User-Friendly Privacy-Preserving Blockchain-based Data Trading

Jiahui Cao[1], Junyao Ye*[2],Junzuo Lai[3]
School of Information Engineering Jingdezhen Ceramic University, Jingdezhen, 333403 Jiangxi Province, China[1,2]
College of Information Science Technology, Jinan University, China[3]

***Abstract*—As the digital economy flourishes, the use of blockchain technology for data trading has seen a surge in popularity. Yet, previous approaches have frequently faltered in harmonizing security with user experience, culminating in suboptimal transactional efficiency. This study introduces a personalized local differential privacy framework, adeptly tackling data security concerns while accommodating the individual privacy preferences of data owners. Furthermore, the framework bolsters transaction flexibility and efficiency by catering to needs of data consumers for detailed queries and enabling data owners to effortlessly elevate their privacy budget to achieve greater financial returns. The efficacy of our approach is validated through a comprehensive series of experimental validations.**

***Keywords*—Data trading; blockchain; personalized local differential privacy; data security; user-friendly**

I. Introduction

The ongoing shift towards informatization in society has resulted in a tremendous increase in data volume. Data trading, evolving as a novel business model, is gaining pivotal importance in today's digital economy. A notable number of users are inclined to offer their personal data in return for access to online services. Nevertheless, as individuals become more aware of the ramifications of companies utilizing their data, understanding the potential consequences and recognizing the intrinsic value of personal data, there is a growing trend towards expecting compensation for the usage of such data [1].

To facilitate this model of data trading, private data trading has emerged as a significant research field, prompting the development of various innovative solutions like FairQuery [2], FairInnerProduct [3], SingleMindedQuery [4], and SmartAuction [5]. These methods utilize Differential Privacy (DP) [6, 7] to safeguard data while providing query results to data consumers (DC), instead of directly handing over the data. Commonly, these solutions engage three key stakeholders: Data Owners (DO), Data Consumers, and a data broker (DB).

DO are individuals who possess data and are interested in commercializing it. This group includes people with diverse types of data, such as social, financial, location, or health-related data. Entities like advertisers, software developers, and retailers represent DC—those in search of external data to support their decision-making processes. They aim to query aggregated information tailored to certain demographics, all within a specified budget. DB collaborates with DO, collects data, and provides query results to DC, thereby benefiting financially from this process.

The depicted transaction model is fundamentally segmented into two principal components: Value Exchange and Information Processing, as delineated in Fig. 1. Within the Value Exchange phase, inputs include DO' data valuation, privacy requirements, and DC' budget. The consequent outputs encompass the remuneration for DO partaking in the transaction, along with the privacy compensation accorded to them. The Information Processing segment entails furnishing DC. with query outcomes, augmented with noise, which typically conform to differential privacy standards to guarantee robust protection of DO' privacy. The architecture of these solutions customarily incorporates several essential attributes to ascertain equity in data trading, such as incentive compatibility, individual rationality, and budget feasibility.

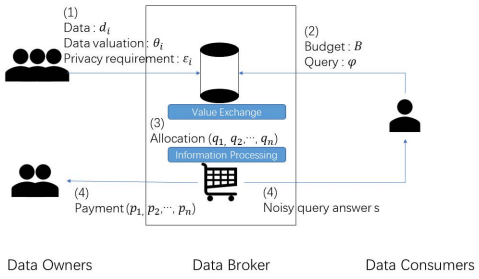

Fig. 1. Data trading.

However, previous models depend on a trusted third party for storing the original data of DO. While centralized storage enhances data integration and processing efficiency, it introduces potential security vulnerabilities. For instance, should the central server be compromised by hackers or if internal staff illicitly access data, the confidentiality of sensitive information cannot be assured. Such uncertainties undermine privacy and integrity of data, impacting the viability and trustworthiness of data transactions. Despite efforts to address these issues through local differential privacy (LDP) [8] and blockchain-based data trading, these approaches have not adequately accounted for the unique privacy preferences of DO and the budgetary limitations of DC, making the process less user-friendly and decreasing transaction efficiency.

To navigate these challenges, personalized local differential privacy (PLDP) [9] emerges as a refined strategy. In this framework, DO are not required to upload their raw data to DB's database. Instead, they apply PLDP measures tailored to their privacy needs and upload the altered data. This method not only safeguards individual privacy but also accommodates the varied privacy demands of different DO, maintaining data

usability and enabling DC to perform statistical analyses. Essentially, this technique eliminates the risk of data exposure since only data that has been processed for PLDP is shared, keeping the original datasets confidential and securely with the DO.

Moreover, to ensure transactions are both fair and adaptable, DC must be empowered to request additional conditions, such as more detailed queries, thus filtering out data not meeting specified privacy standards. This provision fosters a balanced data trading ecosystem and encourages DO to supply data of higher quality and relevance.

As DO engage in multiple transactions and see tangible rewards, their confidence in the data trading system grows. Eventually, they might be inclined to increase their privacy budgets for better compensation. However, frequent data re-uploads can significantly hamper transactional efficiency. Therefore, we introduce a solution enabling DO to effortlessly raise their privacy budgets with the assistance of DB under suitable conditions. This arrangement not only streamlines transactions but also guides DO in aligning their data more accurately with its real-world value, thus reinforcing the reliability and steadiness of data exchanges.

In summary, we have developed a data security and user-friendly data trading model that innovatively employs PLDP technology. This method ensures that the original data of DO does not need to be uploaded, fundamentally preventing privacy risks associated with data breaches. Additionally, we provide DO with a convenient method to increase their privacy budgets. However, due to the current limitations of PLDP technology, this method is currently only applicable to numerical data, which represents a limitation of this study.

Overall, the principal contributions are summarized as follows:

- The deployment of PLDP technology markedly bolstered data security and minimized leakage risks while adeptly catering to DO individual privacy preferences.
- The refinement of query mechanisms to accommodate DC requirements for granular inquiries, thereby elevating data precision and pertinence, which in turn enhances the utility of data and improves the consumer experience.
- The formulation of a scheme enabling data owners to augment their privacy budgets in pursuit of greater compensation, thereby fostering the dissemination of superior data.
- we have made our entire source code and the detailed experimental procedures available on GitHub [10] (https://github.com/cjh20000613/User-Friendly-Privacy-Preserving-Blockchain-Based-Data-Trading).

II. Related Work

A. Private Data Trading

Our research includes a comprehensive review of privacy-preserving data queries. The seminal work by Ghosh and Roth [2] established fundamental frameworks in this domain, notably the Value Exchange and Information Processing, and introduced the FairQuery (FQ) concept. FQ utilizes a greedy algorithm for reverse auctions, aiming to maximize the selection of DO in value exchanges. Additionally, it employs the Laplace mechanism [11] for information processing, facilitating count queries on binary data (0/1 values).

Danderkar et al. [3] extended this research to more general query types, specifically linear predictors, and developed FairInnerProduct (FIP). FIP employs a knapsack problem-solving mechanism for value exchange and provides extra compensation to DO with the highest data value. This model effectively deters DO from underreporting their data value to gain compensation.

Nget et al. [12] proposed two distinct compensation mechanisms: a logarithmic function for conservative approaches (low risk, low return) and a sub-linear function for liberal approaches (high risk, high return). Their goal was to engage DO with varied privacy expectations. Additionally, they tackled the issue of DC arbitrage by employing sampling before querying and imposing restrictions on DC to prevent repetitive queries.

Mengxiao Zhang et al. [4] introduced a pivotal assumption that DO are single-minded, agreeing to sell data only if their privacy demands are met. Building on this, they developed the SingleMindedQuery (SMQ), which incorporates the Bayesian static game approach in its value exchange mechanism and an enhanced exponential mechanism [13] for information processing, thus achieving genuine personalized differential privacy protection. Further, [5]they adapted this mechanism to the blockchain, integrating RSA encryption and signature technology to secure data transmission processes.

Wang et al. [14] and Fallah et al. [15] presented the PDQS, enabling data owners to locally distort their private data to guarantee LDP. Nonetheless, they overlooked the budget limitations of DC. Li et al. [16] proposed a perturbation mechanism that permits DO to submit either accurate values or randomized values with a specific probability. This approach, by weaving together the facets of value exchange and information processing, seeks to refine the precision of query results.

B. Blockchain-based Data Trading

Blockchain-based data trading presents solutions to several issues inherent in traditional centralized data platforms, adeptly addressing concerns such as privacy violations, elevated transaction costs, and limited interoperability. Thanks to blockchain's distributed architecture, data trading activities are decentralized, occurring across various nodes in the network, which diminishes the dependence on DB.

Xiong et al. [17] developed a data trading platform that harnesses smart contracts. DO store their data with dedicated data storage entities. Upon completion of a transaction, DO transfer tokens to DC, who in turn utilize these tokens to access the data. To ensure transactional fairness, an arbitration entity is implemented. If DC discover that the downloaded data fails to meet their criteria, they can seek arbitration. The arbitration entity leverages similarity learning technology to evaluate the consistency of data. In cases of identified inconsistencies, DC are compensated with a refund, while the deposits of DO are seized.

Dai et al. [18] developed a Data Exchange Ecosystem (SDTE) grounded in Ethereum and Intel SGX technologies. In their system, buyers are not granted direct access to raw data. Rather, they receive only the analytical results or processed outputs of the specific data elements they require. Enhanced security is achieved through SGX's authentication mechanisms, which facilitate the secure exchange of keys necessary for encryption. The use of enclaves ensures that both data and key codes remain shielded from external access. This architecture not only guarantees data security and privacy protection but also adeptly addresses the challenges faced by DC and DB in the data transaction process.

III. Preliminaries

A. Personalized Local Differential Privacy

Local Differential Privacy (LDP) [8], recognized as a robust privacy protection mechanism underpinned by a solid mathematical foundation, negates the necessity of trusting any third party and effectively safeguards user data privacy. In LDP algorithms, users apply randomization techniques to introduce noise into their sensitive data at a local level. This altered data is then transmitted to the server, rendering it infeasible for attackers to ascertain the original data of any individual user.

Definition 1. Local Differential Privacy: Considering a specified privacy budget $\varepsilon > 0$, a random algorithm $A : D \to G$ adheres to $\varepsilon-$LDP for users. For any pair of inputs $d \in D$ and $d' \in D$, and for any resultant output $g \in G$, the algorithm meets the ensuing inequality:

$$Pr(A(d) = g) \leq e^{\varepsilon} \times Pr(A(d') = g) \tag{1}$$

Here, Pr denotes the probability derived from the coin toss in mechanism A.

While LDP offers robust privacy protection, it may not always align with the diverse privacy preferences of individual users in practical scenarios. For instance, celebrities and students might have different sensitivities towards their address data. To address this, PLDP [9] model is proposed, providing customization to meet varied user privacy needs. In PLDP, users are required to define two parameters: the security parameter τ and the privacy budget ε.

Definition 2. Personalized Local Differential Privacy: For any two privacy parameters ε and τ of a user, a random algorithm $A : D \to G$ is considered $(\tau, \varepsilon)-$PLDP compliant for that user. For any output $g \in G$, user records $d \in \tau$, and any other value d' within τ, $\tau \in D$, the model adheres to the following inequality:

$$Pr(A(d) = g) \leq e^{\varepsilon} \times Pr(A(d') = g) \tag{2}$$

Here, Pr denotes the probability generated during the coin toss in mechanism A.

In the (τ, ε) −PLDP framework, the parameter τ specifies the range within which user records are indistinguishable from one another. For example, if a user's data is 0.5 and they select τ as $[0, 1]$, then under PLDP, the value 0.5 is indistinguishable from any other values in the $[0, 1]$ range. The parameter ε signifies the level of indistinguishability. If all users set D as their security region and standardize the privacy budget ε, then PLDP effectively becomes equivalent to the standard LDP model.

B. Smart Contract

A smart contract[19], as embedded in blockchain technology, operates in a manner akin to traditional contracts. It is essentially a code that outlines a set of predetermined rules and autonomously enforces them through its execution. On the Ethereum platform [20, 21], a smart contract represents a compilation of code and data situated at a specific blockchain address, often referred to as a contract account. Notably, smart contracts maintain their own balance and can receive transactions, yet they remain beyond the control of any individual entity.

Once deployed on the blockchain, a smart contract is rendered immutable, making it impervious to removal. This feature implies that all interactions with the contract are permanent and irreversible. Such immutability is a fundamental attribute of blockchain technology, ensuring that records inscribed onto the blockchain are resistant to alteration. Consequently, this enforces the reliability and security of the contract's execution.

C. Symmetric Encryption

Symmetric Encryption (SE) [22] is a cryptographic method where the same key is employed for both the encryption and decryption processes. In this approach, both the sender and receiver must possess the same key in advance. This key is utilized to encrypt data for transmission and subsequently decrypt it upon receipt.

Definition 3. Symmetric Encryption:

- $Setup(1^{\lambda}) \to k$: The initialization algorithm. It takes a security parameter 1^{λ} as input and generates the encryption key k.
- $Encrypt(k, M) \to C$: The encryption algorithm. Given the key k and plaintext message M as input, it produces the corresponding ciphertext C.
- $Decrypt(k, C) \to M$: The decryption algorithm. Using the key k and ciphertext C as inputs, it reconstructs the original plaintext message M.

In the selection of a symmetric encryption algorithm, factors like the encryption process and key length play pivotal roles. In this context, the Advanced Encryption Standard (AES) [23, 24] emerges as a superior option. Consequently, we have chosen the AES128 symmetric encryption algorithm to assure the security of data during its transmission and storage phases.

D. Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) [25, 26] is a form of public-key encryption, with its security hinging on the complexity of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) [27]. The core challenge in ECC is to identify an integer k for which $Q = kP$ holds true for two given points P and Q on an elliptic curve, a task that is computationally demanding. The robustness of ECC lies in the inherent difficulty of efficiently resolving the ECDLP within a finite timeframe.

Definition 5. Elliptic Curve Cryptography:

- $KenGen(1^\lambda) \rightarrow (pk, sk)$: The key generation algorithm.
  - Input the security parameter 1^λ.
  - A random elliptic curve E and a generator point G on it are selected. A large prime number n is chosen as the order of points on the curve.
  - A private key $k \in [1, n-1]$ is picked.
  - The public key $Q = [k]G$ is calculated.
  - The outputs are the public key $pk = (E, G, Q)$ and the private key $sk = k$.
- $Enc(M, pk) \rightarrow C$: The encryption algorithm.
  - Takes the plaintext message M and public key $pk = (E, G, Q)$ as inputs.
  - A random number $r \in [1, n-1]$ is generated.
  - The elliptic curve point $C_1 = [r]G$ is computed.
  - The elliptic curve point $S = [r]Q$ is calculated. If S is the point at infinity, a new k is selected and recalculated.
  - The ciphertext $C_2 = M \oplus H(S)$ is formed, where $\oplus$ represents the XOR operation and H is a hash function.
  - The ciphertext $C = (C_1, C_2)$ is produced.
- $Dec(C, sk) \rightarrow M$: The decryption algorithm.
  - Inputs the ciphertext $C = (C_1, C_2)$ and private key $sk = k$.
  - Computes $S = [k]C1$.
  - Derives the plaintext $M = C_2 \oplus H(S)$.
  - Outputs the plaintext message M.

The choice of appropriate elliptic curve parameters is critical for the security of ECC. Hence, in this paper, we have selected the SM2 [28] elliptic curve standard.

E. InterPlanetary File System

The InterPlanetary File System (IPFS) [29] represents a paradigm shift in file storage and sharing, designed as a distributed system that contrasts sharply with traditional centralized storage approaches. Unlike conventional systems where files are stored on a single central server, IPFS distributes files across multiple network nodes. It utilizes content addressing, where a file's unique identifier is derived from its content's hash value. Consequently, even minor alterations in the file content lead to a drastically different hash, thereby assuring the file's uniqueness and integrity. This architecture positions IPFS as a decentralized, secure, and reliable alternative for data storage and distribution.

IV. Trading Framework

The process of our private data trading framework, as depicted in Fig. 2, integrates the value exchange within the blockchain, ensuring that DB computation and publication of results are transparent and subject to user oversight. This placement combats the potential underutilization of DC' budgets by the DB. DB still processes information locally to retain the efficiency of data handling.

In our framework, the transactions are not limited to one-on-one interactions but involve multiple DC and DO, with

Fig. 2. private data trading.

DB facilitating transactions between them. For simplicity, we will first describe a single transaction before expanding on the broader mechanics of value exchange and information processing within the scheme.

A. Value Exchange

In every transaction within our framework, a single Data Consumer (DC_j) engages with multiple Data Owners (DO). For each Data Owner (DO_i), where $1 \leq i \leq n$, the following principles are applied:

- DO_i commits to active participation in the transaction once they receive sufficient compensation, which is calculated based on their personal data valuation, denoted as θ_i. This valuation θ_i indicates the worth of DO_i's data and is bounded within $0 < \underline{\theta} \leq \theta \leq \overline{\theta}$, where $\underline{\theta}$ and $\overline{\theta}$ represent the minimum and maximum value limits, respectively.
- For data privacy, DO_i specifies a secure region $\tau_i \subseteq [-1, 1]$ and a positive privacy budget ε_i before entering the value exchange process. They apply Personalized Local Differential Privacy (PLDP) to their data, guided by these two parameters. The actual value of DO_i's data is thus a function of its privacy protection. A narrower τ_i yields less deviation from the original data and retains a higher true value, whereas a wider τ_i increases data variation post-PLDP, reducing its overall value. For the privacy budget ε_i, a smaller value leads to heavily noised data and lower actual value, while a larger ε_i brings the value closer to the data's original state, signifying a user's consent to limited information disclosure.

Therefore, the actual value of user data, denoted as $v_i = f(\theta_i, \tau_i, \varepsilon_i)$, is expected to conform to certain correlation conditions. Specifically, when a user's data value θ_i is established, the actual value v_i tends to decrease with an increase in the range of the secure region τ_i. Conversely, as the privacy budget ε_i increases, v_i should correspondingly increase. This dual effect ensures that under the umbrella of privacy protection, the real value of user data is optimized, maximizing the extraction of useful information in the value exchange.

In parallel, the valuation θ_i of DO_i's data may inadvertently reveal sensitive information. For instance, a higher valuation in medical health data might suggest a more severe medical condition. Concerns about such privacy breaches might lead some DO_i to underreport their data valuation θ_i intentionally. To counteract this, DB implements incentive measures to encourage DO_i to disclose their true valuations. Specifically, a portion of DC_j's budget is reserved for compensating privacy losses. The compensation received by DO_i is proportionate to the privacy loss θ_i they incur. This additional privacy compensation mechanism is designed to

alleviate DO_i's concerns, encouraging more honest reporting of data valuations, thereby ensuring fairness and transparency in the transaction process.

During the data trading process, DC_j can tailor their resource allocation to align with specific needs and privacy preferences, enabling more nuanced and precise data queries. Specifically, when initiating a transaction, DCj proposes a privacy requirement εdc, stipulating that the privacy budget of DOi involved in the transaction must exceed this value. The lower limit of εdc requires no additional expenditure from DCj. However, as εdc approaches the upper limit of the privacy budget, the cost escalates significantly, potentially reaching infinity. This framework allows DC_j to flexibly balance the privacy level and cost of queries, thereby catering to personalized information needs more effectively.

Integrating these conditions, we conceptualize the value exchange as a 0-1 knapsack problem, where the knapsack's capacity is defined as $B' = B - \frac{(\bar{\varepsilon} - \underline{\varepsilon})e^{\frac{\varepsilon_{dc}-\underline{\varepsilon}}{\bar{\varepsilon}-\underline{\varepsilon}}}}{\bar{\varepsilon}-\varepsilon_{dc}}(Fee + fee)$, each item's weight is given by θ_i, and the value is $v_i = \frac{4\varepsilon_i}{4e^{\frac{w_i}{4}}(\varepsilon_i+1)}\theta_i$. With n items in total, q_i denotes the inclusion of item i in the knapsack. In this scenario, Fee is the intermediary fee by DB, fee is the privacy compensation for DOi, εdc is the privacy requirement of DC_j, w_i is the size of the secure region τ_i for DO_i, and $\underline{\varepsilon}$ and $\bar{\varepsilon}$ represent the lower and upper bounds of the privacy budget. The goal is to maximize the actual value of the data, ensuring the total value does not surpass DC_j's budget.

Definition 7. The optimal value exchange, aimed at maximizing the actual value of data, must adhere to the ensuing equation:

$$\begin{aligned} \underset{q_i, v_i, \theta_i}{\text{maximize}} \quad & \sum_{i=1}^{n} q_i v_i \\ \text{subject to} \quad & \sum_{i=1}^{n} q_i \theta_i \leq B' \\ & \underline{\theta} \leq \theta_i \leq \overline{\theta} \\ & \varepsilon_i > 0 \\ & \underline{\varepsilon} \leq \varepsilon_{dc} \leq \varepsilon_i \leq \bar{\varepsilon} \end{aligned} \tag{3}$$

By resolving this equation, we can deduce a solution that maximizes the actual value of the data while adhering to budgetary constraints. This solution exhibits the following characteristics:

(1) Incentive Compatibility (IC): This attribute ensures that DO_i is motivated to truthfully declare their valuation θ_i. This approach guarantees they receive the maximum privacy compensation $q_i\theta_i + \frac{(\bar{\varepsilon} - \underline{\varepsilon})e^{\frac{\varepsilon_{dc}-\underline{\varepsilon}}{\bar{\varepsilon}-\underline{\varepsilon}}}\theta_i fee}{(\bar{\varepsilon}-\varepsilon_{dc})\sum_{i=1}^{n}\theta_i}$.

(2) Individual Rationality (IR): This principle ensures each Data Owner's willingness to participate, as the benefits of participation outweigh those of non-participation. Assuming non-participation yields a profit of zero, participation results in no privacy breach and entitles them to privacy compensation of $\frac{(\bar{\varepsilon} - \underline{\varepsilon})e^{\frac{\varepsilon_{dc}-\underline{\varepsilon}}{\bar{\varepsilon}-\underline{\varepsilon}}}\theta_i fee}{(\bar{\varepsilon}-\varepsilon_{dc})\sum_{i=1}^{n}\theta_i}$.

(3) Budget Feasibility (BF): This criterion guarantees that the aggregate compensation awarded to DO remains within the fiscal limits of DC_j. Specifically, the total compensation should not surpass the budget B, expressed as $\sum_{i=1}^{n} q_i\theta_i \leq B' < B$.

Satisfying these three properties—Incentive Compatibility, Individual Rationality, and Budget Feasibility—ensures the fairness, effectiveness, and sustainability of the data value trading process. Moreover, it also safeguards the privacy rights of DO and facilitates the smooth progression of data trading activities.

B. Information Processing

In real-world data trading scenarios, DC often face budgetary constraints that prevent them from incorporating data from DO. Consequently, the value exchange mechanism involves a comparatively smaller group of DO than the total number available. This limitation poses challenges in acquiring a comprehensive understanding of the entire dataset through subsequent counting queries. In practical terms, this restriction might lead to queries that diverge significantly from the actual data, obscuring the overarching trends and characteristics of the dataset.

To address this issue, we propose a strategy where DC_j's query requests focus primarily on mean queries and linear predictors. This method enables DC_j to discern the general trends and characteristics of the dataset and facilitates reasonable predictions about the unexplored segments of the data.

In this framework, the Piecewise mechanism with Personalized Local Differential Privacy (PWP) [30] is recognized as a highly effective PLDP algorithm. PWP builds upon the original Piecewise Mechanism [31], adapting it to support PLDP and introducing constraints to ensure the parameters in the probability density function achieve an integral of 1 across the entire range.

According to the predefined data boundaries, the data of each Data Owner (DO_i) is normalized to a specific value within the $[-1, 1]$ range, denoted as d_i. The size of the secure region τ, represented by w_i, and the center point of τ_i, denoted as h_i, are determined based on the privacy parameters (τ_i, ε_i).

The perturbation process within the PWP is outlined in Algorithm I. Initially, DO_i shifts their secure region to a zero-centered symmetric region, effectively moving h_i to the zero point. Consequently, d_i is transformed into $t_i = d_i - h_i$. PWP then processes t_i to produce a sanitized value $\widetilde{t_i}$ within the range $[-C_i, C_i]$, where

$$C_i = \frac{w_i}{2} \cdot \frac{e^{\frac{\varepsilon_i}{2}} + 1}{e^{\frac{\varepsilon_i}{2}} - 1} \tag{4}$$

The probability density function (pdf) of $\widetilde{t_i}$ is a piecewise function as follows:

$$\Pr\left(\widetilde{t_i} = x \mid t_i\right) = \begin{cases} p, & \text{if } x \in [l_i, r_i) \\ \frac{e^{\frac{\varepsilon_i}{2}}}{e^{\frac{\varepsilon_i}{2}}+1}, & \text{if } x \in [-C_i, l_i) \cup [r_i, C_i] \end{cases} \quad (5)$$

where

$$p = \frac{e^{\varepsilon_i} - e^{\frac{\varepsilon_i}{2}}}{w_i(e^{\frac{\varepsilon_i}{2}}+1)} \quad (6)$$

$$l_i = \frac{2t_i \cdot e^{\frac{\varepsilon_i}{2}} - w_i}{2(e^{\frac{\varepsilon_i}{2}}-1)} \quad (7)$$

$$r_i = \frac{2t_i \cdot e^{\frac{\varepsilon_i}{2}} + w_i}{2(e^{\frac{\varepsilon_i}{2}}-1)} \quad (8)$$

Upon determining $\widetilde{t_i}$, DO_i reverts the region to its original position and computes the noisy version of d_i, designated as $\widetilde{d_i} = \widetilde{t_i} + h_i$, which is then expanded to match the data range (Table I).

TABLE I. PWP: PIECEWISE MECHANISM WITH PLDP

Input: Personal privacy parameters (τ_i, ε_i), data d_i of do_i.
Output: sanitized values $\widetilde{d_i}$.
1. $w_i = \lvert\tau_i\rvert$, h_i is the center point of τ_i, $t_i = d_i - h_i$.
2. Sample a random variable a uniformly from $[0, 1]$.
3. If $a < \frac{e^{\frac{\varepsilon_i}{2}}}{e^{\frac{\varepsilon_i}{2}}+1}$:
3.1 Sample $\widetilde{t_i}$ uniformly from $[l_i, r_i)$.
4. Else:
4.1 Sample $\widetilde{t_i}$ uniformly from $[-C_i, l_i) \cup [r_i, C_i]$.
5. $\widetilde{d_i} = \widetilde{t_i} + h_i$.

Definition 8. Algorithm I adheres to the (τ, ε)-PLDP standards for each data owner DO_i with their respective (τ, ε)parameters. Moreover, with an input value of d_i, the algorithm generates a perturbed value such that the expected value $E[\widetilde{d_i}] = d_i$, and the variance is given by:

$$\text{Var}[\widetilde{d_i}] = \frac{(d_i - h_i)}{e^{\frac{\varepsilon_i}{2}}-1} + \frac{w_i^2(e^{\frac{\varepsilon_i}{2}}+3)}{12(e^{\frac{\varepsilon_i}{2}}-1)^2} \cdot \frac{1}{e^{\varepsilon_i}+1}$$

Subsequently, DO_i uploads the perturbed data along with certain non-perturbed feature data to DB. Based on the requests of the data buyer, DB performs queries and sends the encrypted results to the data buyer.

Property 1. Sequential Compositionality: Consider two random algorithms, A_1 and A_2, each conforming to $(\tau, \varepsilon_1)-$PLDP. When these algorithms are sequentially composed as $A = (A_1, A_2)$, the composite satisfies $(\tau, \varepsilon_1 + \varepsilon_2)-$PLDP. A fundamental condition for this property to be valid is that the data d must remain within the secure region τ after being processed by the random algorithm A_1.

Proof

$$\begin{aligned} \frac{\Pr\left(A(d) = g\right)}{\Pr\left(A(d') = g\right)} &= \frac{\Pr\left(A_2(g') = g\right)}{\Pr\left(A_2(g') = g\right)} \times \frac{\Pr\left(A_1(d) = g'\right)}{\Pr\left(A_1(d) = g'\right)} \\ &\leq e^{\varepsilon_2} \times e^{\varepsilon_1} \\ &= e^{\varepsilon_1 + \varepsilon_2} \end{aligned} \quad (9)$$

Thus, $A = (A_1, A_2)$ satisfies $(\tau, \varepsilon_1 + \varepsilon_2)-$PLDP.

After DO_i submits their data, DB reviews it to assess whether further PLDP processing is feasible. Should it be practical to proceed, DO_i, after engaging in multiple transactions, has the option to enhance their privacy level without needing to reapply PLDP with a higher privacy budget and resend the data. Instead, they can authorize DB to apply additional PLDP on the data that has already undergone initial PLDP processing. This approach is designed to minimize costs associated with data retransmission and revalidation, simultaneously reducing the risk of information leakage during the data transfer process.

V. PROTOCOL DETAILS

Prior to examining the specifics of smart contracts and the scheme's overarching process, it is essential to understand the security strategy employed by DO. DO_i may hold various types of data, such as credit card information and health records. Utilizing a single encryption key for all data types presents inherent risks, given that data stored on IPFS is accessible to all, and a key compromise could expose all associated information. To bolster security, DO_i elects to use distinct symmetric keys k for encrypting each data type. This approach ensures that even if one data type's key is compromised, the other data types remain secure. DO_i then amalgamates all the symmetric keys and encrypts them using DB's public key before sending them to DB for data verification. This encryption strategy effectively mitigates potential risks, thereby elevating the overall security level of the data.

With a comprehensive understanding of the DO's security strategy in place, we can now explore the detailed functionalities of smart contracts. These include their pivotal roles in data transactions and the verification process.

A. Smart Contract Functionalities

The smart contract $\widetilde{SC}$ offers the following key functionalities:

Data Broker:

1. 'constructor(_fee)': A constructor function that sets the contract owner, intermediary fee (Fee), and privacy compensation (fee).
2. 'DO_data()': This function enables DB to access information and locations of the first Data Owner in the request queue and integrate them into the DO array.
3. 'update_DO(_site,_i,_change,_introduction)': It allows DB to tag whether the data of DO_i is eligible for subsequent PLDP and includes data introductory details.
4. 'delete_DO(_site,_i)': Used by DB to remove the corresponding DO_i at a given position in instances where data fails verification or when DO_i resubmits data, determining if they should be extracted from the DO array.
5. 'tx_generate()': This function facilitates DB in generating transactions based on the queue and retrieving

query request information from the initial Data Consumer.

6. 'tx_process(_es,_choose,_num,_budget,_fee)': Enables DB to conclude transactions, dispatching ciphertext and the residual budget to DC_j. Should the count of participating DO be zero, the transaction is considered unsuccessful, and the budget is refunded.

Data Owner:

7. 'dataOwner_Join(_value,_cid,_ek,_privacy,_tao)': This function permits DO_i to apply for participation but restricts them from joining directly via the contract address.
8. 'dataOwner_Withdraw()': Enables DO_i to withdraw their earnings, employing a check-influence-swap pattern to mitigate the risk of reentrancy attack vulnerabilities.
9. 'dataOwner_Update(_privacy,_j)': This function allows DO_i to request an increase in privacy budget for a specific record, signaling that the corresponding data is eligible for another round of PLDP. DB then executes the requisite data adjustments locally.

Data Consumer:

10. 'dataConsumer_Purchase(_privacy,_request)payable': Facilitates Data Consumers (DC_j) in submitting purchase requests while prohibiting direct joining through the contract address and barring repeat purchases before the completion of an ongoing transaction.
11. 'dataConsumer_Result()': This function enables DC_j to access the ciphertext of their query results.

B. Overall Process

Now, we will delve into a comprehensive understanding of our solution's operational process by examining the intricacies of data transmission and processing, as well as the pivotal role played by DB. The detailed steps of our solution's overall process are outlined below, with the corresponding sequence diagram depicted in Fig. 3. This thorough exploration will provide insights into how each component interacts and contributes to the efficient functioning of the system.

During the initialization phase, DO_i executes the key initialization algorithm $Setup(1^\lambda)$ to generate their symmetric encryption key k_i. They then apply PWP, informed by their selected privacy parameters (τ_i, ε_i). The data intended for encryption, post-PWP processing, is encrypted using the algorithm $Encrypt(k_i, \widetilde{d}_i|D_i)$, creating the ciphertext C_i. This ciphertext C_i is then uploaded to IPFS, generating a unique hash $hash_i$. Concurrently, DC_j and DB each run the key initialization algorithm $KenGen(1^\lambda)$ to obtain their respective pairs of encryption keys (pk_j, sk_j) and (pk_{DB}, sk_{DB}). Following this, the smart contract $\widetilde{SC}$ is deployed to the blockchain, establishing the intermediary fee (Fee) and privacy compensation (fee).

Data Collection: In the data collection phase, DO_i, having acquired the public key pk_{db} from DB, executes the encryption algorithm $Enc(k_i, pk_{db})$ to produce the encrypted key ek_i. DO_i then applies to join the data trading platform via the smart contract $\widetilde{SC}$, uploading details $(hash_i, ek_i, \theta_i, \varepsilon_i, \tau_i)$ while awaiting verification from DB. Upon receipt of the information $(hash_i, ek_i, \theta_i, \varepsilon_i, \tau_i)$, DB utilizes the decryption algorithm $Dec(ek_i, sk_{db})$ to retrieve DO_i's symmetric key k_i. Following this, DB, referencing $hash_i$, downloads DO_i's ciphertext C_i. Subsequently, by executing $Decrypt(k_i, C_i)$, DB acquires the perturbed data $\widetilde{d}_i$ and the feature data D_i post-PLDP. This data, upon inspection, leads to the approval of DO_i's membership application, coupled with an assessment of the feasibility of further PLDP and inclusion of data introduction details.

Data Purchase: DC_j, utilizing the functionality of the smart contract $\widetilde{SC}$, submits a data query request φ along with their privacy budget ε_j, allocating the budget B for the transaction.

Exchange And Processing: Upon receipt of the query details $(B, \varepsilon_j, \varphi)$ from DC_j, DB implements the value exchange mechanism $E(\theta, \epsilon, \tau, B, \varepsilon_j, Fee, fee)$, resulting in the selection of a set q and the number n' of participating Data Owners for the transaction. This also includes the calculation of the remaining budget b. Following this, based on the query request φ, the operation $P(\widetilde{d}, D, q, \varphi)$ is performed to derive the query result s_j. The result s_j is then encrypted using DC_j's public key pk_j through $Enc(s_j, pk_j)$, generating the ciphertext es_j. DB subsequently transmits the ciphertext es_j and the remaining budget b to DC_j via $\widetilde{SC}$. Upon receipt of es_j, DC_j executes $Dec(es_j, sk_j)$ to retrieve the query result s_j.

Withdraw And Update: After a specified period, DO_i can withdraw their earnings from prior transactions $(q_i\theta_i + \frac{(\bar{\varepsilon}-\underline{\varepsilon})e^{\frac{\varepsilon dc-\underline{\varepsilon}}{\bar{\varepsilon}-\underline{\varepsilon}}}\theta_i fee}{(\bar{\varepsilon}-\varepsilon_{dc})\sum_{i=1}^{n}\theta_i})$ via the $\widetilde{SC}$ contract. Additionally, DO_i can augment their privacy budget through the smart contract $\widetilde{SC}$.

VI. Implementation and Evaluation

In this section, we conduct an experimental evaluation of our scheme by deploying the smart contract on the Sepolia testnet and simulating interactions between the Data Broker, Data Owners, and Data Consumers. A critical aspect of this evaluation involves testing the relative error between the results received by Data Consumers and the actual data outputs.

A. Security Analysis

In the transactions, all cryptographic algorithms used, including SE, ECC, and the hash algorithms of the IPFS, have been extensively validated and are secure. The PLDP algorithm used for data processing also meets the $(\tau, \varepsilon)-$PLDP security standard. The smart contracts employed have undergone unit testing. Therefore, the transaction process is secure. DO' original data is retained locally, and only perturbed data is uploaded. This ensures that DB cannot access the true data of any specific data owner, providing good confidentiality and preventing Man-In-The-Middle (MITM) attacks. Blockchain technology offers tamper-resistance and traceability; events occurring on the blockchain are fully recorded in logs. Thus, the operation information of all entities during the data trading process is completely documented, ensuring good integrity and

preventing any entity from denying their actions during the transaction.

B. Gas Consumption

Within the smart contract framework, DB bears the responsibility for deploying the contract and managing its function executions. Other participants in the network, serving as users on the Ethereum platform, have the flexibility to join at any time. The gas fees associated with deploying the smart contract and executing its various functions are contingent on the specific operations being performed. These costs are comprehensively outlined in Table II, offering a detailed breakdown of the gas consumption for different actions.

TABLE II. TRANSACTION FEE

Function	Transaction Fee (ETH)	Gas Price (Gwei)
Deployed	0.0064124	1.58120
dataOwner_Join	0.0031329	1.59848
dataOwner_Withdraw	0.0000554	1.61893
dataOwner_Update	0.0000630	1.73067
dataConsumer_Purchase	0.0003310	1.61856
DO_data	0.0004541	1.60607
update_DO	0.0002137	1.62734
tx_generate	0.0001068	1.60340
tx_process	0.0027604	1.59467

C. Experimental Design

Experimental Environment. The components for value exchange and information processing, computed locally, are implemented using Python. These components are operated on a computer equipped with an AMD Ryzen 5 5600 6-Core Processor and 32GB of RAM. Each experimental iteration is conducted 50 times to ensure accuracy, with the average results being reported for consistency.

Query Types. Our testing encompassed various query types, including average queries and linear predictors. For average queries, we determined the participating Data Owners through the value exchange mechanism, comparing the perturbed mean with the actual mean. In the case of linear predictors, the last row of data was treated as the predictive value, with other rows representing existing Data Owners. We chose a sensitive attribute as the label and other attributes as features. A linear model was constructed using the least squares method, and its predictive outcomes were compared against actual values.

Metrics. One of the key metrics employed is the Relative Error (RE). This metric is crucial in evaluating the scheme's accuracy in mean estimation, measured as follows:

$$RE = \frac{|T_m - E_m|}{|T_m|} \tag{10}$$

Here, T_m denotes the actual value result, while E_m signifies the perturbed value result.

Dataset. For our experiments, we selected four real-world datasets: the Obesity dataset [32], Student Performance dataset [33], Job Salary dataset [34], and the Obsessive-Compulsive Disorder (OCD) dataset [35]. The details of these datasets are as follows:

- **Obesity Dataset:** The sensitive attribute selected is **age**, ranging from $[15, 56]$. Other attributes are treated as feature attributes and encoded accordingly. After processing, there are a total of 1552 records.
- **Student Performance Dataset:** Here, the sensitive attribute is the **math score**, within the range of $[0, 100]$. Other attributes are designated as feature attributes and are similarly encoded. After processing, there are a total of 964 records.
- **Job Salary Dataset:** The sensitive attribute, **Salary**, is compressed to the range of $[100,000, 180,000]$. Attributes other than the job title are considered feature attributes and are encoded accordingly. After processing, there are a total of 1654 records.
- **OCD Dataset:** For this dataset, **Duration of Symptoms** is the sensitive attribute, with a range of $[6, 240]$. The remaining attributes are classified as feature attributes and encoded as such. After processing, there are a total of 1497 records.

Privacy Parameters(τ_i, ε_i) and Data Value θ_i. The values of the secure region's upper and lower bounds, τ_i, are restricted to the range of $[-1, -0.5, 0, 0.5, 1]$, with specific values being the two closest to d_i, resulting in w_i being set at 0.5. For instance, if $d_i = -0.35$, the secure region would be $[-0.5, 0]$. For the privacy budget ε_i, values are uniformly distributed within $[1, 5]$, randomly selected and rounded to two decimal places. The data value θ_i is randomly determined, selecting integers within the range of $[1, 50]$.

Experiment 1. In our first experiment, we focused on assessing the efficiency of the data processing component. We conducted a thorough comparison between our method and the Laplace mechanism, both of which support continued PLDP. This comparison aimed to highlight the performance disparities between these two data processing approaches under various conditions.

Experiment 2. The second experiment was designed to evaluate the efficiency of the value exchange component. We compared our method against the value exchange mechanisms of FQ and SMQ. The objective of this comparison was to delve into the performance differences among these diverse value exchange methods.

In both experiments, the budget B' allocated by DC for purchasing data varied within the range of $[1000, 20000]$, without any additional privacy budget expenditures.

D. Experimental Result

Experiment 1

As depicted in Fig. 4, the mean query results across the Obesity, Student Performance, Job Salary, and OCD datasets highlight the enhanced precision of our BPPDT method over the LAP approach. Notably, the RE diminishes progressively with the increase in budget B', underscoring our method's ability to leverage additional resources to improve accuracy.

The linear predictor results displayed in Fig. 5 reveal distinct trends across various datasets. In the Obesity and Student Performance datasets, there is a gradual reduction in RE as the budget increases, with the trend eventually plateauing. Our BPPDT method shows superior performance over the LAP-based scheme in these datasets. In the Job Salary dataset, although the LAP scheme starts with an advantage, it experiences significant fluctuations in RE with increased budgets, whereas our method shows a consistent decline in RE. The OCD dataset presents challenges for both methods, with poor performance hinting at weak linear correlations within the data.

Experiment 2

The total value exchange efficiency of our approach is depicted in Fig. 6, where it is evident that our method excels in the value exchange component, attaining the highest level of value exchange efficiency.

The mean query results presented in Fig. 7 demonstrate that across all datasets—Obesity, Student Performance, Job Salary, and OCD—our BPPDT approach consistently yields smaller RE when compared to the FQ and SMQ methods. This advantage is substantial and becomes more pronounced as the budget increases, indicating the superior efficiency of our method in managing value exchange.

The results for the linear predictor as illustrated in Fig. 8 indicate a distinct trend across different datasets. In the Obesity dataset, while the FQ scheme initially exhibits smaller RE at lower budgets, our BPPDT approach surpasses all other schemes with increasing budget. In the Student Performance dataset, the BPPDT method shows competitive REs similar to the SMQ scheme and outperforms other methods, especially at moderate budget levels. Notably, as the budget nears 20000, the SMQ scheme's REs start to decrease significantly. For the Job Salary dataset, the SMQ scheme demonstrates better performance. In contrast, in the OCD dataset, our BPPDT approach maintains commendable performance at lower budgets, showcasing its efficiency.

According to the results of two experiments, the accuracy of mean queries is significantly higher than that of linear queries. This is because linear queries reduce the correlation of the data after submitting perturbed data, whereas mean queries are not affected by this. The more budget DC have, the more data they can purchase, and the higher the accuracy of the data will be. Additionally, the smaller the range of data values, the smaller the added perturbation, and the higher the accuracy of the data. Therefore, this trading model performs better when processing datasets such as grades and salaries.

VII. Conclusion

We introduce a data trading model employing PLDP to achieve a harmonious balance between user-friendliness and privacy protection in data transactions. Our innovative approach not only complies with IC, IR, and BF but also satisfies (τ, ε) −PLDP requirements. It adeptly caters to DC' demands for more detailed queries and fulfills DO' inclination towards augmented privacy budgets. Our experimental findings confirm that our method delivers superior accuracy, even when operating under identical budget constraints. However, the current PLDP algorithms can only operate on numerical data. As future work, we will discuss the selection of privacy parameters in relation to the value of data owners and aim to expand the trading model by incorporating PLDP algorithms suitable for other types of data, as well as addressing more complex query types.

Acknowledgment

We are grateful to the anonymous referees for their invaluable suggestions. This work is partially supported by the Jiangxi Provincial Department of Education research(GJJ2201037).

References

[1] S. Dutta and I. Mia, "The global information technology report 2010–2011," in *World Economic Forum*, Vol. 24, 2011, pp. 331–391.

[2] A. Ghosh and A. Roth, "Selling privacy at auction," in *Proceedings of the 12th ACM conference on Electronic commerce*, 2011, pp. 199–208.

[3] P. Dandekar, N. Fawaz, and S. Ioannidis, "Privacy auctions for recommender systems," *ACM Transactions on Economics and Computation (TEAC)*, Vol. 2, no. 3, 2014, pp. 1–22.

[4] M. Zhang, F. Beltran, and J. Liu, "Selling data at an auction under privacy constraints," in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 669–678.

[5] M. Zhang, J. Liu, K. Feng, F. Beltran, and Z. Zhang, "Smartauction: A blockchain-based secure implementation of private data queries," *Future Generation Computer Systems*, Vol. 138, 2023, pp. 198–211.

[6] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*. Springer, 2006, pp. 1–12.

[7] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[8] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, Vol. 40, no. 3, 2011, pp. 793–826.

[9] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 289–300.

[10] GitHub, Inc., "Github: Where the world builds software," https://github.com, 2008.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*. Springer, 2006, pp. 265–284.

[12] R. Nget, Y. Cao, and M. Yoshikawa, "How to balance privacy and money through pricing mechanism in personal data market," *arXiv preprint arXiv:1705.02982*, 2017.

[13] Z. Jorgensen, T. Yu, and G. Cormode, "Conservative or liberal? personalized differential privacy," in *2015 IEEE 31St international conference on data engineering*. IEEE, 2015, pp. 1023–1034.

[14] W. Wang, L. Ying, and J. Zhang, "Buying data from privacy-aware individuals: The effect of negative payments," in *Web and Internet Economics: 12th International Conference, WINE 2016, Montreal, Canada, December 11-14, 2016, Proceedings 12*. Springer, 2016, pp. 87–101.

[15] A. Fallah, A. Makhdoumi, A. Malekian, and A. Ozdaglar, "Optimal and differentially private data acquisition: Central and local mechanisms," *Operations Research*, 2023.

[16] W. Li, M. Zhang, L. Zhang, and J. Liu, "Integrated private data trading systems for data marketplaces," *arXiv preprint arXiv:2307.16317*, 2023.

[17] W. Xiong and L. Xiong, "Smart contract based data trading mode using blockchain and machine learning," *IEEE Access*, Vol. 7, 2019, pp. 102 331–102 344.

[18] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, "Sdte: A secure blockchain-based data trading ecosystem," *IEEE Transactions on Information Forensics and Security*, Vol. 15, 2019, pp. 725–737.

[19] N. Szabo, "Formalizing and securing relationships on public networks," *First monday*, 1997.

[20] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, Vol. 151, no. 2014, 2014, pp. 1–32.

[21] C. Dannen, *Introducing Ethereum and solidity*. Springer, 2017, Vol. 1.

[22] M. N. Alenezi, H. Alabdulrazzaq, and N. Q. Mohammad, "Symmetric encryption algorithms: Review and evaluation study," *International Journal of Communication Networks and Information Security*, Vol. 12, no. 2, 2020, pp. 256–272.

[23] D. Selent, "Advanced encryption standard," *Rivier Academic Journal*, Vol. 6, no. 2, 2010, pp. 1–14.

[24] Christof Paar, Jan Pelzl, and Tim Güneysu. The advanced encryption standard (aes). In *Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms*, pages 111–146. Springer, 2024.

[25] V. Kapoor, V. S. Abraham, and R. Singh, "Elliptic curve cryptography," *Ubiquity*, Vol. 2008, no. May, 2008, pp. 1–8.

[26] U Vijay Nikhil, Z Stamenkovic, and SP Raja. A study of elliptic curve cryptography and its applications. *International Journal of Image and Graphics*, page 2550062, 2024.

[27] S. D. Galbraith and P. Gaudry, "Recent progress on the elliptic curve discrete logarithm problem," *Designs, Codes and Cryptography*, Vol. 78, 2016, pp. 51–72.

[28] National Standardization Management Committee of China, "Information security technology—sm2 cryptographic algorithm usage specification," Date of Issue: 2017-12-29, Implementation Date: 2018-07-01, General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China; Standardization Administration of China, Technical Report GB/T 35276-2017, 12 2017, chinese Standard Classification (CCS): L80, International Standard Classification (ICS): 35.040.

[29] Y. Psaras and D. Dias, "The interplanetary file system and the filecoin network," in *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*. IEEE, 2020, pp. 80–80.

[30] Q. Xue, Y. Zhu, and J. Wang, "Mean estimation over numeric data with personalized local differential privacy," *Frontiers of Computer Science*, Vol. 16, 2022, pp. 1–10.

[31] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 638–649.

[32] F. M. Palechor and A. de la Hoz Manotas, "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, peru and mexico," *Data in brief*, Vol. 25, 2019, p. 104344.

[33] J. Seshapanpu. (2023) Students performance in exams. Accessed 2023. [Online]. Available: https://www.kaggle.com/datasets/spscientist/students-performance-in-exams

[34] RANDOMARNAB. (2023) Data science salaries 2023. Accessed 2023. [Online]. Available: https://www.kaggle.com/datasets/arnabchaki/data-science-salaries-2023

[35] S. H. CHOWDHURY. (2023) Ocd patient dataset: Demographics & clinical data. Accessed 2023. [Online]. Available: https://www.kaggle.com/datasets/ohinhaque/ocd-patient-dataset-demographics-and-clinical-data

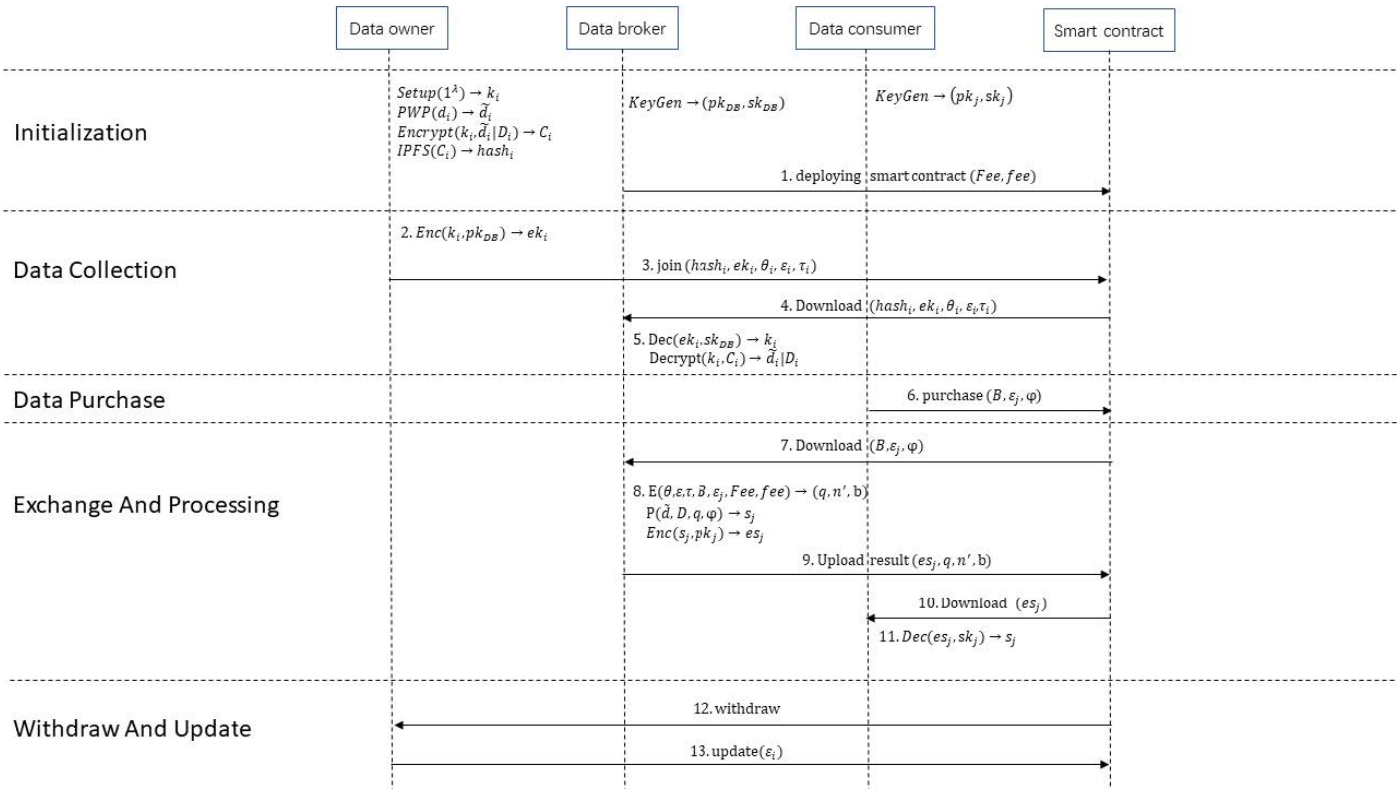

Fig. 3. System sequence diagram.

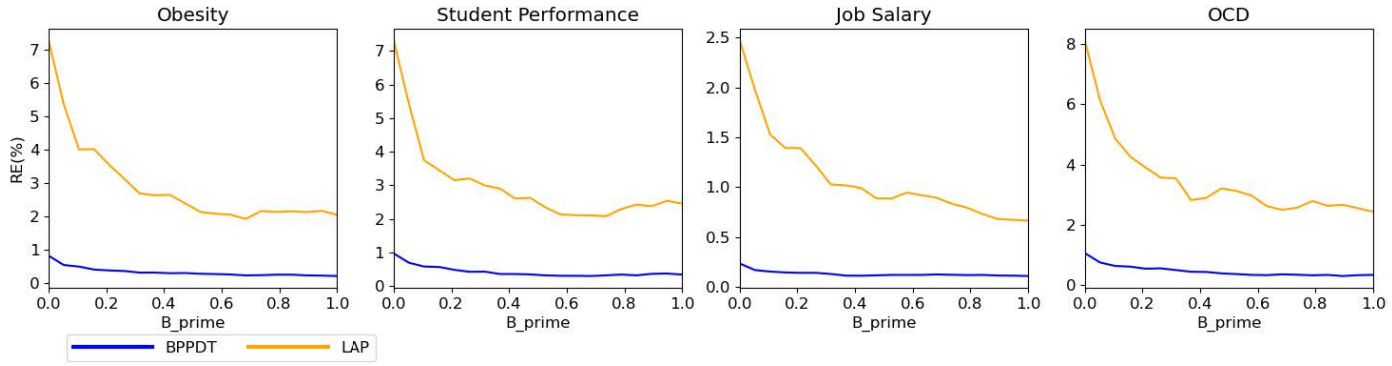

Fig. 4. Experiment 1 mean.

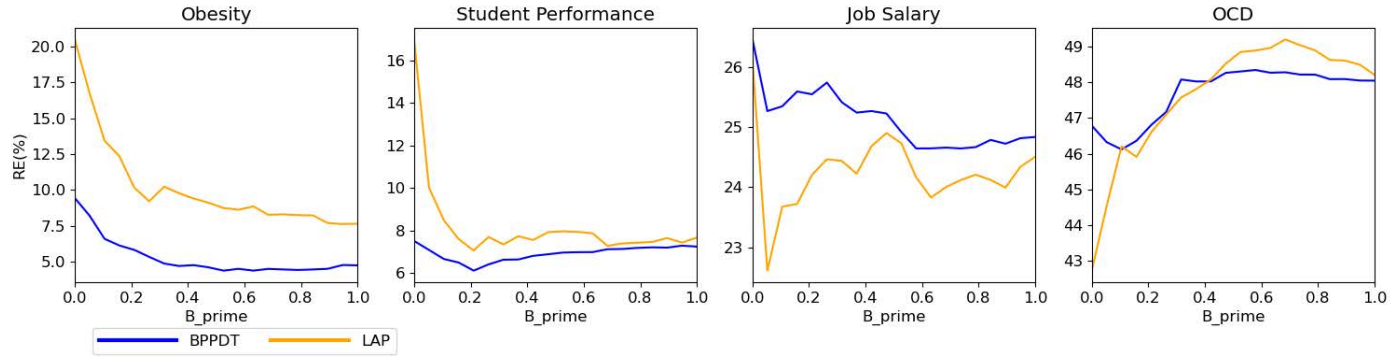

Fig. 5. Experiment 1 linear.

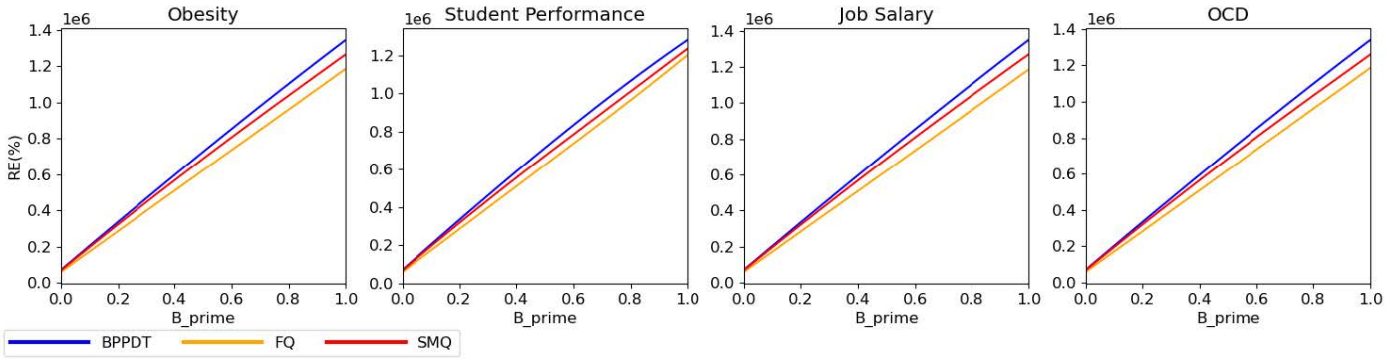

Fig. 6. Experiment 2 value.

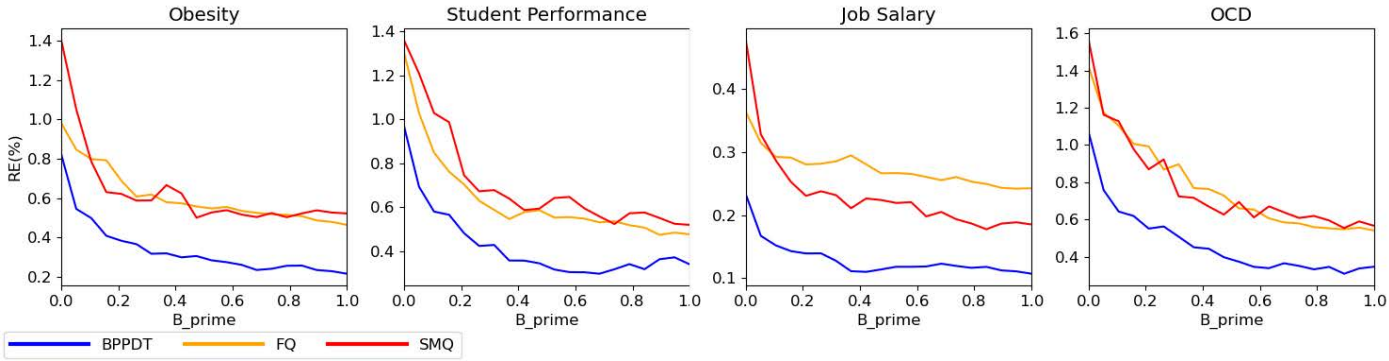

Fig. 7. Experiment 2 mean.

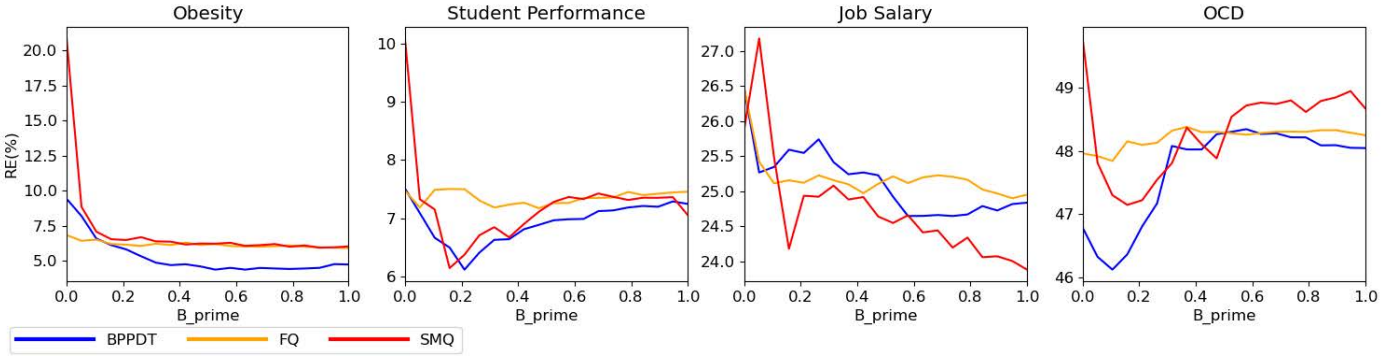

Fig. 8. Experiment 2 linear.