Differentially Private Aggregate Statistics in Contact Tracing

Daniel Kuang (dk6548) Sabrina Tseng (stseng) Xu Zeng (xuzeng) Patrick Zhang (patz)

***Abstract*—The most recent pandemic has driven society into mass social quarantine, a tactic that has been practiced and proven successful in previous pandemics. But, unlike previous pandemics such as the last pandemic over 90 years ago (the Spanish Flu), we now have contact tracing apps that can hasten tracing of recent interactions of infected individuals. By tracing new potential infected individuals, the government can monitor the spread of a disease and ideally keep it contained before it spreads further.**

However, tracing apps have sparked concerns over their system architecture, especially whether privacy will be jeopardized or whether personal data are ever misused. In this paper, we will provide a short overview of two types of system architectures for contact tracing: the centralized model and the decentralized model. We will introduce differential privacy and how it can provide aggregate statistics to health authorities for effective decision making without compromising individual privacy. We will conclude with three types of differentially private statistical techniques and how they can be applied under this setting.

***Index Terms*—Contact tracing, differential privacy, distributed model, k-clustering, regression analyses, histogram**

I. Introduction

An interconnected world with efficient transportation and a virus with high infectivity, low casualty rate and asymptomatic spread is a recipe for a pandemic. By having high infectivity and a asymptomatic spread, a virus can expand its global influence inconspicuously, and by the time it is recognized after wreaking havoc on vulnerable populations, it becomes very hard to control and will likely become a pandemic.

The above is a short description of how Severe Acute Respiratory Syndrome Coronavirus 2 or COVID-19 was declared a pandemic, and the above description will likely be similar for how future pandemics will emerge. The reason is a virus with high casualty rates will be recognized immediately and contained. Likewise, a virus with low infectivity will spread much slowly and stall enough time for it to be labeled as a threat and contained.

Fortunately, unlike previous pandemics, our society today is technologically advanced and highly interconnected; smartphones are ubiquitous and people can communicate over wide networks. This allows for contact tracing apps that can work along with social quarantine to curb a virus's spread. The importance of curbing a virus's spread is threefold:

1) It buys time for other countries or regions to enact isolation policies
2) It alleviates burden on hospitals and their resources so they can more effectively treat victims
3) It gives time for researchers to study the virus and develop vaccines. With contact tracing, authorities can identify local outbreaks and contain them before carriers unknowingly disseminate the virus further

Individuals can also identify whether they are at risk of being a carrier and responsibly isolate themselves before they put their loved ones at risk. Thus, it stands to reason that one would use the contact tracing app, but issues on privacy remain a barrier for full participation. In particular, certain architectures hinge on trust. Aggregate statistics drawn from curated private data do not guarantee individual privacy if adversaries combine data from the server with auxiliary information to narrow down an individual.

However, widespread participation is crucial for the effectiveness of contact tracing apps. The more people participate, the more likely it is that an infected individual can be identified through their contacts and notified early of possible exposure, allowing them to quarantine and avoid infecting others to slow the spread. In addition, widespread participation leads to more accurate data on how and where the virus is spreading, allowing authorities to make better policy decisions. Thus, it is important to resolve the trust and privacy issues to allow contact tracing apps to be more effective.

In this paper we propose methods for collecting and analyzing aggregate data from contact tracing apps while maintaining differential privacy. In particular, we will focus on the centralized architecture and on three types of statistical analysis: histograms, regression, and clustering.

The following sections are structured as follows. Section II will summarize the main architectures of contact tracing apps and introduce how differential privacy can resolve privacy leakages from aggregate statistics. Section III discusses related work in differential privacy. In Section IV we clarify our threat models, and in Section V we detail methods for doing statistical analysis under differential privacy. Discussion and future work is in Section VI, and finally we conclude in Section VII.

II. Preliminaries

A. Contact Tracing

Contact tracing apps help automate the contact tracing process by leveraging GPS and/or Bluetooth capabilities of smartphones to identify individuals who have been in close proximity. In addition, they also automate the aggregation of valuable data that health authorities can use to inform decisions. However, a major challenge in building such systems is security and privacy, since they collect large amounts of data about an individual's location, activity, and interactions with others. Several architectures have been proposed to address these challenges, including BlueTrace [1] and PACT [2]. The survey by Nadeem et al. [3] classifies these into three main architecture types: centralized, decentralized, and hybrid [3]. Since hybrid is a mix of centralized and decentralized, we will focus only on reviewing centralized and decentralized before introducing differential privacy.

1) Centralized: The centralized architecture works as follows:

1) A central tracing app server generates a TempID for each device.

2) The central server encrypts the TempID, with a secret key only known to itself, and sends the encrypted TempID to the device.

3) When two devices encounter each other, they exchange TempIDs over Bluetooth, along with other information such as a timestamp and signal strength (for determining proximity).

4) If an individual tests positive, they can choose to upload their encounters to the central server. The central server can map the encounter data to specific individuals, perform risk analysis and other data processing, and decide if any other devices should be notified.

A key feature of the centralized architecture is that it relies on a trusted central server, which is responsible for storing and managing PII, encounters, and other sensitive information.

2) Decentralized: On the other hand, the decentralized architecture works as follows:

1) Devices generate their own random seeds, generating a new seed every hour.

2) The seed and the current time are used as inputs to a pseudorandom function (PRF) to generate a 'chirp.' Note that this happens on the device itself, not on a central server.

3) Each device broadcasts its chirps via Bluetooth. A phone that receives a chirp stores its value and other information such as the timestamp and the signal strength.

4) If a user tests positive, they can choose to upload their seeds and the corresponding times to a server. Other users can download seeds from the server, and locally reconstruct the chirps and compare their values to the values stored on their phone. Thus, the risk analysis computation happens locally on each user's device.

In contrast with the centralized architecture, the decentralized architecture does not require the central server to store any sensitive data. This makes it a good choice when there is no entity that people trust to keep their information private. A disadvantage of the decentralized architecture is that it may be harder to collect aggregate data.

B. Differential Privacy

Differential Privacy is a method analyzing the privacy of a protocol or algorithm in a quantifiable manner. The intuition behind differential privacy is bounding how much information can be leaked about any single individual from a certain query. More specifically, it acts as a numerical constraint on aggregate statistics from databases. An algorithm is considered differentially private if an adversary cannot determine if a certain individual's information is used. In a tabular database setting, this is synonymous to a single row in a table.

We call two databases that differ by a single piece of information as neighboring databases. In a tabular data setting, this means a single row (as described above), but there can be other settings with other notions of neighboring database. For example, neighboring graph databases can differ either by a node or an edge.

The quantifiable notion of differential privacy is known as ϵ-differential privacy. A randomized algorithm A achieves ϵ differential privacy if

$$Pr[A(D_1) \in S] \leq exp(\epsilon) * Pr[A(D_2) \in S]$$

where D_1 and D_2 or any two neighboring databases and S is set of all possible results. Thus, the higher the value of $epsilon$, the less private, and vice versa. There is also a tradeoff with the accuracy of the result, which decreases as privacy increases. Systems tend to have a privacy budget, a maximum ϵ value, over many queries, and differential privacy allows systems to determine how much noise to add to data/results.

Differential privacy is not a cryptographic technology for achieving privacy but rather a property of any randomized algorithm that provides mathematical guarantees on privacy. Differential privacy guarantees that information about any single person can not be leaked even with external information on the dataset along with the queries.

C. Notions of Differential Privacy

1) Global vs Local Differential Privacy: In a global differential privacy model, we assume there exists a trusted data curator. The trusted data curator will store sensitive data and for any query to the curator, it will return a differentially private answer. In particular, the curator will add noise to the query results to serve as a privacy barrier between the data curator and the authorities behind a query. The additive noise or perturbation to the aggregate function guarantees that membership participation is indistinguishable from non-membership participation. The assumptions in this setting are that 1) the server is a secure link in the system and 2) the server will perform the necessary perturbation on the aggregate function to produce a differentially private answer.

However, the motivation for using differential privacy is to remove trust in the first place. Hence, the first risk behind a global differential privacy model is the data curator. If the individuals behind a data curator are adversarial, then the benefits behind a global differential privacy model is void since they can easily access the sensitive data stored in the data curator.

In a local differential privacy model, data owners add noise to their sensitive information before sending to the data curator. This way, the assumption that the data curator is trustworthy is no longer necessary since the data curator is now containing differentially private information. When a query arrives, the data curator will return query results with additional noise or perturbation added.

Although all sources except the data owners now contain differentially private information, query results will be less accurate than the query results from global differential privacy models. The reason is the additional amount of noise added on each step outside from data owners to ensure differential privacy. In the global differential privacy model, noise is only added in query results. But, in local differential privacy models, data curators also have noise added from each data owner's sensitive information.

As a result, models that implement local differential privacy are mainly for aggregate query results, not finer results. For example, one can query maximum or minimum and expect reasonable accuracy.

2) Differential Privacy on Types of Databases: The notion of differential privacy is based on the idea of neighboring databases. However, different database structures have a different definition of neighbors.

1) Tabular Data - The most common database structure is RDBMS (relational database management system). Data are stored in tables with each row providing data specified by the schema. In this setting, statistics tend to be aggregated on some combination of data between the tables in a tabular format. A neighboring database is defined as when either a single row (user) or a certain column value in a row is different or nonexistent. In most cases, these are equivalent. [4]

2) Graphs - Graph databases have become more popular and algorithms for differentially private graph statistics (such as triangle counting or degree distribution) have been created. In this setting, a neighboring database is defined as either the inclusion or exclusion of an entire node or edge.

3) Documents - Document based databases have also become popular as they are easier to scale with less defined structure. While there may be specific notions of differential privacy in certain structures of documents (e.g. JSON), a general notion of differential privacy for documents is using a "bag-of-words" approach is popular in machine learning to transform text into numerical data. [5]

III. Prior Works

Related works in the field of differential privacy include statistical biomedical analysis, geolocation dataset analysis and the U.S. Census Bureau.

In the biomedical field, Damson [6] has emerged as a strong differential privacy system that optimizes the tradeoff between high accuracy and low privacy costs. Damson allows for strong privacy guarantees on common biomedical analysis tasks, including histograms, data cubes, and other broad statistical learning algorithms. It accomplishes these through query optimization, which optimize the accuracy of the analysis results under constraints of privacy budget usage. The optimization operates under two techniques: Batch Query Processing and Relative Error Minimization. In Batch Query Processing, a query is processed as different sets of queries and their results are combined to answer the original query. Under Relative Error Minimization, Damson minimizes relative error instead of absolute error of queries since the utility of small results are more susceptible to added noise.

Another application of differential privacy is Microsoft's PrivTree system [7]. The PrivTree system incorporates differential privacy to prevent accurate reconstruction of individual's geolocation from their dataset. PrivTree uses a data manipulation technique that pre-processes the geolocation data by first partitioning the location data into sub-regions based on data point density and then applying location perturbation to guarantee privacy while maintaining statistical accuracy. This system can implement differentiallly private algorithms on geolocation data usinf Laplacian noise and partitions.

Recently, the U.S Census Bureau has proposed investigating differential privacy for their datasets. Achieving desired privacy guarantees on the Census data would involve building algorithms to prevent reconstruction of individual data from aggregate statistics. The goal is to apply differential privacy in such a way that the confidentiality protections of their system will not be compromised under complete or additional outside information.

Our work is similar to previous works in that we are analyzing the confidentiality of digital contact tracing through the lens of differential privacy. We apply such techniques in the setting of COVID-contact tracing to reduce the required level of trust in existing parties of the protocol.

IV. Threat Models

In this section, we will outline our proposed threat model. Our threat model falls under a centralized architecture. There exists a central party that controls the contact-tracing app, data from other parties, and communication between parties. This central party is trusted in storing sensitive data such as sex, age, weight, etc. This party is also trusted in answering queries on such data for diagnostic and analytical purposes.

A. Database Architecture and Differential Privacy

We considered two main database architectures, tabular and graph-based. Tabular data would store individuals' sensitive information such as contacts between individuals in rows. For a graph, each individual is a node which holds sensitive information, and contacts are represented as edges.

For the aggregate statistics that we want to compute, tabular data in a RDBMS is the most applicable. The statistics we generally want to compute are on sensitive information of the individuals rather than the actual contacts between individuals. This setting also has the most work done on differentially private statistics.

Due to the centralized nature of our model, we will also be applying global differential privacy while non-perturbed data is stored within a central database for queries.

B. Parties

The parties that participate in our centralized contact tracing model are [3]:

1) *Users*: Individuals who are using contact tracing apps are assumed to be honest but curious. They are trusted to only upload real contact data and diagnosis, but may try to determine the status of other individuals in the network.

2) *Tracing App Server/Database*: The tracing app server holds the database of individuals and is completely trusted with the sensitive information of individuals.

3) *Health Authority*: We consider the health authority to be honest but curious. They aim to authorize data properly, but may try to learn information about certain individuals from the tracing app server.

4) *Hospitals*: Hospitals only play a role in sending diagnoses to individuals or the health authority and are completely trusted to diagnose correctly.

The main difference between our threat model and prior centralized contact tracing threat models is the level of trust in the health authority. We reduce the level of trust required in the health authority; we consider the possibility of the health authority trying to learn information of individuals and aim to remove that possibility.

V. Data Analysis under Differential Privacy

As mentioned in Section IV, contact tracing apps might require or allow users to provide demographic information when they register, such as sex, age, weight, and zip code. In this section we will introduce ways in which the central server can analyze this data and publish aggregate statistics while maintaining differential privacy.

A. Histograms

Health authorities, as well as the general public, might be interested in analyzing the distributions of certain demographic variables, like age, over the space of people who have tested positive. This can help them can identify patterns about how the virus spreads to make more informed decisions and recommendations. Thus, it would be useful to be able to publish differentially private histograms over this data, without leaking private information.

Histograms provide a summarized representation of a distribution by separating the data into bins, which represent ranges of values, and counting the number of data points that fall into each bin. The *structure* of a histogram refers to the choice of bins and what value ranges they represent. Typically, during histogram construction, the structure is optimized for a given number of bins in order to minimize the sum of squared error (SSE) on unit-length count queries.

The definition of a differentially private histogram is that adding or removing a single data point has only a negligible effect on the output histogram, such that an adversary cannot determine whether or not a certain data point is in the set. Formally, a histogram publication mechanism Q satisfies ϵ-differential privacy (ϵ-DP) if it outputs a randomized histogram H such that

$$\forall D, D' H : Pr(Q(D) = H) \leq e^{\epsilon} \cdot Pr(Q(D') = H)$$

for neighboring datasets D and D', i.e. they differ by only one data point [8]. An important note is that the structure of the histogram can also leak information, since adding or removing a data point can cause the optimal structure to change [9]. Thus, there are two general strategies for publishing differentially private histograms:

1) Add noise first to the raw data, then construct an optimized histogram on the result.

2) Construct the histogram from the raw data, then add noise to the transformed data (both the counts and the structure).

For our application, we will focus on strategy 1 because it produces better accuracy for short-range queries, which is

more important for analyzing the shape of the data distribution [9]. We will apply the NoiseFirst method by Xu et al [9].

First, we compute a histogram on the original data with unit-length bins, then add noise using the Laplace Mechanism (LM) [10], which generates noise from a Laplacian distribution. The result from the paper shows that it is sufficient to add noise of magnitude $1/\epsilon$, i.e. $Lap(\frac{1}{\epsilon})$, in order to achieve ϵ-DP. This results in a noisy sequence of counts, $\hat{D} = \{\hat{x_1}, \hat{x_2}, ...\hat{x_n}\}$. Then, we optimize the histogram structure based on $\hat{D}$; this can be done using dynamic programming as proposed by Jagadish et al [11]. Note that we are optimizing the structure based on the noisy counts, so this will not leak any information. Furthermore, because the Laplace noise is centered about 0, when we merge bins in the optimization process the noise is smoothed out, leading to better accuracy than standard LM. This algorithm provides us with an ϵ-differentially private view of the desired histogram.

In addition, Xu et al. [9] also suggest optimizing the value of k, the number of bins in the histogram, by trying all possible values from 1 to n (the range of the data) and choosing the one with the lowest SSE. This can further improve the accuracy of short-range queries to the histogram. Because this computation might take a long time to run, there is a tradeoff between accuracy and having up-to-date data. This tradeoff can be tuned by testing fewer values of k, allowing the histograms to be updated more frequently if needed.

B. Regression Analyses

Another possible analysis whose results are particularly useful is regression analysis. With regression analyses, health authorities can deduce correlations between attributes such as age and mortality rate.

Unfortunately, performing regression analyses and satisfying differential privacy is non-trivial, as regression analysis implies minimizing noise to reach an optimal solution, which is at odds with injecting noise for differential privacy. Hence, one of the challenges is deciding on the minimum amount of noise for an approximately optimal result that satisfies differential privacy.

For example, existing works on differentially private regression analysis include:

1) Injecting noise into regression results

2) Analyzing synthetic points generated from the distribution of original data points

But these methodologies produce inaccurate regression results because the amount of noise injected is often significant enough to skew a minimization from its actual optimum.

For results that remain differentially private and as accurate as possible, we will be using the Functional Mechanism (FM). We will provide a general overview of FM below.

Functional Mechanism is a differentially private framework for optimization analyses proposed in Zhang et al [12]. It assumes our database D satisfies the following for each tuple $t_i = (x_{i1}, x_{i2}, x_{i3}, ..., x_{i(d-1)}, x_{id}, y_i)$:

$$\sqrt{\sum_{l=1}^{l=d} x_{il}^2} \leq 1$$

$$y_i \in [-1, 1]$$

If tuples do not satisfy the above assumption, we can transform or scale our data before regression analysis.

FM relies on perturbing the objective function instead of the results or input data. However, perturbing the objective function has two issues:

1) One cannot inject noise into a function trivially and expect a differentially private accurate result

2) Noise injections can yield unbounded objective functions or remove minimums

To resolve the first issue, Zhang et al [12]. exploits the Laplace Mechanism (LM) and the Stone-Weierstrass Theorem. The former is a differentially private framework that adds Laplace noise to any query output provided the output is a real number. The latter provides a different polynomial representation of the objective function to be minimized. FM essentially converts an objective function f into its polynomial representation $\bar{f}$ with Stone-Weierstrass Theorem and adds Laplace noise using LM to each of $\bar{f}$'s coefficients. The amount of noise injected is formulated as

$$\text{Lap}\left(\frac{2(d+1)^2}{\epsilon}\right)$$

where Lap(x) is a random value drawn from a Laplace distribution with zero mean and a predetermined scale s. ϵ is the privacy budget used in defining differential privacy between two neighbor datasets.

To resolve the second issue, Zhang et al [12]. propose either rerunning FM until a bounded objective function is made or applying regularization or spectral trimming. Proofs and additional information on FM are included in Zhang et al [12].

C. Clustering

Health authorities, and the general public, might also want to query the central database for aggregate statistics such as the relative risks of COVID transmission among a certain age group. Under this setting, the application of a differentially private clustering algorithms will be useful in providing that statistic and will maintain privacy guarantees on the publication of such statistics.

Clustering aims to form clusters of data points under some notion of similarity within clusters and dissimilarity among clusters. Each cluster should contain data points that are more similar to each other than those in other groups. A simple but powerful data analytic clustering algorithm is k-means

clustering, a popular unsupervised machine learning algorithm. Specifically, k-means clustering applies k partitions to a set of n data points, forming k clusters with the cluster centroid serving as the cluster's classification.

More formally, k-means clustering aims to minimize the squared Euclidean distance between a cluster's centroid and all points within that cluster, which by extension minimizes the within-cluster variance.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d}((x_i)^k - (x_j)^k)^2}$$

Given n data examples $(x_1, x_2, x_3, ..., x_n)$, k-means clustering partitions n data points into k cluster sets $C = \{C_1, C_2, ..., C_k\}$ satisfying the objective:

$$\arg\min_{\mu} \arg\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2$$

where μ_i is the mean of cluster C_i

The k-means algorithm operates under iterative refinement, where the algorithm alternates between two steps to find a convergence solution. The algorithm process is as follows:

1) Fix μ, assign each example point to the cluster with the least squared Euclidean distance and optimize on C:

$$\arg\min_{C} \sum_{x \in C_i} |x - \mu_{x_i}|^2$$

2) Fix C, recalculate the centroid for each cluster and optimize on μ:

$$\arg\min_{\mu} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2$$

Calculate μ_i by taking the partial derivative of the above expression with respect to μ_i and set it equal to zero. Solving for μ_i will result in the update of

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

Under our setting, we would like the k-means clustering algorithm to be ϵ-differentially private. To achieve this property, the implementation of a differentially private k-means clustering follows the algorithm proposed in PINQ [13]. The main framework of this method is to implement component aggregation operations such as Count, Sum, Average and Median to be differential private.

The Count and Sum aggregation operators take ϵ as a parameter and returns an accurate count of the desired data with additive Laplace noise with density function $p(x) \propto e^{(-|x|)}$. Average and Median can be implemented using differentially private mechanisms such as the Exponential Mechanism [14].

VI. Discussion and Future Works

A. Evaluation

We provide three useful methods of gathering aggregate statistics in ϵ-differentially private manners for the central tracing app server to send information to the health authority or the general public. Any decisions then made after are with differentially private data; thus no parties in the protocol (except for the central tracing app server) should be able to derive additional sensitive information about any individual from the app server.

B. Future Works

In the future, we could consider applying notions of local differential privacy to a decentralized setting and computing aggregate statistics. We could also explore aggregate statistics on contact tracing information rather than user information for a graph database setting; for example, we could analyze the degree distribution of a graph based on contacts. This could be useful in a decentralized setting for sending automatic updates to people giving approximate likelihoods for having COVID.

VII. Conclusion

When the next pandemic will arrive is indeterminable but whether it will arrive is a positive certainty. Given how much more interconnected the world has become with the rise in faster and affordable transportation, it takes only a single mutation of a virus with the optimal attributes for mass infection to trigger another pandemic. By then, technology will be at least as better as now, and certainly contact tracing apps will return again.

Although contact tracing apps provide health authorities vital information to quickly contain local outbreaks, they still face many privacy concerns from users. As discussed in our threat model in Section IV, current architectures like the centralized and decentralized models face risks such as data misuse. In particular, some architectures assume a trusted data curator. The assurance that aggregate data are only used by "trustworthy authorities" is not a silver bullet to gaining the society's trust. Politicians and authorities can betray expectations by manipulating the semantics of their promises to get away with certain usage on the curated private information. So long as users are hesitant to use the apps, contact tracing apps cannot reach their full potential.

Hence, we introduce differential privacy for aggregate statistics to remove some of the reliance on trust between users and health authorities. Namely, we have provided three possible differentially private aggregate statistics that health authorities can use for data analysis.

References

[1] J. Bay, J. Kek, A. Tan, C. S. Hau, L. Yongquan, J. Tan, and T. A. Quy, "Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders," *Government Technology Agency-Singapore, Tech. Rep*, 2020.

[2] R. L. Rivest, H. Abelson, J. Callas, R. Canetti, K. Esvelt, D. K. Gillmor, L. Ivers, Y. T. Kalai, A. Lysyanskaya, A. Norige, B. Pelletier, R. Raskar, A. Shamir, E. Shen, I. Soibelman, M. Specter, V. Teague, A. Trachtenberg, M. Varia, M. Viera, D. Weitzner, J. Wilkinson, and M. Zissman, "The pact protocol specification," 2020. [Online]. Available: https://pact.mit.edu/wp-content/uploads/2020/11/The-PACT-protocol-specification-2020.pdf

[3] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha, "A survey of covid-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134 577–134 601, 2020.

[4] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" in *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008, pp. 531–540.

[5] N. Fernandes, M. Dras, and A. McIver, "Generalised differential privacy for text document processing," 2019.

[6] M. Winslett, Y. Yang, and Z. Zhang, "Demonstration of damson: Differential privacy for analysis of large data," *IEEE ICDE*, 2012.

[7] J. Zhang, X. Xiao, and X. Xie, "Privtree: A differentially private algorithm for hierarchical decompositions," 2016.

[8] X. Meng, H. Li, and J. Cui, "Different strategies for differentially private histogram publication," *Journal of Communications and Information Networks*, vol. 2, no. 3, pp. 68–77, 2017.

[9] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu, "Differentially private histogram publication," *IEEE ICDE*, 2012.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography*, pp. 265–284, 2006.

[11] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. C. Sevcik, and T. Suel, "Optimal histograms with quality guarantees," *Proceedings of the 24rd International Conference on Very Large Data Bases*, p. 275–286, 1998.

[12] X. X. Y. Y. M. W. Jun Zhang, Zhenjie Zhang, "Functional mechanism: Regression analysis under differential privacy," *VLDB '12*, 2012.

[13] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," 2009.

[14] F. McSherry and K. Talwar, "Mechanism design via differential privacy," *in FOCS*, pp. 94–103, 2007.