Database Query System with Budget Option for Differential Privacy Against Repeated Attacks

Jingyu Jia[1], Yuduo Wu[1], Yue Guo[1], Jiqiang Gao[1], Jin Peng[2], Peng Ran[2], and Min Li[1(✉)]

[1] College of Cyber Science, Nankai University, Tianjin, China
{1511372,1511406,2120180514}@mail.nankai.edu.cn,pdsgjq@163.com, limintj@nankai.edu.cn
[2] China Mobile Communications Corporation Research Institute, Beijing, China
{pengjin,ranpeng}@chinamobile.com

Abstract. Differential privacy enables data analysis while protecting individual privacy. However, existing differential privacy database platforms do not defend against repeated attacks. This paper proposes a practical Database Query System for differential privacy protection against repeated attacks with customizable privacy budget. By limiting adversary's success probability and the number of attacks, the administrator can protect privacy against the repeated attacks. We conduct an evaluation of this solution, and explain the applicability of this system.

Keywords: Differential privacy · Privacy budget · Repeated attack · Laplace mechanism

1 Introduction

Large volumes of personal data collected by various organizations are key resources for statistical analysis. Statistical data analysis and publishing is used extensively for potential economic and social benefits. However, the adversary may get personal information through different statistical query results, which is called differential attack. Differential privacy is proposed by Dwork [6] to resist differential attacks. It allows general statistical analysis of data while protecting individual data, and provides quantitative evaluation method for privacy protection. Although this mechanism guarantees privacy theoretically, the repeated attacks still may obtain personal information in practice. Therefore, the security assurance of differential privacy against repeated attacks need further research.

Generally, the organizations with amounts of private data provide an interface for clients to get statistics of specific data sets. The clients are limited to querying statistical results instead of individual personal information, but it is enough for them to pick out the specific group to send advertisement, to acquire the demand tendency of customers, et al r.

J. Li et al. (Eds.): SPNCE 2019, LNICST 284, pp. 46–57, 2019.
https://doi.org/10.1007/978-3-030-21373-2_5

In most cases, the statisticial query interface is provided in form of relational database that supports SQL queries. Various platforms are proposed for different query requests. Privacy Integrated Queries (PINQ), designed by McSherry et al. [13], provides a platform for interactive data analysis of real-time data with differential privacy. Weighted PINQ (wPINQ) [14], which uses weighted datasets to bypass difficulties encountered (the worst-case requirements of differential privacy for the noise magnitudes), is intended for graph analyses. Johnson et al. [1] develop FLEX with elastic sensitivity, a novel approach for differential privacy of SQL queries and their approach is compatible with real database systems.

These existing platforms provide a number of methods to deal with different query requests, but almost without exception, the platforms select the privacy budget arbitrarily without any specific setting mode. However, it is often of low security in the face of different attack models, especially the repeated attack. Privacy Budget is a parameter on the degree of privacy protection, concerning to the balance between security and utility. It is surely non-trivial to choose an appropriate value for privacy budget in accordance with whether the intuitive theoretical interpretation in the definition or the practical experience.

This paper proposes **Database Query System** for differential privacy protection against repeated attacks with customizable privacy budget, for statistical SQL queries. In contrast to existing work, our approach calculates the upper bound of privacy budget according to input parameters, which are related to specific attack model. We focus on repeated attacks model, by which the attackers can guess the real data. In this attack model, the guessed right or wrong depends on the amount of added noise and the number of attacks, which underscoring the importance of privacy preserving level. The success probability of a specific attack algorithm is computationally feasible, and hence the upper bound of privacy budget can be calculated by prescribing a limit to probability of success attack. The administrator input the number of attacks and the probability of success that in line with their expectations, then the privacy budget value that meets its requirements would be worked out.

Other than parameter choosing mechanism, elastic sensitivity is enforced to analyse the sensitivity of every SQL query. Compared to existing mechanisms for global sensitivity and local sensitivity, this mechanism is more efficient and supports majority of statistical queries in practice.

Furthermore, it is unnecessary for administrators to have knowledge of difference privacy. As long as administrators limit the cost for successful attack by attackers, the system would run well. This approach can make differential privacy more flexible to be used and help clients to find the most appropriate balance between protecting client privacy and ensuring data availability.

Contributions. We make two primary contributions in this paper:

1. Existing database platforms do not provide protection against repeated attack. In this paper, our system realizes the protection of repeated attacks by limiting attack times and attack success probability. The server calculates

privacy budget and the client's privacy budget allocation appropriately limits interaction numbers with the database.
2. This system allows administrators without background knowledge to set restrictions only by the acceptable risk level. It is more flexible and convenient for practical use in the organizations of different fields.

Paper Outline. The paper continues in three parts, the rest parts are organized as follows. In Sect. 2, we introduce the specific definition of differential privacy and the key methods used in our system design. The system is detailed in Sect. 3, and we show how clients can use the system to defend against attacks, as well as the experimental evaluation of the system, and Sect. 4 concludes this work and talks about the improvement direction.

2 Mathematical Foundations

2.1 Differential Privacy

Differential privacy is a privacy preserving method with rigorous theoretical basis, which provides higher security compared to others (k-anonymity, l-diversity, t-closeness). Under the protection of differential privacy, the specific personal data cannot be obtained even if the attacker has the greatest background knowledge.

This mechanism considers a set of databases. The distance measures how many records differ between two databases, which is denoted as $|D_1 - D_2|$. Two databases are neighbors if there is only one different record between them. The query function set for the database is $F = \{f_1, f_2, \cdots, f_n\}$, and algorithm M represents a randomized mechanism that meets the requirement of differential privacy. The randomized mechanism applies to the result of query function F. For any two neighbors D and D', P_M is set of all possible output of M, and O is its subsets ($O \in P_M$). If M satisfies the following inequalities, it is (ε, δ) -differential privacy.

$$Pr[M(D) \in O] \leq \exp(\varepsilon) Pr[M(D') \in O] + \delta \tag{1}$$

The core thought of differential privacy is to reduce the impact of a single record on query results by a randomized way. Of the two parameters, ε is privacy budget concerning the level of privacy protection, while δ is typically a function that grows more slowly than the inverse of any polynomial in the database size. We choose $\delta = n^{-\varepsilon \ln n}$ in the following sections according to Dwork's work [5].

2.2 Laplace Mechanism

Difference privacy is generally implemented by adding noise to the query result. Similar to other platforms, our system use Laplace mechanism to add noise. Laplace mechanism adds random noise of the Laplace distribution to the query

results to enforcing (ε, δ)-differential privacy. The size of the noise added is closely related to privacy budget ε and the query's sensitivity $\triangle f$. For different queries, the system will return the sum of query results and noise to satisfy difference privacy by algorithm M.

$$M(x) = f(x) + Lap(\frac{\triangle f}{\varepsilon}) \tag{2}$$

If the position parameter μ is set to zero, and the scale parameter set to b, the probability density function is $p(x) = \frac{1}{2b}e^{-\frac{|x|}{b}}$. For it's an absolute value function, the cumulative distribution function can be easily get by integrating:

$$F(x) = \frac{1}{2} + \frac{1}{2}sgnx(1 - e^{-\frac{|x|}{b}}) \tag{3}$$

In that case, the noise added by Laplace can be calculated according to the inverse cumulative distribution function. The scale parameter $b = \frac{\triangle f}{\varepsilon}$, and p denotes the probability of random selection from $(0, 1)$. $F^{-1}(p)$ is the noise added ultimately.

$$F^{-1}(p) = -bsgn(p - 0.5)\ln(1 - 2|p - 0.5|) \tag{4}$$

As shown in Fig. 1, the noise increases with the increase of b. So with the decrease of ε, more noise would be added to the primary result which means higher security but lower utility. When ε are 0, it means that a single piece of data has no effect on the query result. At this time, the added noise is very big, and the data is meaningless. If ε is set to a big value, the database is more likely to be attacked. Therefore, how to set the privacy budget becomes an important issue.

2.3 Privacy Budget

Privacy budget controls the level of privacy protection, and its selection is the key to achieve differential privacy. Several existing works have researched on this issue. Lee and Clifton [3] propose an attack model based on the prior and posterior belief, and give a method to calculate the upper bound of privacy budget. Then He et al. [10] improve the attack model by the definition of Laplace distribution and provide how to determine whether the object of attack is presence or not. Above schemes consider only a single attack, however, the attack on a specific object generally perform repeatedly to get individual information at a higher success probability. Therefore, the attack model of repeated attack [12] is put forward based on the former, as well as a more sophisticated way to compute the upper bound of privacy budget.

The result of a differential privacy mechanism is $f(D) + x$ for the query function f. Since the noise is of Laplace distribution, it is impossible for adversary to obtain x. Considering the features of some aggregation functions like count,

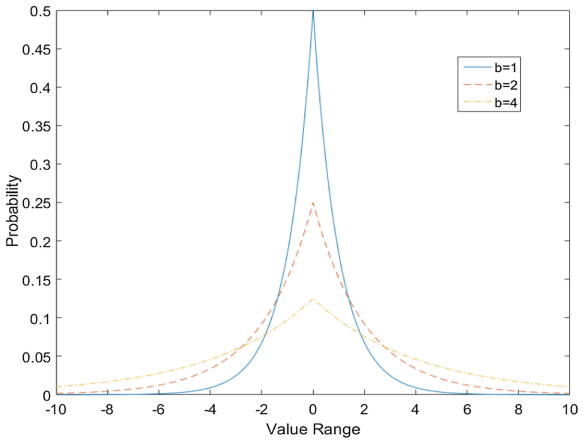

Fig. 1. Laplace probability density function

the real result could be leaked as long as the noise is in the range of certain interval. It is called fault-tolerant interval (which is a dangerous interval for the owner of individual data), and L denotes the half length of the interval. If L = 0.5, fault-tolerant interval is [−0.5,0.5], the real value of $f(D)$ could be inferred from the result of a count query. A single attack is ever so unreliable, therefore the repeated attack model performs the same query for N times, and guesses the private data of attack target in every query result according to L. Then the final determination could be made by the guess with higher probability. For a single attack, the success probability can be calculated by cumulative distribution value of the interval. The possibility of $f(D)+x$ in $(-\infty, f(D)+\mu+L)$ is equal to that of x in $(-\infty, \mu+L)$. Then according to the cumulative distribution function, the probability that x falls in this region is:

$$F(\mu + L) = 1 - \frac{1}{2}e^{-\frac{L\varepsilon}{\triangle f}}(\mu = 0) \tag{5}$$

If the success probability is limited to ρ, the upper bound of privacy budget calculate formula is:

$$\varepsilon \leq -\frac{\triangle f \ln 2(1-\rho)}{L} \tag{6}$$

The repeated attack follows binomial distribution $B(n, \rho)$. If the attacker attacks N times, n attacks occur in above interval, the possibility of this event is:

$$\sum_{i=1}^{n+1} C_{2n+1}^{n+i}(1 - \frac{1}{2}e^{-\frac{\varepsilon}{2}})^{n+i}(\frac{1}{2}e^{-\frac{\varepsilon}{2}})^{n+1-i} \tag{7}$$

While if the fact is $f(D) = y+1$, the probability in this case is the same as the result above. In conclusion, the probability of the adversary attacks successfully is the result above. According to this theorem, the privacy budget could be limited to a small range as long as the administrator set a certain value to the successful probability of the attacker. As seen in the inequality, the range of parameter selection is related to the sensitivity of a query, the number of attacks, fault-tolerance interval and the successful probability of the attacker

$$\rho \leq \sum_{i=1}^{n+1} C_N^{n+i} (1 - \frac{1}{2} e^{-\frac{L\varepsilon}{\triangle f}})^{n+i} (\frac{1}{2} e^{-\frac{L\varepsilon}{\triangle f}})^{n+1-i} \tag{8}$$

Actually, the ideal case is to make adversary in a dilemma of figuring out the presence or absencewhich happens when the probabilities of success and failure both are 50%. Therefore, what we can do is to limit the probability close to this value.

2.4 Elastic Sensitivity

Johnson et al. [1] propose Elastic Sensitivity, a mechanism that enforces differential privacy on majority of queries through analysis of SQL statements in practice. Elastic sensitivity meets the requirement of universal practical difference privacy, which can approximate the local sensitivity of queries with general equijoin.

The definition is expressed as $\acute{S}_R^{(k)}(r, x)$, which means the elastic sensitivity of query or relation r at distance k from the true database x. It origins from $mf_k(a, r, x)$, the maximum frequency of attribute a at distance k. When $k = 0$ the value $mf(a, r, x)$ is the database metric that needs to be precomputed. As calculating the approximation of upper bound, we should consider the worst condition in each case. The details of this method are listed in Table 1.

Before adding Laplace noise it should be smoothed by smooth sensitivity, as it can reduce the amount of noise to improve the utility of differential privacy. It is calculated by using the maximum value of elastic sensitivity at k multiplied by an exponentially decaying function in βk. By Nissim et al. [2] $\beta = \frac{\varepsilon}{2\ln(2/\delta)}$ suffices to provide differential privacy when applying Laplace mechanism. In the end, we release the query result as $q(x) + Lap(2S/\varepsilon)$.

3 Database Query System with Budget Option

After introducing the background knowledge, the following gives details of our system. The system design draws partly from PINQ and FLEX, their analysis of SQL statements brings inspiration to our design. We design Database Query System with a flexible privacy budget selection mechanism, to fight against differential privacy attack methods.

Table 1. Definition of elastic sensitivity

$Table\ t$	$mf_k(a,t,x) = mf(a,t,x) + k$	$\acute{S}_R^{(k)}(t,x) = 1$
$r_1 \bowtie r_2$ $a = b$	$mf_k\left(a_1, \begin{matrix} r_1 \bowtie r_2 \\ a2 = a3 \end{matrix}, x\right) =$ $\begin{cases} mf_k(a_1,r_1,x)\, mf_k(a_3,r_2,x) & a_1 \in r_1 \\ mf_k(a_1,r_2,x)\, mf_k(a_2,r_1,x) & a_1 \in r_2 \end{cases}$	$Self\ Join$
		$\acute{S}_R^{(k)}\left(\begin{matrix} r_1 \bowtie r_2 \\ a = b \end{matrix}, x\right) =$ $mf_k(a,r_1,x)\, \acute{S}_R^{(k)}(r_2,x) +$ $mf_k(a,r_2,x)\, \acute{S}_R^{(k)}(r_1,x) +$ $\acute{S}_R^{(k)}(r_1,x)\, \acute{S}_R^{(k)}(r_2,x)$
		$Non-self\ join$
		$\acute{S}_R^{(k)}\left(\begin{matrix} r_1 \bowtie r_2 \\ a = b \end{matrix}, x\right) = max$ $mf_k(a,r_1,x)\, \acute{S}_R^{(k)}(r_2,x)\,,$ $mf_k(a,r_2,x)\, \acute{S}_R^{(k)}(r_1,x)$
$\pi_{a1,\cdots,an} r$	$mf_k(a,\pi_{a1,\cdots,an} r,x) = mf_k(a,r,x)$	$\acute{S}_R^{(k)}(\pi_{a1,\cdots,an} r,x) = \acute{S}_R^{(k)}(r,x)$
$\sigma_\varphi r$	$mf_k(a,\sigma_\varphi r,x) = mf_k(a,r,x)$	$\acute{S}_R^{(k)}(\sigma_\varphi r,x) = \acute{S}_R^{(k)}(r,x)$
$Counting$	$mf_k\left(a, Count\left(r\right), x\right) = \perp$	$Count(r)$
		$\acute{S}_R^{(k)}\left(Count\left(r\right)\right) = 1$ $\acute{S}_R^{(k)}\left(Count\left(r\right), x\right) = \acute{S}_R^{(k)}\left(r,x\right)$
		$Count\ with\ gourping \begin{matrix} Count\left(r\right) \\ G1 \ldots Gn \end{matrix}$
		$\acute{S}_R^{(k)}\left(\begin{matrix} Count\left(r\right) \\ G1 \ldots Gn \end{matrix}, x\right) =$ $2\acute{S}_R^{(k)}(r,x)$

3.1 System Design

Considering the shortcomings of the existing database platforms based on differential privacy protection, our system provides more privacy protection management authority for data publishers. The administrator can limit attack times and adversary's success probability, then they can estimate the attack cost of the adversary. Similar to PINQ, administrator can assign a query budget to clients, and each query of clients will consume part of the query budget until the query budget is used up.

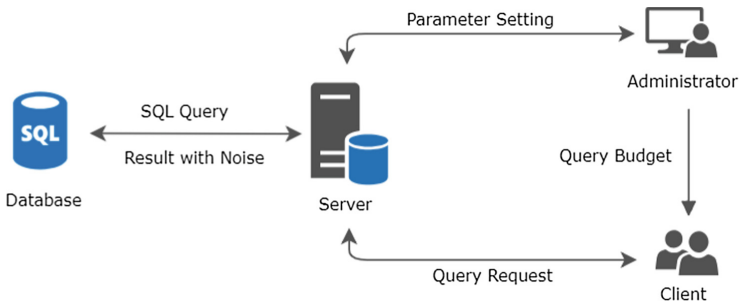

Fig. 2. System interactions

The interaction process is sketched in Fig. 2. Before data release, the administrator inputs the attack times and success probability as expected into the system. Then, the system calculates the privacy budget based on the parameters input by the administrator and returns privacy budget to the administrator. According to the privacy budget obtained, the administrator limits the scope of client access to the database and assigns a query budget to the client, and each query of the client will consume part of the budget. When the budget is insufficient, the client can apply to the administrator, who decides whether to assign another query budget to the client. When the client poses a query request to the server, the server sends a SQL statement to the database of query, and calculates the elastic sensitivity according to the query statement. The client's query budget decreases accordingly. Afterwards, the database returns the query results to the sever, and the server adds noise of Laplace distribution to the query results. Finally, the result with noise is returned to the client.

3.2 Budget Management

Here's more information on how to calculate a privacy budget. The administrator inputs the probability of success and attack times based on the security requirements, and calculates the privacy budget according to the repeated attack model proposed above. Considering the attack model, the probability of success under repeated attacks is actually equal to a binomial distribution of the success probability of a single attack. Therefore, we can simplify the probability formula of repeated attacks to the following formula:

$$\rho' \leq \sum_{i=1}^{n+1} C_N^{n+i} (\rho)^{n+i} (1-\rho)^{n+1-i} \tag{9}$$

N is the number of repeated attacks, N = 2n+1, ρ is the success probability of a single attack, and ρ' is the success probability of repeated attacks.

In this way, the transformation calculation of the success probability of a single attack becomes polynomial. Based on the single success probability ρ, it is easy to calculate the privacy budget ε. Under the same privacy budget, the success probability increases with the increasing number of attacks. Therefore, controlling the number of queries is an effective way to protect personal data security. The administrator can assign the query budget to the client based on the privacy budget. Each query of the client consumes the query budget. When the query budget is exhausted, the client needs to apply to the administrator for another budget, and the administrator decides whether to assign another budget to the client based on specific usage of client and security requirements attack success probability. When setting parameters, the attack success probability should be greater than 0.5. In fact, under the optimal protection, the success probability of the adversary will be maintained at 0.5. Because under optimal protection, even if the opponent has the most background knowledge, it cannot profit from it. That is, the adversary can only determine the information of the attack target by guessing rather than extracting useful data from the

dataset. Under optimal protection, an attack on a counting query is like guessing the heads and tails of a coin, whose success rate is 50%.

3.3 Security Evaluation

The essence of Laplace mechanism is to add a random noise to the result, and the random noise is in the interval $(-\infty, +\infty)$ with an average of 0. Therefore, the target value can be obtained according to the feature of the average value as long as executing the same query enough number of times.

Considering the attack model by predicting the noise, take the count query for example. For different values of privacy budget ε the adversary averages the noise of S number of attack times experimental results. When the average value is within the $[-0.5, 0.5]$ interval (known as the "fault-tolerant interval"), the adversary can infer that the average value of noise is 0. Therefore, we need pay attention to the probability of the average value of S times noise in the dangerous interval.

The Table 2 shows the data distribution and mean values of noise under different conditions for different values of privacy budget ε. For example, when the value of privacy budget ε is 0.1, 90% noise is within the range of $[-23.24, 23.24]$ and 95% noise is within the range of $[-29.00, 29.99]$. Averaging 100 different noise, the average falls in the danger range with 25.59% probability; and averaging 1,000 noise, the average falls in the danger range with 73.75% probability. That is to say, when privacy budget ε is 0.1, the adversary has a 73.75% probability of getting the true value of the query result through executing the same query 1,000 times.

Table 2. The Laplace noise distribution and the possibility of noise mean within danger interval in different ε

ε	Data distribution of noise (Tabsolute value of T)				The probability that the average noise of S times in dangerous interval			
	90%	95%	99%	99.9%	100	1000	10000	100000
1	2.29	2.98	4.50	6.43	100.00	100.00	100.00	100.00
0.1	23.24	29.99	45.51	66.56	25.59	73.75	99.99	100.00
0.01	227.97	296.22	463.48	677.26	2.72	9.12	27.85	73.70

The above attack model is an attack on the characteristics of Laplace noise. While considering the adversary's background knowledge, the adversary can use another repeated attack model to guess the information of the target. That is, the repeated attack model introduced in Sect. 2.3. Take count query as an example. Because the adversary has extensive background knowledge, the adversary knows that the value of the query result $f(D)$ is y or $y+1$. Set $N = 2n+1$, the adversary executes the same query N times and counts the result $f(D) + noise$ in $(-\infty, \text{y}+0.5)$ or in $(\text{y}+0.5, +\infty)$. If the result in $(-\infty, \text{y}+0.5)$ is more than $(\text{y}+0.5, +\infty)$, the adversary will guess the query result $\text{f(D)} = \text{y}$. Otherwise, the

adversary will guess the query result $f(D) = y + 1$. In Sect. 2.3, we list the success probability of this attack model is:

$$\rho \leq \sum_{i=1}^{n+1} C_N^{n+i} (1 - \frac{1}{2} e^{-\frac{L\varepsilon}{\triangle f}})^{n+i} (\frac{1}{2} e^{-\frac{L\varepsilon}{\triangle f}})^{n+1-i}$$

According to the formula and Fig. 3, under the same privacy budget ε, the success probability increases with the increasing of query times.

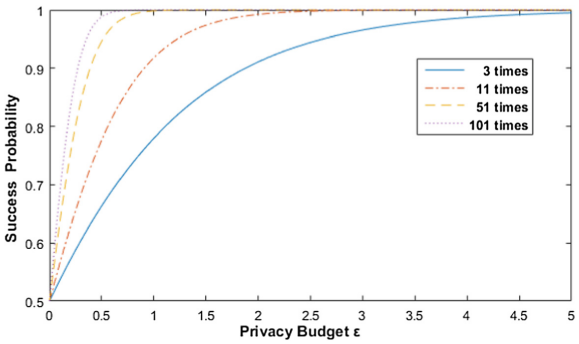

Fig. 3. The success probability of counting query under different attack times

To evaluate the security of the Database Query System, we analyze the input parameters and privacy budget ε in Table 3. We also list the success probability of the first attack model. As mentioned above, the ideally attack success probability is 0.5. The significance of 0.5 is that the adversary cannot get advantage by his extensive background knowledge and can only guess target value. Based on this privacy budget, the predicting noise attack model success probability is less than 50%. That is to say, by calculating the average value of the noise, the success accuracy is evenworse than that of random guessing.

Table 3. Protection model evaluation

Attack times	Success probability	Privacy budget	The probability average noise in dangerous interval	Noise mean
51	0.6	0.0717	14.57%	13.93
101	0.6	0.0510	13.80%	19.67
201	0.6	0.0359	13.86%	27.94
101	0.7	0.1068	30.01%	9.3319
101	0.55	0.0250	7.17 %	39.89
101	0.65	0.0779	22.04 %	12.90

3.4 Performance Analysis

The main time overhead before data publishing is the polynomial computation process of privacy budget. After releasing the database, neither the calculation of elastic sensitivity nor the generation of Laplace noise involves complex calculation, and the time overhead is relatively small. In each query, in addition to ordinary interactions with the database, only a simple calculation is required to achieve differential privacy protection, and the calculation overhead can be ignored. This system has the advantage of easy implementation. The privacy budget is set reasonably, and the elastic sensitivity value is small and more consistent with the original data characteristics. Therefore, the data distortion degree after adding noise is small, and the actual accuracy of query results can be controlled by the administrator freely.

4 Conclusions

This paper presents Database Query System, a platform for privacy preserving statistical analysis against repeated attacks. In consideration of the attack in practical, we refer to several attack model and calculate the privacy budget of Laplace Mechanism by limitation of number of attacks and success probability. For further research, the aggregate functions except for count should be take into consideration. And abstracting out the complicated attack in practice into a detailed algorithm is a challenge as well.

References

1. Johnson, N.M., Near, J.P., Song, D.: Towards practical differential privacy for SQL queries. PVLDB **11**(5), 526–539 (2018)
2. Nissim, K., Raskhodnikova, S., Smith, A.D.: Smooth sensitivity and sampling in private data analysis. In: STOC 2007, pp. 75–84 (2007)
3. Lee, J., Clifton, C.: How much is enough? choosing ε for differential privacy. In: Lai, X., Zhou, J., Li, H. (eds.) ISC 2011. LNCS, vol. 7001, pp. 325–340. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24861-0_22
4. Johnson, N.M., Near, J.P., Hellerstein, J.M., Song, D.: Chorus: Differential Privacy via Query Rewriting. CoRR abs/1809.07750 (2018)
5. Dwork, C., Lei, J.: Differential privacy and robust statistics. In: STOC 2009, pp. 371–380 (2009)
6. Dwork, C.: Differential Privacy. Encyclopedia of Cryptography and Security, 2nd edn, pp. 338–340. Springer, Boston (2011). https://doi.org/10.1007/978-1-4419-5906-5
7. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
8. Zhu, T., Li, G., Zhou, W., Yu, P.S.: Differentially private data publishing and analysis: a survey. IEEE Trans. Knowl. Data Eng. **29**(8), 1619–1638 (2017)
9. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)

10. He, X., Wang, X.S., Chen, H., Dong, Y.: Study on choosing the parameter ε in differential privacy. J. Commun. **36**(12) (2015)
11. Xiong, P., Zhu, T.-Q., Wang, X.-F.: A survey on differential privacy and applications. Chin. J. Comput. **37**(1), 101–122 (2014)
12. Hao, C., Peng, C., Zhang, P.: Selection method of differential privacy protection parameter ε under epeated attack. Comput. Eng. **44**(7), 145–149 (2018)
13. McSherry, F.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. Commun. ACM **53**(9), 89–97 (2010)
14. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
15. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOCS 2007, pp. 94–103 (2007)