Towards Correlated Queries on Trading of Private Web Browsing History

Hui Cai*, Fan Ye†, Yuanyuan Yang†, Yanmin Zhu*, Jie Li*
*Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
†Department of Electrical and Computer Engineering, Stony Brook University, USA

***Abstract*—With the commoditization of private data, data trading in consideration of user privacy protection has become a fascinating research topic. The trading for private web browsing histories brings huge economic value to data consumers when leveraged by targeted advertising. In this paper, we study the trading of multiple correlated queries on private web browsing history data. We propose *TERBE*, which is a novel trading framework for correlaTed quEries based on pRivate web Browsing historiEs. *TERBE* first devises a modified matrix mechanism to perturb query answers. It then quantifies privacy loss under the relaxation of classical differential privacy and a newly devised mechanism with relaxed matrix sensitivity, and further compensates data owners for their diverse privacy losses in a satisfying manner. Through real-data based experiments, our analysis and evaluation results demonstrate that *TERBE* balances total error and privacy preferences well within acceptable running time, and also achieves all desired economic properties of budget balance, individual rationality, and truthfulness.**

***Index Terms*—Data Trading, Web Browsing History, Data Privacy.**

I. Introduction

Recent years have witnessed explosive growth of users' web browsing histories [1] with the advent of the era of big data. Typical examples of these text-based data include web users' purchasing records on E-commerce platforms like Amazon and browsing records on financial websites like CNN markets [2]. Furthermore, these data have tremendous economic value on data consumers like advertisers. For example, the advertiser saves advertising costs the most when he leverages extracted data to carry out behavior targeting on Twitter [3]. In particular, after the advertiser selects behaviors or preferences of his major target audience on an ad platform, Twitter would deliver his advertisements to those interested users so as to cut down costs of ad impression significantly. However, many data owners are reluctant to share their private data due to privacy concern. To facilitate the commoditization of these private data, more and more data trading markets have emerged to build the bridge between data owners and data consumers. On the one hand, data owners are willing to empower the reliable data broker to access their private data as long as they obtain desired privacy protection and reasonable monetary compensation. On the other hand, the data broker charges data consumers fees on queries as data owners' privacy compensation.

Yanmin Zhu is the corresponding author.

To further extract features of his major target audience, the advertiser usually issues multiple correlated queries to the data broker. For instance, he needs the age distribution histogram of his major target audience with a high consumption ability, and thus the data broker divides his audience into multiple ranges based on the attribute 'age'. The reason why he would not issue single queries one by one lies in the fact that the data broker answers each query independently without consideration of the correlation within multiple queries. Thus, the total error becomes much larger with more queries. Fortunately, issuing multiple queries simultaneously probably produces a smaller total error by leveraging the correlations reasonably.

In this paper, we investigate a novel trading problem of multiple correlated queries that maximizes data consumer's utility while guaranteeing data owners' privacy preferences and acceptable time complexity. Three major challenges must be addressed. The first challenge is to quantify each data owner's privacy loss on a set of correlated queries. The main difference between the trading of sensitive private data and traditional goods lies in the possible privacy loss and thus indispensable privacy compensation. Existing private data trading [4]–[7] adopts classical differential privacy [8] to measure each data owner's privacy loss in terms of a single query. However, previous work cannot be directly applied here because simply summing privacy losses of individual queries is not equivalent to the total loss when queries are correlated. Therefore, it is still unsolved work to leverage correlations between multiple queries to quantify each data owner's privacy loss.

The second challenge is on making a suitable trade-off between the data consumer's utility and data owners' privacy protection. A traditional solution is to enforce the same data perturbation mechanism for each correlated query, *i.e.*, add an independent Laplace noise in order to satisfy data owners' privacy preference to some degree. However, such a mechanism probably produces a larger error with more queries, and thus leads to more degraded utility. Although previous matrix mechanism [9] proposed a utility-maximizing solution by incorporating the correlation within multiple queries while achieving ϵ-differential privacy, their work cannot be applied to a realistic scenario because of high time complexity. Thus, it is nontrivial to maximize the data consumer's utility while guaranteeing data owners' privacy requirements in a practical scenario.

The last but not least challenge comes from preventing data

owners' possible strategic behaviors. Previous works [4], [5] consider that each data owner with diverse privacy concern is compensated by a fixed privacy cost, and suffers bounded or unbound privacy loss based on his privacy strategy. However, the assumption possibly leads to biased selection when applied to our case. Because each data owner suffers bounded privacy loss in our scenario, fixed compensation cost possibly eliminates some conservative data owners who are unsatisfied with low privacy compensation. Consequently, a fraction of users' records cannot reflect the whole population, and thus lead to biased results. Therefore, a feasible privacy compensation mechanism should compensate data owners by auction so as to prevent them game the data market. Each player may report a higher value than actual privacy cost for higher benefit by auction. Thus, the proposed mechanism has to satisfy the property of truthfulness. This further increases the complexity of designing a practical data trading mechanism.

In this paper, by jointly considering the above three challenges, we propose *TERBE*, a novel framework for trading correlaTed quEries based on pRivate web Browsing historiEs, which consists of a data perturbation mechanism and a privacy compensation mechanism. *TERBE* first employs the strategy matrix to depict correlations between multiple queries. Due to high total error of the traditional mechanism and high time complexity of the optimal mechanism, we devise a modified matrix mechanism by relaxing the sensitivity of the strategy matrix, so as to guarantee practical running time and a comparable total error with the optimal mechanism, but at the cost of the increase of acceptable privacy loss. To comply with this new matrix mechanism, *TERBE* next defines each data owner's privacy loss on multiple correlated queries based on the relaxation of classical differential privacy, and further gives its upper bound. According to the upper bound, we propose a reasonable privacy compensation mechanism, which satisfies all desired economic properties.

We highlight main contributions as follows.

- To the best of our knowledge, *TERBE* is the first work that studies the trading of multiple correlated queries based on private web browsing histories from the perspective of a data broker in a data market.
- We propose a new matrix mechanism to make a balance between the total error and acceptable privacy loss, with reasonable time complexity. Besides, *TERBE* quantifies each data owner's privacy loss based on the relaxation of classical differential privacy and devised mechanism. In addition, data owners fairly receive distinct privacy compensation rather than the same compensation because of diverse upper bounds of their privacy losses on multiple correlated queries in a satisfying manner.
- Our real-data based experiments and analysis demonstrate *TERBE* decreases 66.67% of total error than traditional mechanism at least. Besides, it only takes *TERBE* 8% of running time of the optimal mechanism with more queries. Through rigorously theoretical analysis, *TERBE* achieves an acceptable (ϵ, δ)-differential privacy and all desired economics properties.

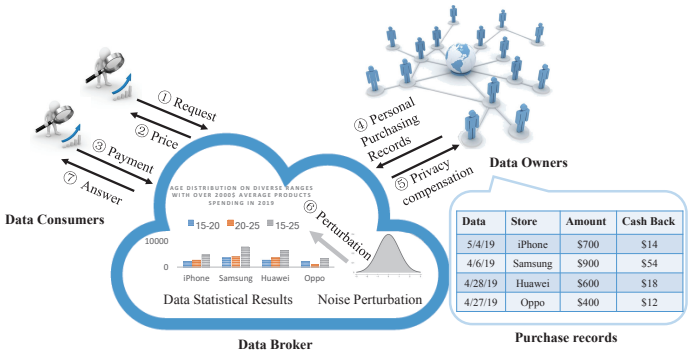

Figure 1. The system model for data markets trading multiple correlated queries based on personal purchasing records.

II. Data Trading Model and Problem Formulation

In this section, we introduce data trading model, correlated differential privacy, matrix mechanism, and design objectives.

A. Trading Model Based on Correlated Queries

1) Data market: As illustrated in Fig. 1, we consider a data market consisting of data owners (*i.e.*, web users), data consumers and a data broker, as a trustworthy third-party platform like Acxiom [10], which is allowed to access users' web browsing histories.

The data broker first procures web browsing histories, which are usually comprised of the page content and corresponding URLs, such as purchase records of electronics or browsing records of financial websites [11], from the data consumer's target audience. We use $X = (t, e, \nu, a)$ to represent extracted key information about any piece of browsing history from any data owner, indicating that the user with the attribute a has the feature ν for the event e on the time t. For example, it takes $\nu = 700\$$ for a 25-year-old female user to purchase iPhone XS on June 5, 2019.

2) Traded data: The data consumer issues multiple correlated queries to the data broker so as to pinpoint his major target audience. In addition, a natural solution to divide the target audience is that he specifies his interested ranges $\boldsymbol{R} = \{R_1, R_2, \ldots, R_m\}$ over some attribute a (*e.g.*, age, gender or income). Next, the data broker aggregates corresponding multiple statistical results based on key information matrix $\boldsymbol{X} = \{X_1, X_2, \ldots\}$ from all related data owners' browsing histories. For instance, Apple Inc. wonders the proportion of target users who have more than $2000\$$ average product spending in 2019 over interested ranges $\boldsymbol{R} = \{$[15-20,female],[15-20,male],[20-25,female],[20-25,male],[15-25,female],[15-25,male]$\}$ about the attribute 'age' and 'gender'. Note that the reason why the data consumer makes correlated queries is that the accumulation of multiple perturbed query answers probably leads to a larger error. Hence, he is willing to make another correlated query in

The sum of some query answers is equivalent to another query answer.

Each range corresponds to one query, and there are 6 correlated queries in total. For example, the query result on [15-25,female] should be equivalent to the sum of query results on [15-20,female] and [20-25,female].

order to acquire a more accurate query result on his interested larger range.

Consequently, we consider customized procurement request by a data consumer is denoted as $\mathcal{Q} = (q, \boldsymbol{R}, \phi, v)$. Here, q determines the group of target users by specifying constraint information (t', e', ν') (*e.g.*, target users with more than 2000$ average costs on purchasing iPhone products in 2019), and $\boldsymbol{R}$ further helps extract features of data consumer's largest group of target users. Besides, ϕ is a numeric function mapping browsing histories $\boldsymbol{X}_q$ of specified target users to a vector of statistical results corresponding to ranges $\boldsymbol{R}$. Finally, v is his maximum tolerant variance of noise added to any true query answer $\zeta_j \in \phi(\boldsymbol{X}_q)$. We consider the data consumer only focuses on multiple correlated counting queries about target audience, and other more complex data analyses like weighted sum and probability distribution fitting are subject to our future work.

To represent multiple correlated counting queries given ranges $\boldsymbol{R}$, we first define mutually-exclusive ranges $\overline{\boldsymbol{R}}$, which is the set of all disjoint ranges from $\boldsymbol{R}$. Let the size of $\overline{\boldsymbol{R}}$ be $|\overline{\boldsymbol{R}}| = n \leq m$, and then $R_i \bigcap R_j = \emptyset$, for any $R_i, R_j \in \overline{\boldsymbol{R}} \subset \boldsymbol{R}$. Next, the data vector corresponding to $\overline{\boldsymbol{R}}$ is denoted as $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^t$, which reflects true statistical results on each range from $\overline{\boldsymbol{R}}$. Then, for any range $R_j \in (\boldsymbol{R} \backslash \overline{\boldsymbol{R}})$, the corresponding statisitcal result on the query j is denoted as $\zeta_j = \sum_i^n q_{ij} x_i$, where q_{ij} is the weight of each data element $x_i \in \boldsymbol{x}$ on the range R_j, $q_{ij} \in \{0, 1\}$, and finally generates the query matrix $\boldsymbol{Q} \in \mathbb{R}^{m*n}$. Thus, the true answer vector on these ranges is $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_m]^t = \boldsymbol{Q}\boldsymbol{x}$. Because an attacker with prior knowledge probably infers any user's real identity based on the true query answer [4]–[7], [11], [12], the data broker answers correlated queries with a randomized mechanism $\mathcal{M}$, and returns perturbed answer vector $\mathcal{M}(\boldsymbol{\zeta})$.

3) Privacy information: Each data owner $i \in \mathcal{N} = \{1, 2 \ldots, L\}$ has a privacy budget ξ_i, indicating his maximum tolerant privacy loss to any data consumer's queries $\mathcal{Q}$. Besides, he submits the bid price c_i as his claimed cost of unit privacy loss (*i.e.*, privacy cost), which probably deviates from the real cost $\overline{c}_i$ in terms of his strategic behavior. After returning the perturbed answer vector $\mathcal{M}(\boldsymbol{\zeta})$, the data broker compensates each data owner with $\psi(\mathcal{Q})$ for his privacy leak $\xi_i(\mathcal{M})$, and charges the data consumer $\pi(\mathcal{Q})$. Clearly, a smaller variance v leads to a larger privacy loss, and thus produces a higher privacy compensation $\psi(\mathcal{Q})$. In addition, a more accurate answer set is returned, and results in a higher charged price $\pi(\mathcal{Q})$. Besides, each data owner's utility is defined as $u_i = \psi(\mathcal{Q}) - \overline{c}_i \cdot \xi_i(\mathcal{M})$.

B. ϵ-Correlated Differential Privacy

The celebrated differential privacy [8] has been adopted widely to protect user privacy, which has to be satisfied as a strong privacy constraint. Specifically, the randomized mechanism has to generate two close answers with a high probability with or without any data owner's any piece of browsing history, which further limits the probability of the attacker's inferring the user's private information.

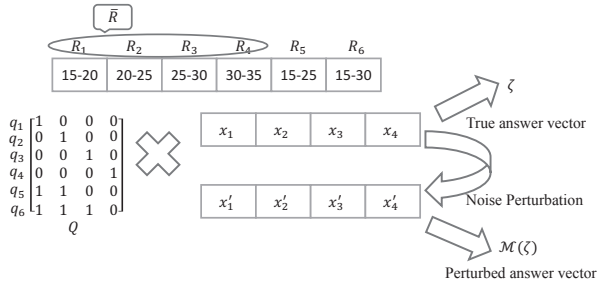

Figure 2. A toy example for the traditional mechanism. There are 6 correlated queries over the attribute 'age'. Besides, each of the last two queries can be represented by the first four ones. After adding Laplace noise, the true answer vector $\boldsymbol{\zeta} = \{\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5, \zeta_6\}$ becomes the perturbed one $\boldsymbol{\zeta}' = \{\zeta_1', \zeta_2', \zeta_3', \zeta_4', \zeta_5', \zeta_6'\}$.

Let ϵ denote privacy budget achieved by $\mathcal{M}$, we have to guarantee $\epsilon \leq \xi$. For our case, we generalize the classic ϵ-differential privacy by incorporating the correlation among multiple queries, namely ϵ-correlated differential privacy. Let $\boldsymbol{x}'$ denote the neighboring data vector for $\boldsymbol{x}$ without the data owner i's any piece of browsing history X_i. Hence, we have $||\boldsymbol{x} - \boldsymbol{x}'||_1 = 1$ due to the disjoint range each data element $x_i \in \boldsymbol{x}$ belongs to.

Definition 1. **(ϵ-*Correlated Differential Privacy*)**. *A randomized mechanism $\mathcal{M}$ satisfies ϵ-correlated differential privacy if for any two neighboring data vectors $\boldsymbol{x}, \boldsymbol{x}'$, a query matrix $\boldsymbol{Q}$ representing the correlation between multiple queries, and any possible output vector $\boldsymbol{S}$, we have*

$$\frac{Pr(\mathcal{M}(\boldsymbol{Q}\boldsymbol{x}) = \boldsymbol{S})}{Pr(\mathcal{M}(\boldsymbol{Q}\boldsymbol{x}') = \boldsymbol{S})} \leq e^{\epsilon}, \tag{1}$$

where a smaller private budget ϵ indicates a smaller privacy loss, and causes a larger error.

Besides, the correlation among multiple queries by the query matrix $\boldsymbol{Q}$ makes our work different from existing work [8]. To achieve the above ϵ-correlated differential privacy, the traditional solution is to add a Laplace noise for each query answer independently, determined by both privacy budget ϵ and the sensitivity of the mapping function $\Delta\phi$, where the sensitivity $\Delta\phi$ over multiple correlated queries caused by modifying any piece of browsing history X_i is:

$$\Delta\phi = \max_{||\boldsymbol{x} - \boldsymbol{x}'||_1 = 1} ||\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x}'||_1 = \max_i \sum_{j=1}^{m} |q_{ij}|, \tag{2}$$

where the sensitivity of query matrix $\Delta_{\boldsymbol{Q}}$ is $\max_i \sum_{j=1}^{m} |q_{ij}|$. The traditional solution for a toy example is given in Fig. 2.

Lemma 1. *For any numeric function ϕ over multiple correlated queries, which are represented by the product between query matrix $\boldsymbol{Q}$ and data vector $\boldsymbol{x}$, the traditional mechanism*

The weight q_{ij} is known, and naturally determined by the correlation among multiple queries.

Also called privacy preference. A smaller privacy budget means the user needs stronger privacy protection while the data broker leverages his private data.

$\mathcal{M}$

$$\mathcal{M}(\boldsymbol{Q}, \boldsymbol{x}) = \phi(\boldsymbol{Q}, \boldsymbol{x}) + \frac{\Delta_{\boldsymbol{Q}}}{\epsilon}\tilde{\boldsymbol{z}} = \boldsymbol{Q}\boldsymbol{x} + \frac{\Delta_{\boldsymbol{Q}}}{\epsilon}\tilde{\boldsymbol{z}}, \tag{3}$$

where $\tilde{\boldsymbol{z}} \in \mathbb{R}^m$ consists of m independent random noises drawn from a m-dimensional Laplace distribution with mean $[0, \ldots, 0]^T$ and scale $[1, \ldots, 1]^T$. $\mathcal{M}$ satisfies ϵ-correlated differential privacy.

Proof. Let $h(\beta) = \frac{1}{(2\lambda)^n} e^{-\frac{||\beta||_1}{\lambda}}$ represent the probability density function of a m-dimensional Laplace distribution where the mean is 0 and the scale is λ, where β denotes the vector of m added Laplace noises. Then we have:

$$\begin{aligned}
&\frac{Pr(\boldsymbol{Q}\boldsymbol{x} + \frac{\Delta_{\boldsymbol{Q}}}{\epsilon}\tilde{\boldsymbol{z}} = \boldsymbol{Q}\boldsymbol{x} + \beta)}{Pr(\boldsymbol{Q}\boldsymbol{x}' + \frac{\Delta_{\boldsymbol{Q}}}{\epsilon}\tilde{\boldsymbol{z}} = \boldsymbol{Q}\boldsymbol{x} + \beta)} \\
&= \frac{1/(2\lambda)^n exp(-\frac{||\beta||_1}{\lambda})}{1/(2\lambda)^n exp(-\frac{||\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x}' + \beta||_1}{\lambda})} \\
&= exp(\frac{||\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x}' + \beta||_1 - ||\beta||_1}{\lambda}) \\
&\leq exp(\frac{||\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x}'||_1}{\lambda}) \leq exp(\epsilon \cdot \frac{||\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x}'||_1}{\Delta_{\boldsymbol{Q}}}) \leq e^{\epsilon}
\end{aligned} \tag{4}$$

, where the first and third inequation hold because of triangle inequality and $||\boldsymbol{x} - \boldsymbol{x}'||_1 = 1$, respectively. Clearly, the traditional mechanism also satisfies ξ-correlated differential privacy because of $\epsilon \leq \xi$. □

In general, we exploit the square error $\mathcal{E}_{\boldsymbol{Q}}$ to measure the data consumer's utility, *i.e.*, $\mathcal{E}(\boldsymbol{Q}) = ||\mathcal{M}(\boldsymbol{Q}, \boldsymbol{x}) - \phi(\boldsymbol{Q}, \boldsymbol{x})||_2^2$. Clearly, a smaller total error indicates a higher utility for each data consumer.

C. Matrix Mechanism and Total Error

Inspired by existing work [9], we adopt an efficient matrix mechanism in order to produce a smaller square error than the above traditional solution. The main idea is to add the linear combination of m independent random noises to each query answer so as to exploit the correlation of multiple queries. Specifically, we have to find a coefficient matrix $\boldsymbol{G}^+$, which consists of a full rank strategy matrix $\boldsymbol{G} \in \mathbb{R}^{m*n}$, $m \geq n$, and $\boldsymbol{G}^+ = (\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{G}^t$. Therefore, the matrix mechanism is given in Definition 2.

Definition 2. (***Matrix Mechanism [9]***). *Given the true answer vector $\boldsymbol{Q}\boldsymbol{x}$ and a strategy matrix $\boldsymbol{G}$, the matrix mechanism $\mathcal{M}_{\boldsymbol{G}}(\boldsymbol{Q}, \boldsymbol{x})$ is defined as:*

$$\begin{aligned}
\mathcal{M}_{\boldsymbol{G}}(\boldsymbol{Q}, \boldsymbol{x}) &= \boldsymbol{Q}\boldsymbol{x} + (\frac{\Delta_{\boldsymbol{G}}}{\epsilon})\boldsymbol{Q}\boldsymbol{G}^+\tilde{\boldsymbol{z}} \\
&= \boldsymbol{Q}\boldsymbol{G}^+(\boldsymbol{G}\boldsymbol{x} + (\frac{\Delta_{\boldsymbol{G}}}{\epsilon})\tilde{\boldsymbol{z}}) \\
&= \boldsymbol{Q}\boldsymbol{G}^+\mathcal{M}(\boldsymbol{G}, \boldsymbol{x}),
\end{aligned} \tag{5}$$

where $\Delta_{\boldsymbol{G}}$ means the sensitivity of $\boldsymbol{G}$, and $\Delta_{\boldsymbol{G}} = \max_i \sum_{j=1}^m |g_{ij}|$, where g_{ij} represents any matrix element of $\boldsymbol{G}$. Besides, estimated values $\hat{\boldsymbol{x}}_{\boldsymbol{G}}$ for data vector $\boldsymbol{x}$ is denoted as $\hat{\boldsymbol{x}}_{\boldsymbol{G}} = \boldsymbol{G}^+\mathcal{M}(\boldsymbol{G}, \boldsymbol{x})$. Clearly, we have $\Delta_{\boldsymbol{G}} > 0$ because $\Delta_{\boldsymbol{G}} = 0$ means there is no any added random noise at all. Note that the matrix mechanism also satisfies ϵ-correlated differential privacy, because it is actually a linear combination of outputs by the traditional mechanism $\mathcal{M}(\boldsymbol{Q}, \boldsymbol{x})$ which is proved to be ϵ-correlated differential privacy. Next, we define the total error based on the above matrix mechanism.

Definition 3. (***Total Error***). *For each query $\boldsymbol{q}$ as each row of $\boldsymbol{Q}$, the corresponding error is defined as $\gamma_G(\boldsymbol{q}) = E[(\boldsymbol{q}\boldsymbol{x} - \boldsymbol{q}\hat{\boldsymbol{x}}_{\boldsymbol{G}})^2]$. Therefore, the total error on the query matrix $\boldsymbol{Q}$ is given as $\mathcal{E}_G(\boldsymbol{Q}) = \sum_{\boldsymbol{q}_i \in \boldsymbol{Q}} \gamma_G(\boldsymbol{q}_i)$.*

Lemma 2. *Given the strategy matrix $\boldsymbol{G}$, for any query $\boldsymbol{q} \in \boldsymbol{Q}$, the error is calculated as $\gamma_G(\boldsymbol{q}) = (\frac{\Delta_{\boldsymbol{G}}}{\epsilon})^2 2\boldsymbol{q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{q}^t$. Hence, the expectation of the total error is equal to $\mathcal{E}_G(\boldsymbol{Q}) = (\frac{2}{\epsilon^2})\Delta_{\boldsymbol{G}}^2 trace(\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)$.*

Proof.

$$\begin{aligned}
\gamma_G(\boldsymbol{q}) &= \text{var}(\boldsymbol{q}\hat{\boldsymbol{x}}_{\boldsymbol{G}}) \\
&= \text{var}(\boldsymbol{q}\boldsymbol{x} + \frac{\Delta_{\boldsymbol{G}}}{\epsilon}\boldsymbol{q}\boldsymbol{G}^+\tilde{\boldsymbol{z}}) = (\frac{\Delta_{\boldsymbol{G}}}{\epsilon})^2 \text{var}(\boldsymbol{q}\boldsymbol{G}^+\tilde{\boldsymbol{z}}),
\end{aligned} \tag{6}$$

where we can find $\text{var}(\boldsymbol{q}\boldsymbol{G}^+\tilde{\boldsymbol{z}}) = \boldsymbol{q}\boldsymbol{G}^+\text{var}(\tilde{\boldsymbol{z}})(\boldsymbol{q}\boldsymbol{G}^+)^t = \boldsymbol{q}\boldsymbol{G}^+ 2I_m(\boldsymbol{q}\boldsymbol{G}^+)^t = 2\boldsymbol{q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{q}^t$. Thus, we have $\gamma_G(\boldsymbol{q}) = (\frac{\Delta_{\boldsymbol{G}}}{\epsilon})^2 2\boldsymbol{q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{q}^t$. Since each query $\boldsymbol{q}$ belongs to each row of $\boldsymbol{Q}$, the error $\gamma_G(\boldsymbol{q}_i)$ is actually the i^{th} diagonal element of the diagonal matrix $(\frac{2}{\epsilon^2})\Delta_{\boldsymbol{G}}^2(\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)$. Consequently, the total error $\mathcal{E}_G(\boldsymbol{Q})$ is the sum of all diagonal elements from the diagonal matrix, which can be represented as the trace of this matrix. □

According to similar derivation, the expectation of the total error for traditional mechanism is $\mathcal{E}(\boldsymbol{Q}) = \frac{2m}{\epsilon^2}\Delta_{\boldsymbol{Q}}^2$. By comparing the two expectation values, we can find the reason why it is possible for the matrix mechanism to produce a smaller error compared with the traditional solution. Let $W = (\boldsymbol{G}^t\boldsymbol{G})^{-1}$ denote the combinatorial term of $\gamma_G(\boldsymbol{q})$, and it exactly depicts covariance of estimated values $\hat{\boldsymbol{x}}_{\boldsymbol{G}}$. Specifically, the diagonal term w_{ii} of W means the variance about the estimate of x_i from $\hat{\boldsymbol{x}}_{\boldsymbol{G}}$, and the off-diagonal element w_{ij} indicates the covariance of the estimate of x_i and x_j. Hence, we have $w_{ii} > 0$, but w_{ij} is possibly less than zero. Moreover, for each query $\boldsymbol{q}$, the error term of $\gamma_G(\boldsymbol{q})$ is:

$$\boldsymbol{q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{q}^t = \sum_{i<n} q_i^2 w_{ii} + \sum_{i<j} 2q_i q_j w_{ij}. \tag{7}$$

Since w_{ij} may be negative, the error term is possibly smaller than 1 when the query error on each disjoint range is high, but the accuracy on other correlated queries (*i.e.*, the linear combination of query answers on disjoint ranges) is significantly improved. Therefore, it is possible to produce a smaller total error when carefully picking the strategy matrix $\boldsymbol{G}$ such that $\Delta_{\boldsymbol{G}} \leq \Delta_{\boldsymbol{Q}}$ and $\text{trace}(\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t) \leq m$.

D. Design Objectives

In this work, we aim at designing a practical trading mechanism for multiple correlated queries based on web browsing histories, which satisfies the following desired properties.

- *ξ-Privacy Preservation*: The randomized mechanism $\mathcal{M}$ has to achieve ξ-correlated differential privacy for any data owner at least.
- *Least Square Error*: Once the privacy constraint is reached, the data broker tries to find the optimal strategy matrix $\boldsymbol{G}$ to minimize the total error $\mathcal{E}_G(\boldsymbol{Q})$ when perturbing the correlated answer vector.
- *Budget Balance*: The charged price for the data consumer can afford all chosen data owners' privacy compensation, *i.e.*, $\sum_i \psi_i(\mathcal{Q}) \leq \pi(\mathcal{Q})$.
- *Truthfulness*: Any data owner i would never get a higher utility because of the untruthful bid $\tilde{c}_i$, *i.e.*, $u(c_i, \mathbf{c}_{-i}) \geq u(\tilde{c}_i, \mathbf{c}_{-i})$, where $\mathbf{c}_{-i}$ denotes the set of bid except c_i.
- *Individual Rationality*: Any data owner i has nonnegative utility for the truthful bid, *i.e.*, $u_i \geq 0$.

III. Design of *TERBE*

In this section, we present *TERBE* with aforementioned design objectives. *TERBE* includes three important components. *First*, *TERBE* performs data preprocessing to obtain true statistical results for interested ranges. *Second*, *TERBE* calculates the optimal strategy matrix G to minimize $\mathcal{E}_G(\boldsymbol{Q})$ with the privacy constraints, and then adopts the newly devised matrix mechanism to perturb correlated answer vector. *Finally*, *TERBE* quantifies each chosen data owner's privacy loss on multiple correlated queries, and then calculates the corresponding privacy compensation.

A. Data Preprocessing for Private Web Browsing History

The first component of *TERBE* is data preprocessing. Upon receiving the data consumer's request $\mathcal{Q}$, the data broker first finds all related data owners' purchasing records about some product (*e.g.*, iPhone), and then generates the original matrix [11]. Then the data broker extracts the key information X_i from each piece of web browsing history, and next obtains key information matrix $\boldsymbol{X}$. For example, the popular cashback website (*e.g.*, Ebates [13]) acquires any user's purchasing amount ν about any product e on the time t by sharing a fraction of profits with them. After that, the data broker counts statistical results $\boldsymbol{x}$ on disjoint ranges $\overline{\boldsymbol{R}}$, generates the query matrix $\boldsymbol{Q}$ based on the correlation between multiple queries, and finally obtains true answer vector $\boldsymbol{\zeta} = \boldsymbol{Q}\boldsymbol{x}$ on all queries.

B. Data Perturbation for True Answer Vector

Next, we consider the second component of *TERBE*, namely data perturbation mechanism for true answer vector. The data broker has to determine how to perturb the true answer vector in order to achieve ξ-correlated differential privacy at least and as small total error as possible simultaneously. Since the matrix mechanism can achieve a smaller total error theoretically by exploiting the correlation between multiple queries, the data broker determines to adopt the mechanism. Specifically, he first calculates an optimal strategy matrix $\boldsymbol{G}$ to minimize the total error $\mathcal{E}_G(\boldsymbol{Q})$, and then achieves the matrix mechanism in Definition 2.

However, to minimize the total error $\mathcal{E}_G(\boldsymbol{Q})$ with the privacy constraint and $\Delta_{\boldsymbol{G}} > 0$ is hard. Since there are two interactive parts in this objective function, *i.e.*, sensitivity term $\Delta_{\boldsymbol{G}}$ and linear combination of combinatorial term (*i.e.*, trace$(\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)$, latter abbreviated as trace term), the minimization problem is more complex especially when the scale (*i.e.*, the number m of total queries and size n of data vector $\boldsymbol{x}$) becomes larger.

Fortunately, we can exploit the semidefinite programming with rank constraint [14] to solve the above problem. The main idea is that we first try to minimize one term (*e.g.*, trace term) to calculate the optimal matrix $\boldsymbol{G}$ when the other term (*e.g.*, sensitivity term) is set to be less than 1. Next, we map the minimization problem into the semidefinite programming problem according to the Schur complement [15] in lemma 3.

The idea is reasonable because the objective function only relies on the combinatorial term, and there is probably not the unique strategy matrix $\boldsymbol{G}$. Hence, any of two terms can be scaled down within 1 under the initial goal. To further simplify the problem, we transfer the original query matrix as $\boldsymbol{Q} \in \mathbb{R}^{n*n}$, which can be achieved by matrix decomposition [16]. The consideration is reasonable because the strategy matrix $\boldsymbol{G}$ minimizes the total error $\mathcal{E}_G(\boldsymbol{Q})$ for any query matrix $\boldsymbol{Q}$, and also minimizes it for any other matrix D as long as $\boldsymbol{Q}^t\boldsymbol{Q} = \boldsymbol{D}^t\boldsymbol{D}$. Then the data broker can perturb the true answer vector $\boldsymbol{\zeta}$ based on the matrix mechanism once $\boldsymbol{G}$ is determined.

1) Original solution to calculate matrix $\boldsymbol{G}$: The key point for calculating the optimal strategy matrix $\boldsymbol{G}$ is to minimize the trace term while the sensitivity term is limited to $\Delta_{\boldsymbol{G}} \leq 1$. Minimizing the trace term is equivalent to minimizing the sum of the upper bound of each diagonal element $(\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)_{ii}$. The reason why we minimize the sum of upper bound values is that we try to solve the original minimization problem by typical semidefinite programming [14]. Next, we convert the minimization problem by constructing a positive semidefinite matrix M in Lemma 3.

Lemma 3. (***Schur Complement***). *Consider a matrix $M = \begin{pmatrix} Z & \mathcal{L} \\ \mathcal{L}^t & \mu \end{pmatrix}$, where Z is a positive semidefinite matrix, $\mathcal{L}$ is a n-dimensional vector and μ is a constant. Then we have M is positive semidefinite if and only if $\mu \geq \mathcal{L}^t Z^{-1} \mathcal{L}$ [15] [17].*

Let μ_i represent the upper bound of each diagonal element $(\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)_{ii}$. For any $\mu_i \geq (\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)_{ii}$, if we further set $(Z^{-1})_{m+i,m+i} = (\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)_{ii}$ and $\mathcal{L} = e_i$ which is a n-dimensional vector whose the i^{th} element is one and others are zero, we then construct a positive semidefinite matrix M_i based on Lemma 3. Besides, we define rank $\begin{pmatrix} I_m & \boldsymbol{G} \\ \boldsymbol{G}^t & P \end{pmatrix} = m$ to generate the matrix $P = (\boldsymbol{G}^t\boldsymbol{G})^{-1}$ as our rank constraint, and further limit $\Delta_{\boldsymbol{G}} \leq 1$. Therefore, minimizing the sum of μ_i is converted to a semidefinite programming problem with n semidefinite constraint matrices,

For any two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ are equivalent strategy matrix as long as $(\boldsymbol{A}^t\boldsymbol{A})^{-1} = (\boldsymbol{B}^t\boldsymbol{B})^{-1}$.

i.e., $M_i \succ 0$, which can be solved by the classic semidefinite programming with rank constraint [14].

Next, we give time complexity of the above problem. Because the semidefinite programming has time complexity $O(n^3)$ when there is only one constraint matrix M_i with size $n \times n$. There are n constraint matrices with size $m+n$ in total, and thus the problem has high time complexity $O(m^3 n^3)$, where m is usually large.

2) Simplified solution to calculate matrix $\boldsymbol{G}$: It is inefficient to solve the above optimization problem by original solution in a practical data trading market in consideration of high time complexity. Therefore, we try to simplify the above problem in order to reduce time complexity. The key principle is that we minimize the sensitivity term $\Delta_{\boldsymbol{G}}$ in turn while limiting each diagonal element to $(\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)_{ii} \leq 1$. To further simplify the minimization problem, we replace the l_1 norm sensitivity $\Delta_{\boldsymbol{G}}$ of $\boldsymbol{G}$ with the l_2 norm sensitivity $\Delta_{\boldsymbol{G}}^{'}$, where $\Delta_{\boldsymbol{G}}^{'} = \max_{||\boldsymbol{x}-\boldsymbol{x}^{'}||_2=1} ||\boldsymbol{G}\boldsymbol{x} - \boldsymbol{G}\boldsymbol{x}^{'}||_2 = \sqrt{\max_{1\leq i\leq n} |\lambda_i|}$ and λ_i is any eigenvalue of $\boldsymbol{G}^t\boldsymbol{G}$. Hence, minimizing the sensitivity term is exactly minimizing the maximum eigenvalue of $\boldsymbol{G}^t\boldsymbol{G}$. The reason why we use the l_2 norm sensitivity is that we try to regard the combinatorial term $W = (\boldsymbol{G}^t\boldsymbol{G})^{-1}$ as a whole, and output it by the converted semidefinite programming. Then we can exploit the fact that minimizing the maximum eigenvalue of $\boldsymbol{G}^t\boldsymbol{G}$ is equivalent to minimizing the sum of the upper bound of each diagonal element of $\boldsymbol{G}^t\boldsymbol{G}$, so as to simplify the objective function.

Next, we convert the above minimization problem by constructing a positive semidefinite matrix. Let $\mu_i^{'}$ denote the upper bound of each diagonal element of $\boldsymbol{G}^t\boldsymbol{G}$. For any $\mu_i^{'} \geq (\boldsymbol{G}^t\boldsymbol{G})_{ii}$, if we further set $Z^{'} = (\boldsymbol{G}^t\boldsymbol{G})^{-1}$, $\mathcal{L}^{'} = e_i$, we then construct a positive semidefinite matrix $M_i^{'}$ with size n based on lemma 3. Moreover, we set the rank constraint as rank $\begin{pmatrix} I_n & \boldsymbol{G} \\ \boldsymbol{G}^t & W^{(-1)} \end{pmatrix} = n$, and further limit $(\boldsymbol{Q}(\boldsymbol{G}^t\boldsymbol{G})^{-1}\boldsymbol{Q}^t)_{ii} \leq 1$. Hence, minimizing the sum of $\mu_i^{'}$ is converted to a semidefinite programming problem with n semidefinite constraint matrices, *i.e.*, $M_i^{'} \succ 0$, as our first subproblem. Clearly, the subproblem can be solved by the semidefinite programming with lower time complexity $O(n^3)$.

In addition, the other subproblem is to calculate the matrix $\boldsymbol{G}$ based on the previous output W in order to minimize its sensitivity, which can be also solved by the semidefinite programming with rank constraint [14] with low time complexity $O(n^3)$. After that, we determines $\boldsymbol{G}$.

Next, we analyze the connection between the original and simplifed solution in Lemma 4.

Lemma 4. *Given the query matrix $\boldsymbol{Q}$, we have*

$$\mathcal{E}_{G'}(\boldsymbol{Q}) \leq n\mathcal{E}_G(\boldsymbol{Q}), \tag{8}$$

where $\boldsymbol{G}$ and $\boldsymbol{G}^{'}$ is the solution of original and simplifed problem, respectively, and n is the dimension of matrix $\boldsymbol{G}$.

The inequality holds because of the basic property, *i.e.*, $||G||_1 \leq \sqrt{n}||G||_2$. From Lemma 4, we can observe the simplified solution actually produces a higher total error than the original one, but the difference is usually not large because of small size n. It is worth noting that we still have $\mathcal{E}_{G'}(\boldsymbol{Q}) \leq \mathcal{E}(\boldsymbol{Q})$ for a carefully-picked matrix $\boldsymbol{G}$ because of the large error. According to lemma 4, we can see the simplifed solution has a measurable objective function value compared with the original solution. However, the simplifed problem produces a slightly different privacy guarantee compared with ϵ-correlated differential privacy in terms of using l_2 norm sensitivity of $\Delta_{\boldsymbol{G}}^{'}$ instead, which is defined as follws.

Definition 4. **((ϵ, δ)-*Correlated Differential Privacy*).** *A randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-correlated differential privacy if for any two neighboring data vectors $\boldsymbol{x}, \boldsymbol{x}^{'}$, a query matrix $\boldsymbol{Q}$ representing the correlation between multiple queries, and any possible output vector $\boldsymbol{S}$, we have*

$$Pr(\mathcal{M}(\boldsymbol{Q}\boldsymbol{x}) = \boldsymbol{S}) \leq e^{\epsilon} \times Pr(\mathcal{M}(\boldsymbol{Q}\boldsymbol{x}^{'}) = \boldsymbol{S}) + \delta, \tag{9}$$

where $0 < \delta \leq 1$ is a constant set by the data broker, to guarantee ξ-correlated differential privacy at least, and we have to satisfy $\delta \leq 2e^{-\epsilon/8}$.

Therefore, the new privacy constraint is actually a relaxation of ϵ-correlated differential privacy, which further increases privacy budget. However, with a carefully-picked constant δ, we can still guarantees the maximum tolerant privacy budget ξ. Note that our simplified solution produces a higher total error than the optimal solution, and also has higher privacy loss than the traditional solution because of the increase of privacy budget. However, our approach still have large advantage in terms of high error of traditional solution and high time complexity of optimal solution.

To satisfy the above (ϵ, δ)-correlated differential privacy, we propose a newly devised matrix mechanism, *i.e.*, $\mathcal{M}_{\boldsymbol{G}}^{'}(\boldsymbol{Q}, \boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{G}^{+}\mathcal{M}^{'}(\boldsymbol{G}, \boldsymbol{x})$. Specificially, we still exploit the original matrix mechanism in Definition 2, but only add diverse random noises, *i.e.*, $\mathcal{M}^{'}(\boldsymbol{G}, \boldsymbol{x}) = \boldsymbol{G}\boldsymbol{x} + \frac{\Delta_{\boldsymbol{G}}^{'}}{\epsilon}\tilde{\boldsymbol{z}}_{\delta}$, where $\Delta_{\boldsymbol{G}}^{'}$ refers to l_2 norm sensitivity of $\boldsymbol{G}$, and $\tilde{\boldsymbol{z}}_{\delta} \in \mathbb{R}^m$ is comprised of m independent random noises drawn from a m-dimensional Gaussian distribution with mean $[0, \ldots, 0]^T$ and variance $[8ln(2/\delta), \ldots, 8ln(2/\delta)]^T$, where $\delta \leq 1$ and $\epsilon \leq 8ln(2/\delta)$.

Theorem 1. *The newly devised noise perturbation mechanism $\mathcal{M}^{'}(\boldsymbol{G}, \boldsymbol{x})$ satisfies (ϵ, δ)-correlated differential privacy.*

Proof. Please refer to our technical report [18] for the proof.

C. Privacy Compensation

Finally, we consider the third component of *TERBE*, *i.e.*, privacy compensation mechanism. Specifically, we first quantify each data owner's privacy loss, and then calculate affordable monetary compensation.

1) Privacy loss: Based on the above data perturbation mechanism $\mathcal{M}_{\boldsymbol{G}}^{'}(.)$, some data owners have to suffer privacy loss because of the data broker's leveraging their privacy information. Next, we first define each data owner's privacy

Algorithm 1: Correlated Query Trading Mechanism

Input: Procurement request $\mathcal{Q}$, set $\mathcal{N}$ of data owners, query matrix $\boldsymbol{Q}$, upper bound set of privacy loss $\boldsymbol{\eta} = \{\eta_1, \eta_2, \ldots, \eta_L\}$, set $\mathbf{c}$ of privacy costs, and budget B.

Output: Set $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|q|}\}$ of chosen data owners, payment vector $\mathbf{p}$, and perturbed answer vector $\boldsymbol{\zeta}^{'}$.

1 $\mathcal{C} \leftarrow \vec{\emptyset}$, $\mathbf{p} \leftarrow \mathbf{0}$;
2 // *Data Owner Selection* ;
3 Sort all data owners from $\mathcal{N}$ in the increasing order of c_i;
4 Find the largest index r such that $c_r \cdot \eta_r \leq \frac{B}{r}$;
5 **if** $r \geq |q|$ *and* $8(\frac{\Delta_{\mathcal{Q}}}{\epsilon})^2 ln(2/\delta) \leq v$ **then**
6 Choose the first $|q|$ data owners as winners;
7 **for** $i = 1$ *to* $|q|$ **do**
8 $\mathcal{C} \leftarrow \mathcal{C} \bigcup \{i\}$;
9 // *Privacy Compensation Mechanism* ;
10 $p_i = p_i + \min(\frac{B}{r}, c_{r+1} \cdot \eta_i)$;
11 **end**
12 // *Data Perturbation Mechanism* ;
13 $\boldsymbol{x}$ = dataPreprocessing($\mathcal{C}$);
14 $\mathcal{M}_{\boldsymbol{G}}^{'}(\boldsymbol{Q}, \boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{G}^{+}\mathcal{M}'(\boldsymbol{G}, \boldsymbol{x})$, $\boldsymbol{\zeta}^{'} = \mathcal{M}_{\boldsymbol{G}}^{'}(\boldsymbol{Q}, \boldsymbol{x})$;
15 **end**
16 **return** $(\mathcal{C}, \boldsymbol{p}, \boldsymbol{\zeta}^{'})$;

loss. For any two neighboring data vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{'}$ with or without any data owner's any piece of browsing history X_i, his privacy loss $\xi_i(\mathcal{M}_{\boldsymbol{G}}^{'})$ is given as follows.

Definition 5. *Any data owner's privacy loss by the randomized mechanism $\mathcal{M}_{\boldsymbol{G}}^{'}(.)$ over multiple correlated queries $\mathcal{Q}$ based on various web browsing histories is:*

$$\xi_i(\mathcal{M}_{\boldsymbol{G}}^{'}) = sup_{\boldsymbol{x}, \boldsymbol{S}} \left| \log \frac{Pr(\mathcal{M}_{\boldsymbol{G}}^{'}(\boldsymbol{Q}\boldsymbol{x}) = \boldsymbol{S})}{Pr(\mathcal{M}_{\boldsymbol{G}}^{'}(\boldsymbol{Q}\boldsymbol{x}^{'}) = \boldsymbol{S})} \right|. \tag{10}$$

The main idea of Definition 5 is to compare outputs by $\mathcal{M}_{\boldsymbol{G}}^{'}(.)$ over these two neighboring data vectors. Next, we further give the upper bound $\eta_i(\mathcal{M}_{\boldsymbol{G}}^{'})$ of any data owner's privacy loss. □

Theorem 2. *Let $\boldsymbol{G}$ be the optimal strategy matrix for the simplified problem, $\mathcal{M}_{\boldsymbol{G}}^{'}(.)$ denote the newly devised matrix mechanism, v represent the variance of added Gaussian noise, and δ be fixed privacy parameter. Then each data owner i's privacy loss is above bounded by*

$$\eta_i(\mathcal{M}_{\boldsymbol{G}}^{'}) = \frac{\underset{i \to k}{eig}(\boldsymbol{G}^t\boldsymbol{G}) + 4n \cdot \sqrt{\underset{i \to k}{eig}(\boldsymbol{G}^t\boldsymbol{G})} \sqrt{2ln(2/\delta)}}{2v} \tag{11}$$

Proof. Please refer to our technical report [18] for the proof. □

2) Monetary compensation: The monetary compensation for each data owner is calculated as the product between the upper bound of his privacy loss and received privacy cost. Next, inspired by the work [11], we introduce a practical privacy compensation mechanism in terms of a realistic scenario.

In a practical trading market, we consider data owners have diverse privacy costs, which are unknown to the data broker. Besides, conservative users have a high privacy cost, while liberal users are usually less concerned about their privacy and have a smaller privacy cost. Note that fixed privacy compensation probably leads to biased statistical results because most conservative users are probably unwilling to sell their private information in terms of unsatisfied privacy compensation. Hence, *TERBE* assumes that each data owner reports his privacy cost by auction, *i.e.*, $\boldsymbol{c} = \{c_1, c_2, \ldots, c_L\}$. Moreover, suppose that the data broker's budget for any data consumer's request $\mathcal{Q}$ is B. Combined with the data perturbation mechanism, we give an intuitive correlated query trading mechanism in Algorithm 1.

In consideration of limited budget B, he first picks data owners with affordable compensation costs in advance according to line 3-4. Besides, the matrix mechanism tries to minimize the sensitivity term, and thus outputs the strategy matrix $\boldsymbol{G}$ with a smaller sensitivity than $\boldsymbol{Q}$, *i.e.*, $\Delta_{\boldsymbol{G}} \leq \Delta_{\boldsymbol{Q}}$. Hence, if $8(\frac{\Delta_{\mathcal{Q}}}{\epsilon})^2 ln(2/\delta) \leq v$, then the data consumer's accuracy requirement is achieved. In line 7-11, if there are sufficient affordable data owners and the maximum tolerant variance v can be satisfied, then the data broker would choose the first $|q|$ data owners, and distribute the payment $\phi_i(\mathcal{Q}) = \min(\frac{B}{r}, c_{r+1} \cdot \eta_i)$ to each winner. Next, he leverages chosen data owners' records to generate true data vector $\boldsymbol{x}$ in line 13, which is achieved by the first component of *TERBE*, namely the function dataPreprocessing(.). Finally, the perturbed answer vector is returned by the proposed data perturbed mechanism $\mathcal{M}_{\boldsymbol{G}}^{'}$. We next show economic properties of *TERBE* in Theorem 3.

Theorem 3. *TERBE achieves individual rationality, truthfulness and budget balance.*

Proof. Please refer to our technical report [18] for the proof. □

IV. Evaluation

In this section, we present evaluation results of *TERBE* in consideration of the total error of data perturbation and diverse privacy compensation for each data owner.

A. Evaluation Settings

1) Dataset: We first introduce a real-world dataset from an open dataset community [19]. There are over 1000 vendors who have 14946 purchasing records about more than 1000 commodity goods. Each purchasing record includes purchasing date and amount, which exactly indicates that some vendor spends money ν for buying some commodity e on the date t. Because there are no available open personal purchasing records including individuals' private information on the Internet, we generate personal users' purchasing records based on the above dataset. Specifically, we replace vendors as personal web users, and generate their attributes a like 'age', 'gender' and 'income'. Besides, we simulate each data consumer's multiple correlated queries $\mathcal{Q}$ like the toy example in Fig. 2, where more ranges $\overline{\boldsymbol{R}}$ are refined for more queries

The budget can be calculated by the arbitrage-free pricing of query by existing works [4] [6].

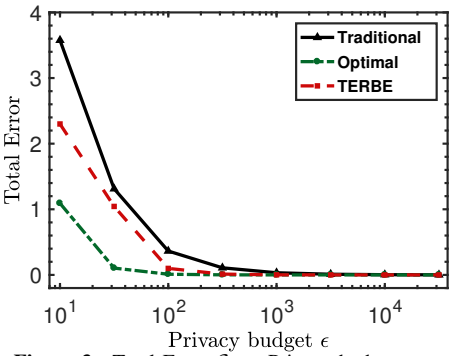

Figure 3. Total Error $\mathcal{E}$ vs. Privacy budget ϵ.

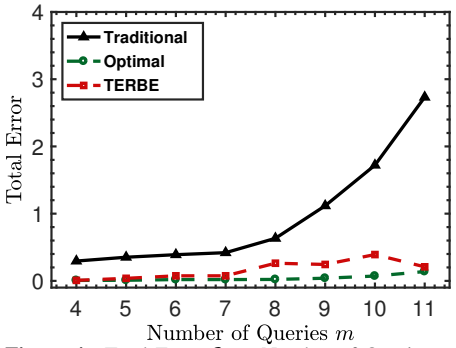

Figure 4. Total Error $\mathcal{E}$ vs. Number of Queries m.

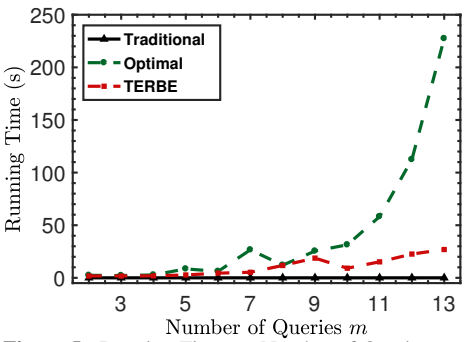

Figure 5. Running Time vs. Number of Queries m.

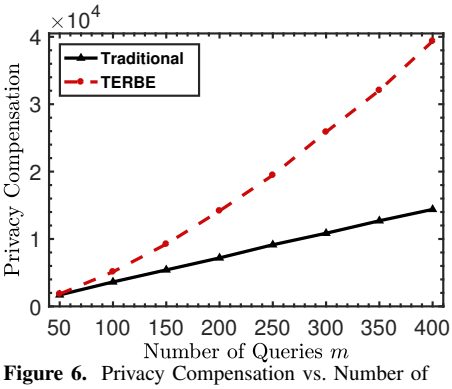

Figure 6. Privacy Compensation vs. Number of Queries m.

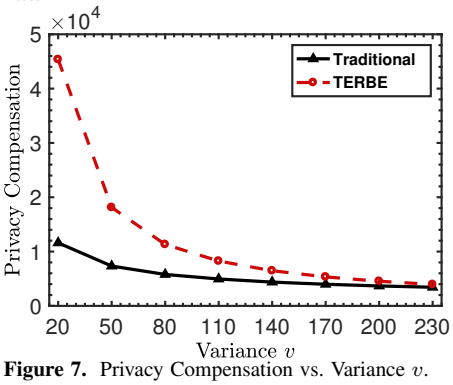

Figure 7. Privacy Compensation vs. Variance v.

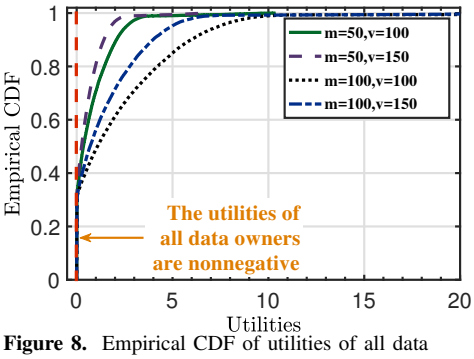

Figure 8. Empirical CDF of utilities of all data owners.

m. For example, we count the number of users who have over 5000$ average spending about consult service in 2014 for each range like 25-30 about 'age'. Finally, we obtain the true data vector $\boldsymbol{x}$ for $\mathcal{Q}$.

2) Settings: Supposed that data owners' privacy costs are drawn from the exponential distribution with mean 0.5. Let $B = 10^6$ and $\xi = 1.2\epsilon$. In addition, we vary privacy budget ϵ within $[10^1, 10^{1.5}, \ldots, 10^5]$, and set the parameter $\delta = 0.5$ for *TERBE* in order to satisfy ξ-correlated differential privacy. By default, we fix the number of correlated queries and privacy budget as $m = 5$ and $\epsilon = 100$, respectively. Moreover, we evaluate the performance of *TERBE*, and compare it with traditional and optimal mechanism, respectively. Note that optimal mechanism refers to the matrix mechanism by original solution to calculate the optimal matrix $\boldsymbol{G}$. The metrics include total error, running time and privacy compensation which refers to total payments of all chosen data owners (*i.e.*, $\sum_{i \in \mathcal{C}} p_i$). For privacy compensation, we only present evaluation results for *TERBE* and traditional mechanism, because optimal mechanism adds the same Laplace noise with traditional mechanism basically with the same compensation. In addition, each data point is the average value after running 200 iterates.

B. Evaluation Results

First, we evaluate the performance of the data perturbation mechanism for *TERBE*.

1) Evaluation of total error: From Fig. 3, it can be seen that the total error decreases when privacy budget ϵ increases from 10^1 to 10^5. This is because a higher privacy budget means a smaller added noise, and thus leads to a smaller total error. Besides, *TERBE* outperforms than the traditional mechanism, but is inferior to the optimal mechanism, which conforms to our expectation.

In Fig. 4, we can observe *TERBE* decreases 66.67% of the total error than the traditional mechanism when $m = 8$, and even 90% of the total error for more queries, while *TERBE* is close to that of the optimal mechanism with a small gap. The reason lies in the fact that the traditional mechanism never leverages correlations between multiple queries. Moreover, the accumulated error would be larger with the increase of the number of queries in terms of an added Laplace noise for each query. However, the other two mechanisms exploit correlations to produce a smaller total error.

2) Evaluation of running time: Fig. 5 shows *TERBE* decreases 92% of running time than the optimal mechanism when the number of queries increases to $m = 13$, which further verifies the optimal mechanism cannot be applied to a realistic scenario in terms of a large number of queries. Fortunately, it takes the data broker acceptable time to achieve data perturbation with much lower running time than the optimal mechanism. It can be observed that the traditional mechanism executes quickly because it never needs to calculate the optimal matrix $\boldsymbol{G}$, which usually takes a long time.

The above evaluation results show *TERBE* indeed balances total error and user privacy well within acceptable running time. Next, we turn to the privacy compensation mechanism.

3) Evaluation of privacy compensation: Supposed that the traditional mechanism adopts similar privacy compensation

with *TERBE*. The only difference is that the upper bound of any data owner i's privacy loss for the traditional mechanism is $\eta_i(\mathcal{M}(\boldsymbol{Q}, \boldsymbol{x})) = \frac{\sum_{j \in m} |q_{kj}|}{\sqrt{0.5v}}$ [4], where the subscript k indicates the k^{th} data element x_k this data owner i's browsing history is counted to. In Fig. 6, we can find that *TERBE* pays more privacy compensation than the traditional mechanism when the number of queries increases from 50 to 400. The reason is that *TERBE* generates perturbed answer vector with smaller total error than traditional mechanism but at the cost of a higher monetary compensation within an affordable budget.

Fig. 7 demonstrates privacy compensation for both mechanisms goes down with the increase of variance from 20 to 230. It is clear that a lower variance means the data consumer's higher accuracy requirement for the same number $m = 100$ of correlated queries. Consequently, smaller noises are added, and probably cause higher privacy loss for each chosen data owner. Therefore, higher privacy compensation should be distributed to them.

According to Theorem 2, we can see that each data owner have a diverse upper bound of privacy loss, and thus they would be paid for diverse privacy compensation. Fig. 8 exactly depicts the percentage of their utility under 4 kinds of different settings. Specifically, it can be observed that there would be a larger percentage of data owners who obtain higher utility for a larger number $m = 100$ of queries and the lower variance $v = 100$. It is reasonable because more data owners suffer higher privacy loss for more refined ranges and a higher accuracy requirement, and thus they have to be compensated more. Besides, from Fig. 8, we can see all data owners have nonnegative utilities, which further verifies *TERBE* achieves the property of individual rationality.

The above evaluation results illustrate *TERBE* pays more privacy compensation than traditional mechanism but still guarantees budget balance.

V. Related Work

A. Data Market Design

A growing number of pieces of related literature have focus on data market design in recent years. Research work from database field first study arbitrage-free pricing of queries over common user relational database [20], [21]. Li *et al.* [9] then propose the matrix mechanism to minimize the error of returned perturbed queries. Follow-up work by Li *et al.* [5] further considers the pricing of a single linear query by achieving arbitrage-freeness, as well as a privacy compensation mechanism for data owners with diverse privacy strategies. Based on Li *et al.*'s work, Niu *et al.* [4] next propose a query trading mechanism for common aggregate statistics especially in terms of correlations between diverse individuals. Different from above work, Niu *et al.* [6] and Jin *et al.* [7] aim at trading real-world datasets, *i.e.*, personal users' time series data and sensing workers' location privacy, respectively. Specifically, Niu *et al.* [6] borrows the idea of pufferfish privacy [22] to quantify each data owner's privacy loss at temporal correlation. Besides, Jin *et al.* [7] design a location obfuscation mechanism to protect sensing workers' location privacy, and compensate them for both sensing cost and privacy cost. Similarly, Zhang *et al.* [12] propose a privacy-preserving outsourcing mechanism for social media data like Twitter data. Based on Zhang *et al.*'s work, Cai *et al.* [11] further design the trading mechanism for web browsing history especially in terms of data consumer's utility, and try to trade the whole perturbed dataset.

However, none of the above work has taken the trading of multiple correlated queries into consideration, and further considered privacy compensation mechanism for data owners with diverse privacy losses.

B. Incentive Mechanism for Trading

Other previous work investigate incentive mechanism design for data trading so as to motivate data owners to report their privacy valuation truthfully. Ghosh *et al.* [23] regard user privacy as a commodity, and trade each counting query by running auction. Wang *et al.* [24] assume that each data owner reports a noisy data version in terms of an untrusted data collector, and obtains privacy compensation in the context of game theory.

Nevertheless, these work aim at trading private data by a game-theoretic model, rather than the pricing of data privacy as our key idea.

VI. Conclusions

In this paper, we have proposed a privacy-preserving framework *TERBE* for trading multiple correlated queries based on web browsing histories. In *TERBE*, data owners have to report their real privacy costs, and then can get reasonable privacy compensation for their diverse privacy losses in a satisfying manner. Besides, each data consumer can purchase interested multiple correlated queries with a comparable total error with the optimal mechanism. We have evaluated the performance of *TERBE* through real-data based experiments. Evaluations and analysis demonstrate *TERBE* achieves a satisfying trade-off between user privacy and the total error within acceptable running time, and guarantees all desired economic properties, which shows the usefulness and feasibility of *TERBE*.

VII. Acknowledgment

This research is supported in part by the 2030 National Key AI Program of China 2018AAA0100503 (2018AAA0100500), National Science Foundation of China (No. 61772341, No. 61472254, No. 61772338 and No. 61672240), Shanghai Municipal Science and Technology Commission (No. 18511103002, No. 19510760500, and No. 19511101500), the Innovation and Entrepreneurship Foundation for oversea high-level talents of Shenzhen (No. KQJSCX20180329191021388), the Program for Changjiang Young Scholars in University of China, the Program for China Top Young Talents, the Program for Shanghai Top Young Talents, Shanghai Engineering Research Center of Digital Education Equipment, and SJTU Global Strategic Partnership Fund (2019 SJTU-HKUST).

REFERENCES

[1] G. Beigi, R. Guo, A. Nou, Y. Zhang, and H. Liu, "Protecting user privacy: An approach for untraceable web browsing history and unambiguous user profiles," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, 2019, pp. 213–221.

[2] "Cnn markets," "https://money.cnn.com/data/markets/", 1980.

[3] "Behavior targeting," "https://business.twitter.com/en/targeting.html.", 2017.

[4] C. Niu, Z. Zheng, F. Wu, S. Tang, X. Gao, and G. Chen, "Unlocking the value of privacy: Trading aggregate statistics over private correlated data," in *KDD 2018, London, UK, August 19-23*, 2018, pp. 2031–2040.

[5] C. Li, D. Y. Li, G. Miklau, and D. Suciu, "A theory of pricing private data," *Commun. ACM*, vol. 60, no. 12, pp. 79–86, 2017.

[6] C. Niu, Z. Zheng, S. Tang, X. Gao, and F. Wu, "Making big money from small sensors: Trading time-series data under pufferfish privacy," in *2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019*, 2019, pp. 568–576.

[7] W. Jin, M. Xiao, M. Li, and L. Guo, "If you do not care about it, sell it: Trading location privacy in mobile crowd sensing," in *2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019*, 2019, pp. 1045–1053.

[8] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, no. 3-4, pp. 211–407, 2014.

[9] C. Li, G. Miklau, M. Hay, A. McGregor, and V. Rastogi, "The matrix mechanism: optimizing linear counting queries under differential privacy," *VLDB J.*, no. 6, pp. 757–781, 2015.

[10] "Axicom," "https://www.acxiom.com", 1969.

[11] H. Cai, F. Ye, Y. Yang, Y. Zhu, and J. Li, "Towards privacy-preserving data trading for web browsing history," in *Proceedings of the International Symposium on Quality of Service, IWQoS 2019, Phoenix, AZ, USA, June 24-25, 2019.*, 2019, pp. 25:1–25:10.

[12] J. Zhang, J. Sun, R. Zhang, Y. Zhang, and X. Hu, "Privacy-preserving social media data outsourcing," in *INFOCOM 2018, Honolulu, HI, USA, April 16-19, 2018*, 2018, pp. 1106–1114.

[13] "Ebates," "https://www.ebates.com", 1998.

[14] J. Dattorro, *Convex optimization & Euclidean distance geometry*, 2010.

[15] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.

[16] D. M. Witten, R. Tibshirani, and T. Hastie, "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis," *Biostatistics*, vol. 10, no. 3, pp. 515–534, 2009.

[17] P. A. Parrilo and S. Lall, "Semidefinite programming relaxations and algebraic optimization in control," *Eur. J. Control*, vol. 9, no. 2-3, pp. 307–321, 2003.

[18] "Technical Report for TERBE," "https://www.dropbox.com/s/sog68tfdqa0ckkc/Technical_Report_for_TERBE.pdf?dl=0", 2019.

[19] "dataworld," "https://data.world/finance/dc-purchase-orders-2014", 2015.

[20] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu, "Toward practical query pricing with querymarket," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, NY, USA, June 22-27*, 2013, pp. 613–624.

[21] P. Koutris, P. Upadhyaya, M. Balazinska, and B. Howe, "Query-based data pricing," in *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, 2012, pp. 167–178.

[22] S. Song, Y. Wang, and K. Chaudhuri, "Pufferfish privacy mechanisms for correlated data," in *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, 2017, pp. 1291–1306.

[23] A. Ghosh and A. Roth, "Selling privacy at auction," *Games and Economic Behavior*, vol. 91, pp. 334–346, 2015.

[24] W. Wang, L. Ying, and J. Zhang, "The value of privacy: Strategic data subjects, incentive mechanisms and fundamental limits," in *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science, Antibes Juan-Les-Pins, France, June 14-18*, 2016, pp. 249–260.