A Decentralized Information Marketplace Preserving Input and Output Privacy

Steven Golob
Sikha Pentyala
University of Washington Tacoma
USA
{golobs,sikha}@uw.edu

Rafael Dowsley
Monash University
Australia
rafael.dowsley@monash.edu

Bernardo David
IT University of Copenhagen
Denmark
bernardo@bmdavid.com

Mario Larangeira
Tokyo Institute of Technology/IOG
Japan/Singapore
mario{@c.titech.ac.jp,.larangeira@iohk.io}

Martine De Cock*
Anderson Nascimento
University of Washington Tacoma
USA
{mdecock,andclay}@uw.edu

ABSTRACT

Data-driven applications are engines of economic growth and essential for progress in many domains. The data involved is often of a personal nature. We propose a decentralized information marketplace where data held by *data providers*, such as individual users can be made available for computation to *data consumers*, such as government agencies, research institutes, or companies who want to derive actionable insights or train machine learning models with the data while (1) protecting input privacy, (2) protecting output privacy, and (3) compensating data providers for making their sensitive information available for secure computation. We enable this privacy-preserving data exchange through a novel and carefully designed combination of a blockchain that supports smart contracts and two privacy-enhancing technologies: (1) secure multi-party computations, and (2) robust differential privacy guarantees.

CCS CONCEPTS

• **Security and privacy → Economics of security and privacy**; **Privacy protections**; **Management and querying of encrypted data**; **Privacy-preserving protocols**; • **Theory of computation → Cryptographic protocols**.

KEYWORDS

Data holder, data consumer, data economy, privacy budget, Differential Privacy, Secure Multiparty Computation, blockchain.

*Guest Professor at Ghent University

DEC '23, June 18, 2023, Seattle, WA, USA

ACM ISBN 979-8-4007-0846-6/23/06.
https://doi.org/10.1145/3600046.3600047

ACM Reference Format:
Steven Golob, Sikha Pentyala, Rafael Dowsley, Bernardo David, Mario Larangeira, Martine De Cock, and Anderson Nascimento. 2023. A Decentralized Information Marketplace Preserving Input and Output Privacy. In *Second ACM Data Economy Workshop (DEC '23), June 18, 2023, Seattle, WA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3600046.3600047

1 INTRODUCTION

We live in a data science ecosystem where the large-scale collection and processing of data have become commonplace. Personal information is routinely collected by companies and government agencies without giving individuals much control over how their data is used, or the benefits they receive in return for their data. The tension between the desire to promote an economy based on free-flowing data on one hand, and the need to protect privacy on the other hand — as reflected in new regulations such as the GDPR[1], the CCPA[2], and the AI Bill of Rights[3] — can be eased by Privacy-Enhancing Technologies (PETs), as we do in this work. We propose a decentralized information marketplace where data held by *data providers* — such as individual users — can be made available for computation to *data consumers* — such as government agencies, research institutes, or companies who want to derive actionable insights or train machine learning models with the data — while (1) protecting *input privacy*, (2) protecting *output privacy*, and (3) *compensating* data providers for providing their sensitive information as input for secure computations.

Input privacy means keeping the input data delivered by the data providers hidden from the data consumers, or from any parties performing computations on behalf of the data consumers. It is commonly achieved through the use of cryptographic techniques such as Secure Multiparty Computation (MPC) [17] that enable computations over data while it stays encrypted. The obtained outputs are in principle the same as one would obtain with computations over a plaintext version of the data. While often hailed as a strength, the latter is problematic when the outputs – such as trained machine learning (ML) models – are disclosed to the data

[1]European General Data Protection Regulation https://gdpr-info.eu/
[2]California Consumer Privacy Act https://oag.ca.gov/privacy/ccpa
[3]https://www.whitehouse.gov/ostp/ai-bill-of-rights/

consumer, as outputs leak information about the underlying input data that can be recovered via adversarial attacks [15, 21, 37, 40].

Output privacy refers to obfuscating the output so that the ability of such attacks is greatly hindered. It is commonly achieved through Differential Privacy (DP) mechanisms [20] that perturb the outputs by adding noise, the amount of which is controlled by a privacy budget ϵ. In general, a smaller value of ϵ means that more noise is added, and privacy is better protected.

Orthogonal to the above, *compensating data providers* can be done via blockchain-based smart contracts. The concept of smart contract was first introduced by Szabo in 1997 [38] and implementations of smart contract are available in many current blockchain platforms, *e.g.* [14, 43].

While there is existing work on data marketplaces in which data providers get compensated for their data, including approaches that aim to protect input privacy [6, 26, 33, 36], there is a substantial gap in the literature on solutions that provide output privacy as well. To the best of our knowledge, the only proposal that provides both input privacy and output privacy in an information marketplace is centralized, relying on a single trusted execution environment (TEE), which is essentially a trusted third party [24]. A major technical challenge to develop a DP backed data marketplace (for output privacy), combined with decentralized techniques such as MPC and blockchain protocols (for input privacy and financial compensation), is that DP assumes the existence of a trusted curator who perturbs the outputs by adding noise proportional to ϵ. We propose the first framework that formally and rigorously enables:

(1) **Automatic financial compensation for data providers.** Participants who provide data are automatically compensated via cryptocurrency tokens when their data is used to answer queries by data consumers, where the nature of the queries can range from simple counting queries to advanced queries like training ML models.
(2) **Input privacy.** The data submitted by the data providers is handled via a privacy-preserving smart contract (PPSC) constructed by combining MPC and standard (non-privacy-preserving) smart contracts in order to orchestrate an MPC computation of queries to offer meaningful results to data consumers.
(3) **Output privacy.** The application of DP, in particular *a privacy budget control*, assures the users control of their information and prevents misuse by the data consumers.
(4) **Decentralization.** Our framework is fully decentralized via the use of PPSCs based on MPC and standard smart contract platforms. To provide output privacy in a decentralized setting, we replace the central aggregator from the global DP paradigm by MPC protocols for noise generation. In other words, our proposed setting is not based on any trusted third party.

2 RELATED WORK

We characterize solutions for building privacy-preserving data marketplaces by whether or not they address compensation, input or output privacy, and operate in a decentralized manner. By viewing related works through this lens, our contribution is pronounced, as it is the only work that exhibits all four attributes (Tab. 1).

A whole line of research has emerged to explore how combining MPC techniques with cryptocurrencies and smart contracts can facilitate private data trading. The blockchain infrastructure is used to orchestrate MPC protocols that carry-out query-answering computations to data consumers while managing the agreed-upon compensations. The MPC computations are used to maintain input privacy for the data providers. Similarly, fairness is also an outcome in a body of work that uses 'proofs of cheating' [7] to detect maliciously acting participants and penalizing them financially, thereby protecting honest data providers. Upon verifying honest behavior, the smart contracts distribute financial rewards according to the results of the computation/application [1, 3, 4, 7–12, 18, 19, 29–31]. The vast majority of previous works on privacy-preserving information marketplaces only address input privacy. While operating completely decentralized, they do not provide output privacy, thereby leaving data providers vulnerable to reconstruction attacks once the data is shared with the consumer.

To the best of our knowledge, there is only one information marketplace demonstrated in the literature that guarantees output privacy [24]. It utilizes DP, and tracks the privacy budget as data consumers make queries so that providers' data leakage is limited. Queries using higher amounts of 'privacy budget' ϵ can be made to cost more, as they provide greater utility (less noise) to the data consumer, but leak more privacy. The approach by Hynes et al. [24] is centralized as it assumes the use of a TEE, which is essentially a trusted third party. Moreover, TEEs are prone to known vulnerabilities [25] that may undermine both input and output privacy by allowing the adversary to (partially) learn TEEs' internal states. To the best of our knowledge, an approach that provides compensation, input privacy, and output privacy in a decentralized manner, as we propose in this paper, has not yet been described in the open literature.

A plausible explanation for the gap in literature on decentralized marketplaces that provide DP is that the DP paradigm is inherently centralized, in the sense that it assumes the existence of a central curator who receives all the data, performs computations over it, and adds noise to the outputs before disclosing them. Solutions for providing DP in a decentralized manner, including approaches based on replacing the central curator by MPC protocols that are run by distributed servers, have recently been proposed by us and others for training convolutional neural networks (CNNs) [23, 41, 44], decision trees [41], linear support vector machines [41], logistic regression models [34], and even for generating synthetic data [35]. While providing both input and output privacy, these methods were proposed outside of the context of data marketplaces. Unlike the approach that we propose in this paper, they do not include a compensation mechanism, and instead tacitly assume that data providers are willing to donate their data for free and without any control over the privacy budget.

3 THE FRAMEWORK

We divide this section into three parts. We start describing our framework by (1) detailing the main entities involved in the marketplace and how they interact. Then, (2) we introduce the adversary model and the security definitions relevant to our proposed system. Finally, (3) we outline each of the phases of our decentralized information market protocol.

Table 1: Work on privacy-preserving data marketplaces

	Compensation	Input Privacy	Output Privacy	Decentralized
Hynes et al., 2018 [24]	✓	✓	✓	
Koch et al., 2022 [27]	✓	✓		✓
Abadi et al., 2023 [1]	✓	✓		✓
Koutsos et al., 2021 [28]	✓	✓		✓
Baum et al., 2020 [7]	✓	✓		✓
Baum et al., 2021 [10]	✓	✓		✓
Baum et al., 2022 [11]	✓	✓		✓
More et al., 2022 [32]	✓	✓		✓
Giaretta et al., 2021 [22]	✓	✓		✓
Weng et al., 2021 [42]	✓			✓
Aljohani et al., 2023 [2]	✓			✓
Chen et al., 2021 [16]	✓			✓
Tian et al., 2022 [39]	✓	✓		✓
Andrychowicz et al., 2014 [3]	✓	✓		✓
Our Solution	✔	✔	✔	✔

3.1 Marketplace Actors and Building blocks

There are two types of actors in our marketplace:

- Data Consumers C_j: These are organizations that are interested in computing queries (*e.g.* getting statistics or training ML models) on private information from data providers. Data consumers are willing to provide financial compensation for the input data used in their queries. Real world examples of such organizations are the U.S. Census Bureau, content providers who want to personalize their services to their customers, research institutions, etc.;
- Data Providers $\mathcal{D}_i$: These are entities willing to provide personal data for privacy-preserving analysis in exchange for a financial reward. Data providers can be individuals, data brokers, retailers, or any other organization interested in providing data for financial benefit.

We employ smart contracts [38] to automate our marketplace and enforce its privacy guarantees. In a nutshell, smart contracts are programs deployed on top of a cryptocurrency platform based on a distributed public ledger (*e.g.* a blockchain). These programs, each of them with its respective state, are executed by the nodes who maintain the cryptocurrency platform, automatically altering their internal states and outputting financial transactions when activated by input transactions (containing both financial tokens and data). Regular smart contracts (*e.g.* [14, 43]) are unable to handle confidential inputs since all input transactions are publicized in the public ledger. However, our marketplace crucially relies on the abilities (1) to process queries on input data privately (achieving input privacy) and (2) to ensure that query outputs do not leak input data (achieving output privacy). Hence, we build on privacy-preserving smart contracts (PPSC) made possible by recent developments [7, 10, 11] in the intersection of distributed ledger technology and privacy preserving computation:

- Privacy Preserving Smart contracts (PPSC): Our central building block are PPSCs [7, 10, 11] deployed on a public ledger, which automatically process input transactions and issue output transactions while maintaining the smart contract internal state private. We assume an arbitrary data consumer C_j can instantiate a PPSC $\mathcal{R}_j$, depositing a certain amount of cryptocurrency tokens to be distributed as rewards and establishing the parameters of the queries it wishes to compute. The PPSC will orchestrate the marketplace operations for obtaining data from data providers, performing privacy-preserving computations on such data and automatically compensating data providers with cryptocurrency tokens from the reward pool provided by data consumers.

In the description of our protocol, *i.e.*, Section 3.3, we assume that a PPSC platform is given as an ideal resource, *i.e.* as an incorruptible black-box building block. Later on, we discuss how such a PPSC can be constructed departing from a standard non-privacy-preserving smart contract platform and standard cryptographic techniques via [7, 10, 11].

3.2 Security Definitions and Adversarial Model

A randomized algorithm $\mathcal{M}$ is called ϵ-DP if for each pair (D_1, D_2) of adjacent datasets (*i.e.*, datasets that differ in only one entry), and for each subset O of the range of $\mathcal{M}$, $Pr[\mathcal{M}(D_1) \in O] \leq e^{\epsilon} Pr[\mathcal{M}(D_2) \in O]$. The parameter $\epsilon \geq 0$ denotes the *privacy budget*, with smaller values indicating stronger privacy guarantees. DP ensures that the inclusion or exclusion of any entry in the dataset is obscured, in the sense that any output obtained from computations over the dataset would have been similarly likely to be reached whether the entry was present in the dataset or not. This definition can be relaxed by adding a constant δ, resulting in what we call approximate differential privacy: $Pr[\mathcal{M}(D_1) \in O] \leq e^{\epsilon} Pr[\mathcal{M}(D_2) \in O] + \delta$. Our proposed marketplace and protocols can be applied to this approximate differential privacy scenario too. However, for the sake of simplicity, and due to space restrictions, we limit our discussion to the case where $\delta = 0$.

We consider a static active probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, *i.e.* a computationally bounded adversary who may corrupt any subset of data providers $\mathcal{D}$ or data consumers $\mathcal{C}$ *before* the protocol execution starts, and deviates from the protocol arbitrarily. We want to ensure the following security and privacy properties for $\mathcal{D}$ and $\mathcal{C}$ against such an adversary $\mathcal{A}$:

- Input Correctness: Given a publicly available specification of the format and properties expected from the data submitted by data providers, it is infeasible for an $\mathcal{A}$ that corrupts a data provider $\mathcal{D}_i \in \mathcal{D}$ to submit non-compliant data that does not adhere to this specification;
- Output Correctness: Given a publicly available specification of a query requested by a data consumer C_j and data provided by a subset $\mathcal{D}' \subset \mathcal{D}$, it is infeasible for an $\mathcal{A}$ corrupting any subset of $\mathcal{D}$ and $\mathcal{C} \setminus C_j$ to convince C_j that an arbitrary output different from that of computing the specified query on input data from $\mathcal{D}'$ is valid;
- Input Privacy: It is infeasible for any $\mathcal{A}$ corrupting a subset $\mathcal{C}' \subset \mathcal{C}$ and a subset $\mathcal{D}' \subset \mathcal{D}$ to obtain any information but the query outputs obtained by corrupted $\mathcal{C}'$ and the inputs provided by the corrupted subset $\mathcal{D}'$;
- Output Privacy: It is infeasible for any $\mathcal{A}$ corrupting a subset $\mathcal{C}' \subset \mathcal{C}$ and a subset $\mathcal{D}' \subset \mathcal{D}$ to violate the DP guarantees

of honest $\mathcal{D} \setminus \mathcal{D}'$ with respect to the outputs obtained by the corrupted subset C';
- Payment Fairness: It is infeasible for any $\mathcal{A}$ corrupting the subsets $C' \subset C$ and $\mathcal{D}' \subset \mathcal{D}$ to prevent honest $\mathcal{D} \setminus \mathcal{D}'$ from receiving payment when their input data is used to compute a query output obtained by any member of C.

3.3 Protocol Outline

For the sake of simplicity, we state a protocol for the case when a data consumer C_j is interested in the result of a query over data provided by a set of data providers $\mathcal{D}$. Our framework generalizes to more complicated settings, such as training of ML models. We start by providing the intuition of the inner workings and usage of the protocol, before concretely outlining its phases.

Intuition. Our protocol works by having a data consumer $C_j \in C$ create a PPSC $\mathcal{R}_j$, specifying the query it wants to perform on data from providers, and depositing cryptocurrency tokens into it. This PPSC specifies the type of the query, the DP level of privacy that will be guaranteed for this query, how many users will have their data used for this DP query, and the financial rewards paid to each data provider if the threshold of the number of participants is reached and the DP query is performed. Needless to say that the data consumer may advertise such requests to potential data providers using other external channels in order to increase awareness about the PPSC that was created.

Data providers $\mathcal{D}_i \in \mathcal{D}$ who wish to contribute their data interact with the PPSC $\mathcal{R}_j$ in order to privately submit their respective data with the guarantee that they will be rewarded if the query computation executes. Additionally, in order to cope with $\mathcal{D}$ providing non-compliant data to the protocol, we note that $\mathcal{R}_j$ verifies the compliance of the data using zero knowledge proofs of data validity submitted by each contributing $\mathcal{D}_i$. $\mathcal{R}_j$ enforces the policies specified by C_j, and automatically pays (via cryptocurrency) the data providers according to the pre-specified rules once the DP query is executed. Notice that the PPSC keeps all of its input queries and its internal state private, achieving input privacy. The set of rules can also specify a minimal amount of data samples to be collected in order to start performing the query from C_j in order to ensure a minimal data utility threshold. The query result is disclosed privately to C_j (*e.g.* in encrypted form).

For the sake of simplicity, we focus on protocols designed to handle a single query. Extending our proposal to accommodate multiple queries can be achieved by incorporating additional settings in the privacy-preserving smart contract. These settings would specify the total number of queries, and an extra verification step would be added to ensure the availability of the privacy budget before revealing the results of subsequent queries.

Phases of the Protocol. In this protocol, a data consumer $C_j \in C$ who wishes to compute a query interacts with data providers $\mathcal{D}_i \in \mathcal{D}$ who hold private data by means of a PPSC. The interaction takes place in the following phases:

(1) Phase 1 - Query Request. In this phase C_j publishes a request for data.
   (a) C_j creates a PPSC $\mathcal{R}_j$. $\mathcal{R}_j$ specifies: the type and format of the data used to compute the query; the minimum number of data samples necessary for computing the query; the privacy budget ϵ specified for the query, the payment for each data sample provided by data providers.
   (b) C_j publicises the smart contract and waits for $\mathcal{D}_i \in \mathcal{D}$.
(2) Phase 2 - Data Collection. If not enough data is available, this phase waits for $\mathcal{D}_i \in \mathcal{D}$ to submit their data. Otherwise, if a query is required and data is already available, proceed directly to Phase 3 - Query Output.
   (a) $\mathcal{D}_i \in \mathcal{D}$ privately send their data to $\mathcal{R}_j$ along with a zero knowledge proof[4] of data validity (to avoid $\mathcal{D}_i$ providing non-compliant data). $\mathcal{R}_j$ verifies in zero knowledge the correctness of the data and accepts or rejects it.
   (b) If $\mathcal{R}_j$ accepts $\mathcal{D}_i$'s transaction, it registers $\mathcal{D}_i$'s input data.
(3) Phase 3 - Query Output. In this phase, $\mathcal{R}_j$ performs the query, and provides (1) the output to C_j and (2) the payment to $\mathcal{D}_i \in \mathcal{D}$ who contributed data. Thus, $\mathcal{R}_j$ proceeds as follows:
   (a) Check that the minimum threshold of data points (specified by C_j) is reached. If either requirement is not satisfied, ignore next steps;
   (b) Perform the query, adding noise to the output using an appropriate DP mechanism publicly specified by C_j via the PPSC conditions, and release the result of the computation to C_j;
   (c) Pay reward to each $\mathcal{D}_i$ who provided data for the query;

4 SECURITY AND PRIVACY ANALYSIS

In this section, we provide a sketch of the security and privacy analysis of our solution with respect to the security guarantees and adversarial model we specified in Section 3.2. In the full version of our paper we present rigorous security definitions and proofs in the Universal Composability framework [13]. For this short version, assuming that we have access to an ideal PPSC (which we explain how to realize in the next section), we argue that the security guarantees are preserved as follows:

- Input Correctness: $\mathcal{D}_i$ must provide a zero knowledge proof of input data validity attesting that its submitted data fulfills the format and properties specified by C_j in public conditions of $\mathcal{R}_j$. Hence, due to the soundness of such proof, it is infeasible for an $\mathcal{A}$ corrupting $\mathcal{D}_i$ to pass the checks performed by $\mathcal{R}_j$ in step (a) of Phase 2;
- Output Correctness: This follows from the fact that $\mathcal{R}_j$ guarantees that the query specified by C_j in step (a) of Phase 1 is correctly computed on the data provided by $\mathcal{D}' \subset \mathcal{D}$ in step (b) of Phase 3;
- Input Privacy: This follows from the fact that $\mathcal{R}_j$ guarantees that all an $\mathcal{A}$ corrupting a subset $C' \subset C$ and a subset $\mathcal{D}' \subset \mathcal{D}$ can learn are the query outputs delivered to C' in step (b) of Phase 3 and the inputs owned by $\mathcal{D}'$. All other internal state of $\mathcal{R}_j$ is kept private;
- Output Privacy: This follows from input privacy and the fact that $\mathcal{R}_j$ correctly executes the DP mechanism in step (b) of Phase 3 before disclosing the query output to C_j;

[4]The PPSC $\mathcal{R}_j$ guarantees privacy for its inputs and internal state, so it could directly check validity predicates on data provider inputs. However, providing a zero knowledge proof of data validity allows third party entities to verify data validity even before the data is processed by the PPSC, saving on the cost of PPSC execution.

- Payment Fairness: This follows from the fact that $\mathcal{R}_j$ automatically rewards each honest $\mathcal{D} \setminus \mathcal{D}'$ who provided input data to a query in step (c) of Phase 3.

5 INSTANTIATING THE BUILDING BLOCKS

The central component of our proposed framework is a PPSC, which we realize from a number of standard cryptographic building blocks. Moreover, we utilize zero knowledge proofs and DP mechanisms in crucial steps of our protocol. In this section, we discuss potential instantiations for each building block used in our construction.

Privacy Preserving Smart Contract. We instantiate a privacy-preserving smart contract departing from the approach of Insured MPC [7]. In this framework, a standard non-privacy-preserving smart contract is used to orchestrate the execution of an MPC protocol, which computes the PPSC instructions on the private input data, delivering (private) outputs to the parties who contributed financially. Moreover, financial transactions are automatically generated according to the MPC output.

Unlike what we need for our data marketplace, Insured MPC does not distinguish between clients who input data (data providers) or deposit cryptotokens (data consumers) and servers who perform the actual computations. Instead, in Insured MPC, each party who wishes to participate in the PPSC execution registers itself by sending public transactions to the orchestration smart contract, depositing cryptocurrency tokens that will be traded according to the output of the MPC. After all parties are registered, they execute an MPC protocol that performs the actual privacy-preserving computation on their inputs. As per the standard security guarantees of MPC, this approach guarantees that, as long as at least one of the parties executing the MPC protocol is honest, only the output is learned by each party, preserving the privacy of the internal state of the PPSC and the input data.

While it implements the kind of PPSC that we require with Universally Composable security [13], the Insured MPC approach has two important caveats: (1) it requires the parties who provide input to execute the underlying MPC protocol; (2) the underlying MPC protocol with the necessary properties has a high concrete overhead in relation to state-of-the-art MPC protocols. In order to solve these issues, we employ techniques from P2DEX [10], which allows for data providers to very cheaply provide "encrypted" versions of their private inputs to servers who execute the MPC protocol and then provide the outputs to the data consumer. Moreover, using the techniques from P2DEX, our PPSC can be instantiated from any state-of-the-art MPC protocol without extra overheads.

This so-called *outsourced MPC* model is optimal for our scenario, where data providers simply provide their inputs to the PPSC but do not wish to be encumbered by a complex cryptographic protocol execution. Instead the MPC execution is outsourced to servers that can be automatically rewarded for this task (similarly to the parties who execute the underlying standard smart contract and cryptocurrency platform). In this case, our privacy and correctness guarantees are maintained if at least one of the servers executing the underlying outsourced MPC is honest. Notice that now it is the servers that provide the orchestration smart contract with the computed outputs, plus proofs of validity.

In scenarios where our market place must also protect the financial transactions handled by the PPSC, we further augment our PPSC instantiation using techniques from Eagle [11], which has all properties from Insured MPC and the efficiency from P2DEX while allowing for the resulting PPSC to receive and output privacy-preserving financial transactions. While this is not the focus of our work, this property may be desirable in scenarios where the very financial transactions executed by data providers and data consumers may leak information about private data.

MPC Infrastructure We propose to use the SPDZ protocol for the case of malicious adversaries allowed to corrupt a majority of the servers. If we restrict the adversary to corrupt less than 2/3 of the players and be honest-but-curious, we propose to use the protocol by Araki et al. [5].

Protocols for Handling Differential Privacy DP traditionally assumes the existence of a trusted curator that has direct access to the data, computes a query in the clear, samples noise from a pre-specified probability distribution according to a DP mechanism, adds the noise to the query result and makes it public. To remove this central point of failure from our framework, we replace the trusted curator by privacy-preserving smart contracts. The noise generation happens in the underlying MPC protocol, specified in the smart contract, and executed by multiple computing parties. We have already proposed and implemented several of such protocols [34, 35].

6 CONCLUSION

In this paper we have outlined how to create a decentralized data marketplace with input and output privacy, in which data providers are automatically rewarded and an audit trail of all privacy guarantees, including the privacy budget, is kept. As described, this can be achieved by combining existing privacy-preserving smart contracts (PPSCs) based on Secure Multiparty Computation (MPC) and MPC protocols to perform Differential Privacy (DP) queries without reliance on a central aggregator. While one can use existing constructions for each of these technical building blocks, an important next step is the development of more efficient, tailor-made MPC protocols for exact tasks that need to be handled in the data marketplace.

REFERENCES

[1] Aydin Abadi and Steven J Murdoch. 2023. Earn While You Reveal: Private Set Intersection that Rewards Participants. *arXiv preprint arXiv:2301.03889* (2023).
[2] Meshari Aljohani, Ravi Mukkamala, and Stephan Olariu. 2023. A Framework for a Blockchain-Based Decentralized Data Marketplace. In *Wireless Internet: 15th EAI International Conference, WiCON 2022, Virtual Event, November 2022, Proceedings*. Springer, 59–75.
[3] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek. 2014. Fair two-party computations via bitcoin deposits. In *Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers 18*. Springer, 105–121.
[4] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. 2014. Secure Multiparty Computations on Bitcoin. In *2014 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 443–458. https://doi.org/10.1109/SP.2014.35
[5] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara. 2016. High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority. In *ACM CCS 2016*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM Press, 805–817. https://doi.org/10.1145/2976749.2978331

[6] Prabal Banerjee and Sushmita Ruj. 2018. Blockchain Enabled Data Marketplace – Design and Challenges. https://doi.org/10.48550/ARXIV.1811.11462

[7] Carsten Baum, Bernardo David, and Rafael Dowsley. 2020. Insured MPC: Efficient Secure Computation with Financial Penalties. In *FC 2020 (LNCS, Vol. 12059)*, Joseph Bonneau and Nadia Heninger (Eds.). Springer, Heidelberg, 404–420. https://doi.org/10.1007/978-3-030-51280-4_22

[8] Carsten Baum, Bernardo David, Rafael Dowsley, Ravi Kishore, Jesper Buus Nielsen, and Sabine Oechsner. 2023. CRAFT: Composable Randomness Beacons and Output-Independent Abort MPC From Time. In *To appear at PKC 2023. Available at Cryptology ePrint Archive 2020/784.*

[9] Carsten Baum, Bernardo David, Rafael Dowsley, Jesper Buus Nielsen, and Sabine Oechsner. 2021. TARDIS: A Foundation of Time-Lock Puzzles in UC. In *EUROCRYPT 2021, Part III (LNCS, Vol. 12698)*, Anne Canteaut and François-Xavier Standaert (Eds.). Springer, Heidelberg, 429–459. https://doi.org/10.1007/978-3-030-77883-5_15

[10] Carsten Baum, Bernardo David, and Tore Kasper Frederiksen. 2021. P2DEX: Privacy-Preserving Decentralized Cryptocurrency Exchange. In *ACNS 21, Part I (LNCS, Vol. 12726)*, Kazue Sako and Nils Ole Tippenhauer (Eds.). Springer, Heidelberg, 163–194. https://doi.org/10.1007/978-3-030-78372-3_7

[11] Carsten Baum, James Hsin yu Chiang, Bernardo David, and Tore Kasper Frederiksen. 2022. Eagle: Efficient Privacy Preserving Smart Contracts. Cryptology ePrint Archive, Paper 2022/1435. https://eprint.iacr.org/2022/1435 To appear at Financial Cryptography 2023.

[12] Iddo Bentov, Ranjit Kumaresan, and Andrew Miller. 2017. Instantaneous Decentralized Poker. In *ASIACRYPT 2017, Part II (LNCS, Vol. 10625)*, Tsuyoshi Takagi and Thomas Peyrin (Eds.). Springer, Heidelberg, 410–440. https://doi.org/10.1007/978-3-319-70697-9_15

[13] Ran Canetti. 2001. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *42nd FOCS*. IEEE Computer Society Press, 136–145. https://doi.org/10.1109/SFCS.2001.959888

[14] Cardano. 2023. Cardano. https://cardano.org/. [Online; accessed 7-March-2023].

[15] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*. 267–284.

[16] Peng Chen, Peichang Shi, Jie Xu, Xiang Fu, Linhui Li, Tao Zhong, Liangliang Xiang, and Jinzhu Kong. 2021. TeeSwap: Private Data Exchange using Smart Contract and Trusted Execution Environment. In *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, 237–244.

[17] Ronald Cramer, Ivan Damgard, and Jesper Nielsen. 2015. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press Print, New York.

[18] Bernardo David, Rafael Dowsley, and Mario Larangeira. 2018. Kaleidoscope: An Efficient Poker Protocol with Payment Distribution and Penalty Enforcement. In *FC 2018 (LNCS, Vol. 10957)*, Sarah Meiklejohn and Kazue Sako (Eds.). Springer, Heidelberg, 500–519. https://doi.org/10.1007/978-3-662-58387-6_27

[19] Bernardo David, Rafael Dowsley, and Mario Larangeira. 2019. ROYALE: A Framework for Universally Composable Card Games with Financial Rewards and Penalties Enforcement. In *FC 2019 (LNCS, Vol. 11598)*, Ian Goldberg and Tyler Moore (Eds.). Springer, Heidelberg, 282–300. https://doi.org/10.1007/978-3-030-32101-7_18

[20] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*. Springer, 265–284.

[21] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1322–1333.

[22] Lodovico Giaretta, Ioannis Savvidis, Thomas Marchioro, Šarūnas Girdzijauskas, George Pallis, Marios D Dikaiakos, and Evangelos Markatos. 2021. PDS 2: A user-centered decentralized marketplace for privacy preserving data processing. In *2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 92–99.

[23] Xiaolan Gu, Ming Li, and Li Xiong. 2021. Precad: Privacy-preserving and robust federated learning via crypto-aided differential privacy. *arXiv preprint arXiv:2110.11578* (2021).

[24] Nick Hynes, David Dao, David Yan, Raymond Cheng, and Dawn Song. 2018. A demonstration of sterling: A privacy-preserving data marketplace. *Proceedings of the VLDB Endowment* 11, 12 (2018), 2086–2089.

[25] Patrick Jauernig, Ahmad-Reza Sadeghi, and Emmanuel Stapf. 2020. Trusted Execution Environments: Properties, Applications, and Challenges. *IEEE Security & Privacy* 18, 2 (2020), 56–60. https://doi.org/10.1109/MSEC.2019.2947124

[26] Kaleido. 2023. Kaleido. https://www.kaleido.io/resources/data-marketplaces. [Online; accessed 7-March-2023].

[27] Karl Koch, Stephan Krenn, Tilen Marc, Stefan More, and Sebastian Ramacher. 2022. KRAKEN: a privacy-preserving data market for authentic data. In *Proceedings of the 1st International Workshop on Data Economy*. 15–20.

[28] Vlasis Koutsos, Dimitrios Papadopoulos, Dimitris Chatzopoulos, Sasu Tarkoma, and Pan Hui. 2021. Agora: A privacy-aware data marketplace. *IEEE Transactions on Dependable and Secure Computing* 19, 6 (2021), 3728–3740.

[29] Ranjit Kumaresan and Iddo Bentov. 2014. How to Use Bitcoin to Incentivize Correct Computations. In *ACM CCS 2014*, Gail-Joon Ahn, Moti Yung, and Ninghui Li (Eds.). ACM Press, 30–41. https://doi.org/10.1145/2660267.2660380

[30] Ranjit Kumaresan and Iddo Bentov. 2016. Amortizing Secure Computation with Penalties. In *ACM CCS 2016*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM Press, 418–429. https://doi.org/10.1145/2976749.2978424

[31] Ranjit Kumaresan, Vinod Vaikuntanathan, and Prashant Nalini Vasudevan. 2016. Improvements to Secure Computation with Penalties. In *ACM CCS 2016*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM Press, 406–417. https://doi.org/10.1145/2976749.2978421

[32] Stefan More and Lukas Alber. 2022. YOU SHALL NOT COMPUTE on my Data: Access Policies for Privacy-Preserving Data Marketplaces and an Implementation for a Distributed Market using MPC. *arXiv preprint arXiv:2206.07507* (2022).

[33] Nokia. 2023. Blockchain Data Marketplace Can Fuel the Data Economy. https://www.nokia.com/blog/blockchain-data-marketplace-can-fuel-the-data-economy/. [Online; accessed 7-March-2023].

[34] Sikha Pentyala, Davis Railsback, Ricardo Maia, Rafael Dowsley, David Melanson, Anderson Nascimento, and Martine De Cock. 2022. Training Differentially Private Models with Secure Multiparty Computation. Cryptology ePrint Archive, Report 2022/146. https://ia.cr/2022/146.

[35] Mayana Pereira, Sikha Pentyala, Anderson Nascimento, Rafael T de Sousa Jr, and Martine De Cock. 2022. Secure Multiparty Computation for Synthetic Data Generation from Distributed Data. In *SyntheticData4ML workshop at NeurIPS2022.*

[36] Ocean Protocol. 2023. Ocean Protocol. https://oceanprotocol.com/. [Online; accessed 7-March-2023].

[37] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 587–601.

[38] Nick Szabo. 1997. Formalizing and Securing Relationships on Public Networks. *First Monday* 2, 9 (Sep. 1997). https://doi.org/10.5210/fm.v2i9.548

[39] Zhihua Tian, Jian Liu, Jingyu Li, Xinle Cao, Ruoxi Jia, and Kui Ren. 2022. Private Data Valuation and Fair Payment in Data Marketplaces. *arXiv preprint arXiv:2210.08723* (2022).

[40] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction APIs. In *25th USENIX Security Symposium*. 601–618.

[41] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*. 1–11.

[42] Jiasi Weng, Jian Weng, Chengjun Cai, Hongwei Huang, and Cong Wang. 2021. Golden grain: Building a secure and decentralized model marketplace for MLaaS. *IEEE Transactions on Dependable and Secure Computing* 19, 5 (2021), 3149–3167.

[43] Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151 (2014), 1–32.

[44] Sen Yuan, Milan Shen, Ilya Mironov, and Anderson Nascimento. 2021. Label private deep learning training based on secure multiparty computation and differential privacy. In *NeurIPS 2021 Workshop Privacy in Machine Learning*.