret sharing as in [4].

3 Application

We implement PrivFormApp, a cross-platform application that runs on mobile devices and the web. With PrivFormApp, users can centrally create custom forms so that respective copies are automatically set up at the connected online form services (see Figure 2a). Respondents can then be invited by sharing an invitation code. During the submission of a form, the app secret shares the responses between the form services as discussed in Section 2. Alternatively, users may choose non-private submission and submit their plaintext response directly and only to the service of their choice, using either PrivFormApp or the native interface provided by the chosen service. An example of the form submission menu of PrivFormApp is provided in Figure 2b.



(a) Form creation **(b) Form submission**

Figure 2: Screenshots of PrivFormApp

Overall, PrivFormApp serves as a convenient front-end for form submission as well as administration which in contrast to similar services provided by many existing providers also protects the users' answers. The additional layer of protection yielded by the underlying secret sharing is transparent, i.e., it introduces no novel obstacles for users besides a single button that allows them to choose whether they wish to use secret sharing or not.

4 Privacy-Preserving Response Data Analysis

To run data analysis on the collected form responses, we do not download and reconstruct the secret shared answers, but privately run all analysis in MPC, only revealing the final aggregated outputs. Possible aggregation functions include computing the sum or mean of numeric values (e.g., the average salary of study participants), the frequency of specific answers (e.g., to obtain a histogram of participants' ages), the top-k most frequent responses or the heavy hitters (e.g., the most popular time slots for a meeting to be scheduled), and further similar or also composite functions. Such functions can then be securely evaluated using MPC between the servers storing the secret shared answers or, alternatively, another set of non-colluding servers in an outsourcing scenario.

As a proof of concept, we implemented frequency analysis and top-k analysis as exemplary aggregation functions in the MPC framework MP-SPDZ [8] using standard techniques alongside custom functions to deal with our custom secret sharing from Section 2. Our implementations are oblivious to the underlying MPC protocol so that they are easy to adapt to different threat models. Here, we consider the semi-honest model where the servers carrying out the computation are assumed to not deviate from the protocol, but only try to extract knowledge from what they see.

In a LAN setting with two parties, we measure a throughput of processing 730 answers/s considering form elements with 4 answer options. Using three non-colluding parties increases the performance to nearly 29k answers/s. Switching to a WAN setting, the throughput is 24× - 52× lower.

Acknowledgments

This project received funding from the ERC under the European Union's Horizon 2020 research and innovation program (grant agreement No. 850990 PSOTI). It was co-funded by the DFG within SFB 1119 CROSSING/236615297 as well as GRK 2050 Privacy & Trust/251805230.

References

[1] Surya Addanki, Kevin Garbe, Eli Jaffe, Rafail Ostrovsky, and Antigoni Polychroniadou. 2022. Prio+: Privacy Preserving Aggregate Statistics via Boolean Shares. In *SCN*.
[2] Richard Barnes, David Cook, Christopher Patton, and Phillipp Schoppmann. 2023. Verifiable Distributed Aggregation Functions. https://datatracker.ietf.org/doc/draft-irtf-cfrg-vdaf/04/ (visited on 2024/07/29).
[3] John Black and Phillip Rogaway. 2002. Ciphers with Arbitrary Finite Domains. In *CT-RSA*.
[4] Gowri R Chandran, Raine Nieminen, Thomas Schneider, and Ajith Suresh. 2023. PrivMail: A Privacy-Preserving Framework for Secure Emails. In *ESORICS*.
[5] Henry Corrigan-Gibbs and Dan Boneh. 2017. Prio: Private, Robust, and Scalable Computation of Aggregate Statistics. In *NSDI*.
[6] Oded Goldreich, Silvio Micali, and Avi Wigderson. 1987. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In *STOC*.
[7] Jotform. 2024. How to Enable Form Encryption. https://www.jotform.com/help/344-encrypted-forms-and-how-to-use-them/. (visited on 2024/07/29).
[8] Marcel Keller. 2020. MP-SPDZ: A Versatile Framework for Multi-Party Computation. In *CCS*.

[9] Adi Shamir. 1979. How to Share a Secret. *Comm. ACM* 22, 11 (1979).
[10] SurveyCTO. [n. d.]. Encrypting form data (end-to-end encryption). https://docs.surveycto.com/02-designing-forms/02-additional-topics/06.encrypting.html. (visited on 2024/07/29).
[11] Aytekin Tank. 2015. Introducing Encrypted Forms: The Ultimate in Online Form Security. https://www.jotform.com/blog/introducing-encrypted-forms/. (visited on 2024/07/29).
[12] Kassaye Yitbarek Yigzaw, Antonis Michalas, and Johan Gustav Bellika. 2016. Secure and Scalable Statistical Computation of Questionnaire Data in R. *IEEE Access* (2016).