Distributed Cardinality Estimation of Set Operations with Differential Privacy

Rade Stanojevic
Qatar Computing Research Institute
Hamad Bin Khalifa University
Email: rstanojevic@hbku.edu.qa

Mohamed Nabeel
Qatar Computing Research Institute
Hamad Bin Khalifa University
Email: mnabeel@hbku.edu.qa

Ting Yu
Qatar Computing Research Institute
Hamad Bin Khalifa University
Email: tyu@hbku.edu.qa

***Abstract*—In this paper we study the problem of estimating the cardinality of pairwise set operations (union and intersection) over sets possessed by different data owners, while preserving differential privacy. In our problem setting, a data owner could only communicate with an untrusted server, and thus have to perturb its set data for privacy protection before sharing them with the server. This problem setting is relevant to diverse applications in practice, including sensor-based traffic monitoring, cross-domain data integration, and combining findings from multiple surveys. To tackle this problem, we first adopt existing randomized response technique to perturb the bit vector (to achieve differential privacy) and develop tools which the server can use to derive the cardinality of set operations from the randomized bit vectors. However, the variance of the union/intersection estimator grows linearly with the universe (bit-vector) size which is impractical for large universes. To keep the variance low we in addition propose to resort to Bloom filters instead of high-dimensional bit vectors to share set data with the server. The key insight is that in spite of inevitable collisions in BF by keeping its size small we can bound the variance of the union/intersection cardinality estimators. Finally, we show that investing a small part of the privacy budget into reporting (obfuscated) set cardinality can further reduce the estimator errors for up to 20%. Our empirical analysis reveals the impact of various parameters including privacy budget and Bloom filter size on the overall accuracy of the approach and demonstrates the utility of the proposed solution.**

I. Introduction

In this paper we study the following problem. Suppose we have two autonomous data owners and that possess sets K_1 and K_2 respectively. Each member of the two sets is an individual user. A server would like to know the cardinality of the union or intersection of K_1 and K_2. However, the membership of an individual in K_1 and K_2 is sensitive and the server is not trusted with this sensitive information. Further, we assume that the communications could only be between each data owner and the server, but not between the two data owners. Given this setting, the question that we would like to study is: could the server still be able to estimate the cardinality of $|K_1 \cup K_2|$ and $|K_1 \cap K_2|$ with reasonable accuracy without compromising the privacy of each individual?

Though the above problem setting seems rather simple, it indeed could be mapped to quite diverse real-world applications, such as sensor-based traffic monitoring, customer tracking data integration and survey data integration. We will provide more discussions about such applications later in this section.

978-1-5386-1027-5/17/$31.00 ©2017 IEEE

We note that simple solutions such as anonymization, aggregation or substituting with data from similar distributions do not provide strong privacy guarantees and these solutions are proved to be vulnerable to de-anonymization attacks [45], [36]. Differential privacy [19] has been proven to solve such privacy issues of sensitive data publication under various settings. Thus, in this paper, we adopt differential privacy as the target privacy model. Specifically, we model a set as a bit vector, where a bit is set to 1 if and only if the corresponding user belongs to the set. To ensure differential privacy, we adopt an existing randomized response technique to flip each bit with certain probability before sharing it with the server. As the bit vector is heavily perturbed, it is very challenging to derive the cardinality of set operations. We present a novel algorithm to solve the problem along with a detailed theoretical analysis of its accuracy.

As we will show later, when the cardinality of the original sets is much smaller than the universe of all users, the noise due to random bit flipping could significantly overwhelm the actual signal. Further, if the universe is very large, it would also impose high communication costs, which may render the solution impractical for certain applications (e.g., sensors with limited communication capabilities). As a remedy, we further propose to leverage Bloom filters instead of high-dimensional bit vectors to share data with the server. The key observation is that in spite of inevitable collisions in Bloom filter by keeping the BF size small one can bound the variance of the cardinality estimators which otherwise grow linearly with universe size in the bit vector case.

Briefly, the following are our main contributions:

- We develop a technique which can be used to estimate the cardinality of set union/intersection/difference of two sets shared with an untrusted third party in a differentially private way.
- We evaluate the variance of the proposed estimators analytically and empirically and show that randomized Bloom filter approach is superior to bit-vector approach when the universe size is much larger than the sets of interest.

- We show that splitting the privacy budget and sharing explicit set cardinality (with Laplace noise) with the server, in addition to the perturbed BF, can further reduce the errors of the estimator by up to 20%.
- We examine proposed approach empirically, discuss the interplay of the set sizes, privacy budget and the expected errors, and offer guidelines on how to appropriately select the relevant parameters.

One of the closely related work to our paper is proposed by Alaggan et al. [4], which focuses on deriving the cardinality of set intersections of call detail records with differential privacy. Though their approach also utilizes randomized Bloom filter, our analytic methods is completely different, and results in much more accurate estimations. In section IV-F, we briefly describe their scheme and critically evaluate their main results. In section V, we compare empirically with the technique in [4] and show the advantages of our approach.

A. Example Applications

Here we give some example applications where the problem we study could be applied.

Traffic monitoring and analysis. Sensors are increasingly deployed at road intersections and other critical locations to collect data for traffic monitoring and analysis [7], [4]. For example, sensors could collect the bluetooth device MAC addresses of the passing vehicles and send such information through cellular networks to a central server, which could then estimate various traffic statistics, e.g., the intersection of vehicles passing two adjacent sensors at different timestamps could help estimate the severity of congestion along a road; the union of all the sensors in an area during a period of time would give the total number of vehicles in that area. On the other hand, it has been well recognized that collecting all the detailed information of bluetooth device IDs could impose serious privacy risks, as the server could potentially track the trajectories of individuals [5], [37]. Therefore, it is very desirable to have solutions that allow the server to accurately estimate the cardinality of set operations without revealing the private information of individuals collected by each sensor.

Integration of customer tracking data. Nowadays department stores are very interested in tracking the behavior of their customers, e.g., when they enter the store, how long they stay and which sections they visit mainly using WiFi tracking [37]. By combining data from different time or location, a department store could obtain many useful statistics, e.g., the number of regular customers (intersection of the customer sets from each month) or the size of unique customers visiting different branches in a region (union of the customer sets from stores at different locations). Due to privacy regulations and to avoid liability, the collected data may need to be perturbed before they are stored for future analysis[37]. In other words, the above analysis with set operations has to be performed over perturbed data.

Combining data from multiple surveys. Suppose several organizations conduct independent surveys (e.g., one on health history, another on diet habits, and yet another one on exercise) over the same population [43]. Combining the results from multiple surveys could help researchers discover interesting correlations between different aspects of the population: how diet habits correlate with certain diseases, and to what extent regular exercise combined with healthy diet would contribute to the reduction of diabetes. Such analysis requires the cardinality of various set operations over individual survey results. Meanwhile, surveyors are reluctant to share their original survey data. Instead, they must be processed for privacy protection before shared. In fact, due to the sensitive nature of some of these surveys, a certain randomization has already been applied before users send their responses to surveyors. The challenge is then how to derive the cardinality of set operations with such perturbed data.

Finally, a large number of set similarity measures have been proposed including Jaccard index, Sorensen index and Tversky index [30] and virtually every one of them is a function of the cardinality of the set intersection/union/difference. Our approach would enable approximating all mentioned similarity indices while at the same time providing strong privacy guarantees.

We would like to emphasize that the setting of our problem is completely different from that of traditional secure multiparty computation [48]. First, the data owners do not need to know the final cardinality results of set operations. They only share data with the untrusted server while ensuring privacy of individuals. Second, data owners only communicate with the server and do not communicate with each other. This is sometimes due to the design of applications. For example, there is typically no need for sensors to communicate with each other in traffic monitoring systems. It may also be due to practical issues when involving entities from different domains. Often times surveys are conducted by surveyors from different organizations. It would be easier for them to have a one-time sharing of data with an outside researcher than engaging in multi-round communications among surveyors. In some sense, our problem setting adopts the “publishing” model. Data owners publish their data once in a privacy-preserving way. It is then up to the untrusted server to figure out how to derive cardinality of various set operations.

B. Organization

The rest of the paper is organized as follows. We briefly introduce differential privacy and its important properties in section II. In section III, we discuss the use of bit perturbation technique to achieve differential privacy while sharing set data. We also show in detail cardinality estimation of set operations and provide a theoretical analysis of its variance. Section IV proposes optimization using Bloom filters and shows the detailed steps to derive cardinalities through randomly flipped Bloom filters. Empirical evaluation and validation are reported in section V. Section VI discusses closely related work to this paper, followed by our conclusion in section VII.

II. Differential Privacy

The notion of differential privacy (DP) [19] is by now well-known, and here we only present a short summary of its main concepts. It is a statistical model of privacy based on the concept of *neighboring datasets*: two sets K and K' are neighboring sets, denoted $K \sim K'$, if one is a subset of another and their sizes only differ by 1. In other words, K and K' are almost the same except one has exactly one more element than the other. Intuitively, differential privacy requires that the behavior of an algorithm should be (mostly) insensitive to the presence or absence of any element in the input. Formally, we have the following definition:

Definition II.1 (ϵ-differential privacy)**.** Given any pair of neighboring sets K and K', a randomized algorithm $\mathcal{M}$ is *ϵ-differentially private* if for all $S \subseteq Range(\mathcal{M})$:

$$\frac{Pr[\mathcal{M}(K) \in S]}{Pr[\mathcal{M}(K') \in S]} \leq e^{\epsilon}$$

Here ϵ is a public privacy parameter often referred as the *privacy budget*. It controls the strength of privacy guarantees: the smaller ϵ is, the closer the distributions $\mathcal{M}(K)$ and $\mathcal{M}(K')$ are, and thus a stronger privacy is ensured by an algorithm.

Many techniques could be used to achieve differential privacy. For real-valued functions $f : \mathcal{K} \to \mathbb{R}^d$, a common technique is to inject carefully calibrated random noise into the output. The magnitude of the noise is determined by the *global sensitivity* of f, defined as $\Delta f = max_{K \sim K'} ||f(K) - f(K')||_1$, where $|| \cdot ||_1$ is the L_1 norm. Global sensitivity reflects the maximum possible change any one element could make to the output of f. In the Laplace mechanism, a function f could be made ϵ-differentially private by adding random noise drawn from the Laplace distribution with mean zero and scale $\lambda = \frac{\Delta f}{\epsilon}$, denoted Lap($\lambda$), to its output. Clearly, the larger Δf or the smaller ϵ is, the more noise needs to be added to the output.

Differential privacy has several important properties. First is its composability. If we run k mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ on an input K in sequence and each independently satisfies ϵ_i-differential privacy, then the full output is $(\Sigma_{i=1}^{k} \epsilon_i)$-differentially private. In other words, the privacy loss accumulates linearly. Meanwhile, if the mechanisms are applied to *disjoint* inputs, then the resulting full output is $max(\epsilon_i)$-differentially private. Second, any post-processing of the output of a differentially private mechanisms does not reduce its privacy guarantee. With these properties, one could easily design DP mechanisms to perform complex data analysis tasks from simple differentially private components.

As described in Section I, the setting of our problem is composed of two technical components. The first is for a data owner to share its set K with the server. This component has two requirements: (1) the sharing mechanism $\mathcal{M}$ must satisfy differential privacy; and (2) the output from $\mathcal{M}$ should still preserve some membership information of K that could be used by the server. The second component is then an algorithm for the server to do cardinality estimation. Next, we describe our design of each component in detail. For simplicity, in the rest of our discussion we consider the case of two data owners. However, our technique could be easily extended to handle cases with more than two data owners.

III. Cardinality Estimation using Perturbed Bit Vectors

A. Share Set Data with Differential Privacy

Let $U = u_1, \ldots, u_L$ be the set of all possible users in an application domain U, which could be all the vehicles in a city, all the people in a county of the shopping mall, or the survey population. A set K could be translated into a L-dimension bit vector, such that the bit at position i corresponds to user u_i, and it is set to 1 if and only if $u_i \in K$. To share K with differential privacy, we adopt the randomized response technique that was originally designed to conduct surveys with sensitive questions [47]. Though there are several versions of this technique [3], we adopt a simple scheme as follows. For each bit b_i in the bit vector, with probability p we flip it (i.e., replace it by $1 - b_i$), and with probability $q = 1 - p$ we keep its original value. The randomly flipped bit vector is then shared with the server.

Theorem III.1. The random bit-flipping scheme satisfies ϵ-differential privacy with the following value of p:

$$p = \frac{1}{1 + e^{\epsilon}}$$

Proof. Denote the random bit-flipping scheme as $\mathcal{M}$, and the probability to change a bit from v to v' as $pr(v \to v')$ (i.e., $pr(1 \to 1) = pr(0 \to 0) = q$ and $pr(0 \to 1) = pr(1 \to 0) = p$).

Let K and K' be two neighboring sets whose corresponding bit vectors are $V = (v_1, \ldots, v_L)$ and $V' = (v'_1, \ldots, v'_L)$ respectively.

Since $K \sim K'$, there is only one bit difference between V and V'. Without loss of generality, assume that it is the first bit.

Given any output vector $X = (x_1, \ldots, x_L)$, we have $Pr[\mathcal{M}(V) = X] = \Pi_{i=1}^{L} pr(v_i \to x_i)$, and $Pr[\mathcal{M}(V') = X] = \Pi_{i=1}^{L} pr(v'_i \to x_i)$. Since V and V' only differ in the first bit, and $p \leq 0.5$,

$$\frac{Pr[\mathcal{M}(V) = X]}{Pr[\mathcal{M}(V') = X]} = \frac{pr(v_1 \to x_1)}{pr(v'_1 \to x_1)} \leq \frac{q}{1-q} = e^{\epsilon}$$

□

Note that, the goal of privacy protection in our problem setting is to prevent inference of an individual user's membership and the presence or absence of a user's membership could only affect one set. Therefore, even though each data owner invokes the random bit-flipping mechanism with privacy budget ϵ, the overall privacy guarantee would still provide ϵ-differential privacy.

B. Estimate Set Cardinality

Now we investigate how the server derives the cardinalities of the union and intersection of two sets after receiving their corresponding randomly flipped bit vectors. Recall that the dimension of each vector is L, the size of the total population for a particular application domain.

Consider a set K with cardinality $|K|$. Denote its corresponding bit vector as V and the vector after random bit flipping as V_p. Let n_0 (n_1) be the number of 0s (1s) in V, and m_0 (m_1) be the number of 0s (1s) in V_p.

The server can directly get m_0 and m_1 from V_p, which can then be used to estimate n_0 and n_1 using a simple mean-field model. Namely, for large L, m_0 and m_1 can be estimated by:

$$E(m_0) = qn_0 + pn_1,$$

$$E(m_1) = pn_0 + qn_1,$$

where we introduced $q = 1 - p$ earlier. Solving the above linear system gives us the estimators for n_0 and n_1 for given observed m_0 and m_1:

$$n_0 = \frac{qm_0 - pm_1}{q - p}, \quad n_1 = \frac{qm_1 - pm_0}{q - p} \tag{1}$$

We use the estimator for set size:

$$|\hat{K}| = \left(\frac{qm_1 - pm_0}{(q - p)}\right) = L - \frac{qm_0 - pm_1}{q - p} \tag{2}$$

C. Estimate Union/Intersection Cardinality

When having two sets, the estimation of cardinality of set union and intersection can be done following a similar principle. Following the notation above, for two sets K_1 and K_2, we denote their corresponding non-flipped vectors by V_1 and V_2 and the flipped ones by V_{p1} and V_{p2} respectively. We also denote the number of positions s in which $V_1[s] = i$ and $V_2[s] = j$, for $i, j \in \{0, 1\}$ by n_{ij} and similarly the number of positions s in which $V_{p1}[s] = i$ and $V_{p2}[s] = j$, for $i, j \in \{0, 1\}$ by m_{ij} . For example, n_{00} denotes the positions in both non-flipped bit vectors having 0s.

Using the same logic as above, we can derive the mean-field relationship between n's and m's:

$$E(m_{00}) = q^2 n_{00} + pqn_{01} + pqn_{10} + p^2 n_{11},$$

$$E(m_{01}) = pqn_{00} + q^2 n_{01} + p^2 n_{10} + pqn_{11},$$

$$E(m_{10}) = pqn_{00} + p^2 n_{01} + q^2 n_{10} + pqn_{11},$$

$$E(m_{11}) = p^2 n_{00} + pqn_{01} + pqn_{10} + q^2 n_{11},$$

Interestingly, the inverse of the matrix A of this 4×4 linear system is very elegant and provides a concise solution for n's as a function of the observed variables $m_{00}, m_{01}, m_{10}, m_{11}$ as follows:

$$\begin{bmatrix} n_{00} \\ n_{01} \\ n_{10} \\ n_{11} \end{bmatrix} = \frac{1}{(q-p)^2} \begin{bmatrix} q^2 & -pq & -pq & p^2 \\ -pq & q^2 & p^2 & -pq \\ -pq & p^2 & q^2 & -pq \\ p^2 & -pq & -pq & q^2 \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{10} \\ m_{11} \end{bmatrix} \tag{3}$$

Similar to Section III-B, the number of 00 positions in non-flipped vectors V_1 and V_2 is determined by the size of the union of K_1 and K_2:

$$n_{00} = L - |K_1 \cup K_2|, \tag{4}$$

which allows us to estimate the cardinality of the union as:

$$|K_1 \hat{\cup} K_2| = L - \left(\frac{q^2 m_{00} - pqm_{01} - pqm_{10} + p^2 m_{11}}{(q-p)^2}\right) \tag{5}$$

Meanwhile, $\hat{n_{11}}$ serves directly as an estimator of the cardinality of set intersection, i.e.,

$$|K_1 \hat{\cap} K_2| = \hat{n_{11}} = \frac{p^2 m_{00} - pqm_{01} - pqm_{10} + q^2 m_{11}}{(q-p)^2} \tag{6}$$

Next we study the quality of these estimators which could be captured by their variance. We are particularly interested in finding out which parameters have significant impacts on their variance.

D. Variance Analysis

Recall that $|\hat{K}|$ denotes the estimator of the set size using the bit vector of size L given by eq. (2). It is not difficult to derive the variance of $|\hat{K}|$:

$$Var(|\hat{K}|) = Var(\frac{qm_0 - pm_1}{(q-p)}) =$$

$$= Var(\frac{qm_0 + pm_0 - p(m_0 + m_1)}{q - p}) =$$

$$Var(\frac{m_0 - pL}{q - p}) = \frac{1}{(q-p)^2} Var(m_0) = \frac{1}{(q-p)^2} Lpq.$$

Thus, the variance of the set size estimator does not depend on the size of the set $|K|$, but on the universe size (L) and p and q. For a large universe size (L), the error in estimating the set size could be substantial.

The exact variance of the estimator for the cardinality of set union and intersection would be much more involved. In particular, m_{ij}, $i, j \in \{0, 1\}$, is in fact a summation of four binomial distributions. For example, $m_{00} = B(n_{00}, q^2) + B(n_{01}, pq) + B(n_{10}, pq) + B(n_{11}, p^2)$, where $B(n, p)$ denotes a binomial distribution with n trials and p success probability in each trial. Further, the estimators of $|K_1 \cup K_2|$ and $|K_1 \cap K_2|$ are linear combinations of random variables m_{00}, m_{01}, m_{10} and m_{11}. To make the analysis further complicated, m_{00}, m_{01}, m_{10} and m_{11} are not independent due to the fact $m_{00} + m_{01} + m_{10} + m_{11} = L$.

Even though we were not able to derive the explicit expression for the variance of the union estimator, we can show that asymptotically it grows linearly with L.

Theorem III.2. For given K_1, K_2, p and q, there is a constant α which does not depend on L such that:

$$Var(q^2 m_{00} - pqm_{01} - pqm_{10} + p^2 m_{11}) = \alpha L + O(1), L \to \infty.$$

Proof. As we explain above each m_{ij} is a sum of 4 binomial random variables. Take m_{00}, it can be seen as a sum:

$$m_{00} = B(n_{00}, q^2) + B(n_{01}, pq) + B(n_{10}, pq) + B(n_{11}, p^2),$$

where $B(n, r)$ denotes a binomial distribution with n trials and r success probability in each trial. It is rather straightforward to see that for large L: $n_{00} = L - |K_1 \cup K_2| = L + O(1)$ and that the m_{00} is dominated by the $B(n_{00}, q^2)$. We will rewrite the above equation as:

$$m_{00} = B_{00}(q, q) + O(1),$$

where $B_{00}(q, q)$ is the number of positions in the bit vector which has 0 in both flipped and non-flipped vectors. Similarly:

$$m_{01} = B_{00}(q, p) + O(1),$$

$$m_{10} = B_{00}(p, q) + O(1),$$

$$m_{11} = B_{00}(p, p) + O(1).$$

Hence,

$$\begin{aligned} &Var(q^2 m_{00} - pq m_{01} - pq m_{10} + p^2 m_{11}) = \\ &Var(q^2 B_{00}(q, q) - pq B_{00}(q, p) - pq B_{00}(p, q) \\ &\qquad + p^2 B_{00}(p, p)) + O(1). \end{aligned} \tag{7}$$

For any $r, s, r', s' \in \{p, q\}$, $(r, s) \neq (r', s')$:

$$Var(B_{00}(r, s)) = n_{00} rs(1 - rs)$$

$$Cov(B_{00}(r, s), n_{00}(r', s')) = n_{00} pq(rs + r's')$$

Now the variance of the above linear combination can be expressed as:

$$\begin{aligned} &Var(q^2 B_{00}(q, q) - pq B_{00}(q, p) - pq B_{00}(p, q) + p^2 B_{00}(p, p)) = \\ &= n_{00}[q^6(1 - q^2) + 2p^3q^3(1 - pq) + p^6(1 - p^2) - 4p^2q^5 \\ &-4p^5q^2 + 2p^3q^3(p^2 + q^2) + 4p^4q^4] = \alpha n_{00} = \alpha L + O(1). \end{aligned}$$

where α depends on p and q but not L. □

We also show empirically that variance scales linearly with the universe size L. In Figure 1, we report the sample variance on 100 flipping runs, estimating the union using (5) of two sets with 100 elements with union of 150 elements. It is obvious from the figure that the observed variances grow linearly with L.

We observe that, in practical application settings, the universe size L is easily several orders of magnitude higher than the cardinality of the sets from data owners. For example, the number of vehicles in a city is often in the order of hundreds of thousands to millions, while everyday there could only be thousands of vehicles in a region of the city. Similarly, though the potential customer base could be huge, the actual customers visiting a store in a day or a month would be much smaller. Therefore, the above mechanism discussed so far would be likely to generate significant noise and render very inaccurate estimations. Another practical issue is that, though it is relatively easy to know the size of the universe, it is often not possible to enumerate all the users in the universe. Therefore, it is not practical to get the exact mapping between users and positions in the bit vector. These observations suggest the need for a new technique that (1) could limit the dimension of the bit vectors (to reduce variance) and, (2) do not require a pre-defined universe of users. These requirements naturally lead us to Bloom filters [11].

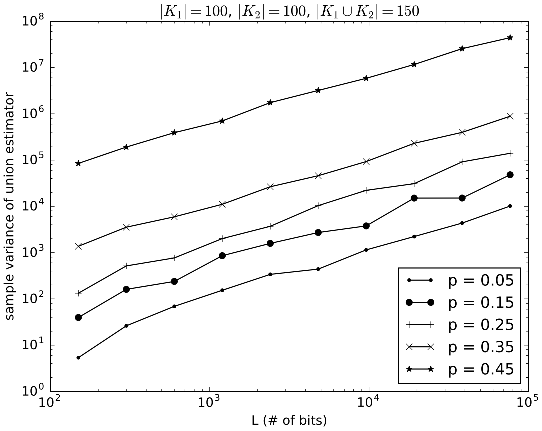

Fig. 1. Sample variance of the union estimator over 100 flipping runs. Note the log-scale. Each of the five lines is (approximately) parallel to the $x = y$ line suggesting linear relationship between the variance and L.

IV. Optimization with Bloom Filters

A. Share Set with Bloom filters

Though in general a Bloom filter could use multiple hash functions, in our approach we only consider Bloom filters with a single hash function. Given a set K, denote its Bloom filter as $BF(K, L)$, where L is the size of the Bloom filter. To ensure differential privacy, a data owner randomly flips each bit in $BF(K, L)$ with probability p, just as described before in section III-A, and then share the perturbed Bloom filter with the server. It is easy to show that this process satisfies differential privacy. The proof is very similar to that in Section III-A, which we omit here.

B. Estimate Set Cardinality using Flipped Bloom Filter

Consider set K with cardinality $|K|$ mapped to a Bloom filter $BF(K, L)$ with size L, and denote with $BF_p(K, L)$ the *flipped Bloom filter* obtained by flipping each bit of $BF(K, L)$ with probability p. As before, denote by n_i the number of i's in $BF(K, L)$, and by m_i the number of i's in $BF_p(K, L)$, for $i \in \{0, 1\}$.

Through m_0 and m_1 observed from $BF_p(K, L)$, the server could estimate n_0 and n_1 using the same mean-field model as in section III-B. Hence n_0 and n_1 can be obtained using the equation (1), which is shown below again for clarity:

$$n_0 = \frac{qm_0 - pm_1}{q - p}, \quad n_1 = \frac{qm_1 - pm_0}{q - p} \tag{8}$$

Meanwhile, there is a strong relationship between the cardinality $|K|$ and n_0, the number of 0s in the non-flipped Bloom filter. For large L (say $L > 10$):

$$E(n_0) = L(1 - \frac{1}{L})^{|K|} \approx Le^{-\frac{|K|}{L}} \tag{9}$$

From (8) and (9) it follows that the set cardinality $|K|$ can be estimated as:

$$|\hat{K}| = -L \cdot log\left(\frac{qm_0 - pm_1}{L(q-p)}\right) \tag{10}$$

Due to non-linear nature of our estimator and because of the approximation in (8) the above estimator is inevitably biased. However, we empirically observe that the bias is very small: for every data point presented Section V it is several times to several orders of magnitude smaller than the standard deviation.

C. Estimate Union/Intersection Cardinality using Flipped Bloom Filter

Given the Bloom filters and their corresponding flipped vectors of two sets K_1 and K_2, we denote n_{ij}, m_{ij} similarly as in section III-C. Using the same logic as in Section III-C above we can derive n's from m's using (3)

Similarly to previous subsection the number of 00 positions in non-flipped Bloom filters $BF(K_1, L)$ and $BF(K_2, L)$ is determined by the size of the union of K_1 and K_2:

$$E(n_{00}) = L(1 - \frac{1}{L})^{|K_1 \cup K_2|} \approx Le^{-\frac{|K_1 \cup K_2|}{L}}, \tag{11}$$

which allows us to estimate the cardinality of the union as:

$$\begin{aligned} \hat{\cup}_0 &= |K_1 \cup K_2| \\ &= -L \cdot log\left(\frac{q^2 m_{00} - pqm_{01} - pqm_{10} + p^2 m_{11}}{L(q-p)^2}\right). \end{aligned} \tag{12}$$

In addition to the above estimator we can also derive an estimator of the cardinality of the union from n_{01} and n_{10}. Namely, for n_{01}, the number of positions s in which $BF(K_1, L)[s] = 0$ and $BF(K_2, L)[s] = 1$, $E(n_{01}) = L(e^{-\frac{|K_1|}{L}} - e^{-\frac{|K_1 \cup K_2|}{L}})$. Thus the cardinality of the union can also be estimated with:

$$\begin{aligned} &\hat{\cup}_1 = \\ &- L \cdot log\left(e^{-\frac{|K_2|}{L}} - \frac{-pqm_{00} + p^2 m_{01} + q^2 m_{10} - pqm_{11}}{L(q-p)^2}\right). \end{aligned} \tag{13}$$

Using the same logic on n_{10} instead of n_{01} we get the third estimator of the union:

$$\begin{aligned} &\hat{\cup}_2 = \\ &- L \cdot log\left(e^{-\frac{|K_1|}{L}} - \frac{-pqm_{00} + q^2 m_{01} + p^2 m_{10} - pqm_{11}}{L(q-p)^2}\right). \end{aligned} \tag{14}$$

Our cardinality of the union estimator is then:

$$\hat{\cup} = \frac{\hat{\cup}_0 + \hat{\cup}_1 + \hat{\cup}_2}{3}. \tag{15}$$

An important remark here is that when set cardinality is not explicitly shared with the server but estimated using (10), the three above estimators are exactly the same and thus: $\hat{\cup}_0 = \hat{\cup}_1 = \hat{\cup}_2 = \hat{\cup}$. However, when the data owners devote a slice of their privacy budget for sharing the set cardinality with the server these three estimators are different and hence averaging them leads to a smaller variance than using each one of them independently. In Section IV-E we discuss the budget splitting. Note also that the three estimators are not independent and we do not know their distributions a priory, hence we use simple average, rather than more sophisticated methods like the one described in Lemma IV.1 for independent random variables.

Estimating the cardinality of the intersection or set difference would follow directly from the basic equations of the set arithmetic:

$$|K_1 \cap K_2| = |K_1| + |K_2| - |K_1 \cup K_2|. \tag{16}$$

$$|K_1 \backslash K_2| = |K_1 \cup K_2| - |K_2|. \tag{17}$$

Most of our paper is focused on understanding of cardinality of union. Qualitatively, most of our findings for union would directly apply to set intersection and difference and we omit them to preserve the flow of exposition.

Remark. In this paper our focus is on set operations between two sets. We strongly believe that our approach outlined above can be generalized to set operations involving more than two sets, albeit the systems of linear equations to be solved would grow exponentially with number of sets involved. Namely for $k = 2$ sets, we have linear system of $2^k = 4$ equations to estimate n_{ij}. With $k > 2$ we would have linear system with 2^k equations to estimate appropriate n's. We leave formal analysis of set operations with more than two sets for future work.

D. Variance Analysis

For deriving the variance in the case of Bloom filter we will use a well known approximation of the variance of a differentiable function of a random variable which follows directly from the Taylor expansions for the moments of functions of random variables [9]:

$$Var(f(X)) \approx (f'(E(X))^2 Var(X). \tag{18}$$

When we apply the above approximation on (10) with $f(\cdot) = log(\cdot)$ we obtain:

$$Var(|\hat{K}|) \approx L^2(\frac{L}{n_0})^2 Var(\frac{n_0}{L}) = \frac{L^2}{E(n_0)^2} Var(n_0) =$$

$$\frac{L^2}{E(n_0)^2} Var(\frac{m_0}{q-p} - \frac{pL}{q-p}) = \frac{L^2}{(q-p)^2 E(n_0)^2} Var(m_0) =$$

$$\frac{L^2}{(q-p)^2 L^2 e^{-2\frac{|K|}{L}}} Lpq = \frac{Lpq}{(q-p)^2 e^{-2\frac{|K|}{L}}}$$

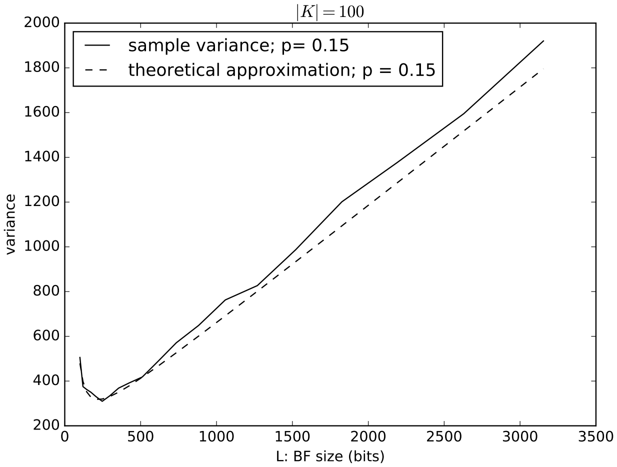

Fig. 2. Actual variance on 100 flipping runs vs. analytical approximation.

In Figure 2 we depict the above analytical approximation for the set with $|K| = 100$ elements as well as empiric standard deviation calculated on a sample of 100 flipping runs and observe a close match between the observed variance and the theoretical approximation.

We observe an interesting tradeoff between L and the above variance. On the one hand, in general the smaller L is, the less the variance, which confirms our prior intuition. Since L is adjustable when using Bloom filters, we could set L to be much smaller than the size of the universe to reduce variance. On the other hand, L cannot be too small. Otherwise, the term $e^{-2\frac{|K|}{L}}$ in the denominator would push the variance higher. In other words, there appear to be an optimal L to minimize the variance for given K. A simple calculus exercise of minimizing $f(t) = te^{2|K|/t}$ leads to $L = 2|K|$ as the one which optimizes the variance of the set cardinality estimation. It is interesting that the optimal L does not depend on the flipping probability p (and thus the privacy budget) but is only determined by the size of the set.

The exact variance of the estimators of union and intersection using Bloom filters is hard to obtain. Still, if we approximate the variance of (5) with αL then the variance of $\hat{\mathrm{U}}$ would be

$$Var(\hat{\mathrm{U}}) \approx \frac{\alpha L}{e^{-2\frac{|K_1 \cup K_2|}{L}}},$$

which is minimized for $L = 2|K_1 \cup K_2|$, which our empirical examination validates (see Section V).

The above analysis also brings an interesting dilemma: to minimize the variance of the estimation of $|K_1 \cup K_2|$, we need to know $|K_1 \cup K_2|$ in the first place to set the optimal L, which apparently cannot be done. Instead, in practice, we could use 2 times maximum expected cardinality of the set as a rough estimation to set a reasonable L. If the distribution of the possible set sizes is known a priory an interesting optimization can be formulated, and solved, on how to choose the optimal L, but that line of work is out of scope of the present paper.

E. Privacy Budget Splitting

In Section IV-C we described three estimators for the cardinality of the union of two sets, given flipped bloom filters. Two of them (13) and (14) actually rely on the individual set cardinality which could be estimated too from their flipped BFs. However, when using the cardinality estimates (10) it is not difficult to see that (13) and (14) are exactly identical to (12). Hence, in this section we explore the possibility of splitting the privacy budget ϵ into two parts ϵ_1 and ϵ_2: the first one devoted to obfuscating the Bloom filter, and the other for reporting the set cardinality using the Laplace mechanism with the goal of obtaining multiple estimates of the union cardinality which averaged lead to the lower variance.

For given ϵ_1, ϵ_2, set K, and Bloom filter size L, one can estimate the set cardinality using (10) as well using the declared cardinality of K perturbed by the Laplace noise $Lap(1/\epsilon_2)$. We know from Section IV-D that the variance of the former can be approximated with $\frac{Lpq}{(q-p)^2 e^{-2\frac{|K|}{L}}}$, while the variance of the latter is $2/\epsilon_2^2$. Here $q = e_1^\epsilon/(1 + e_1^\epsilon)$ and $p = 1 - q$.

Lemma IV.1. For independent random variables $X_1, \ldots, X_r$, and non-negative $w_1, \ldots, w_r$ which sum to 1 we have:

$$Var(\sum_{i=1}^{r} w_i X_i) \geq \frac{1}{\sum_{i=1}^{r} \frac{1}{Var(X_i)}} = A$$

with equality holding for $w_i = \frac{1}{Var(X_i)}/A$.

Proof. Proof follows directly from the Cauchy-Schwarz inequality. Namely,

$$Var(\sum_{i=1}^{r} w_i X_i)\frac{1}{A} = \sum_{i=1}^{r} w_i^2 Var(X_i) \sum_{i=1}^{r} \frac{1}{Var(X_i)} \geq$$

$$\geq (\sum_{i=1}^{r} w_i)^2 = 1,$$

with the equality holding if and only if w_i are inversely proportional to $Var(X_i)$. □

When applying the Lemma IV.1 on our two estimators we conclude that the variance of the weighed sum of two estimators for $|K|$ would be bounded from below by:

$$h(\epsilon_1) = \frac{1}{\frac{2}{(\epsilon - \epsilon_1)^2} + \frac{(e^{\epsilon_1} - 1)^2}{Le^{\epsilon_1 - \frac{2|K|}{L}}}}$$

However, for a given scenario of two different sets K_1 and K_2, privacy budget and BF size, it is generally difficult to analytically derive the optimal split of the budget. In the next section we empirically study the problem of budget splitting and show that it can reduce the standard deviation of the estimators for up to 20%.

F. Alaggan et al.'s Approaach [4]

In this section, we provide a brief description of a similar approach proposed by Alaggan et al. [4] to estimate set intersections. Further, we show why their estimations are quite coarse and relative errors of estimation is much worse than ours. Specifically, similar to the traffic monitoring application mentioned in the section I, they propose to use flipped Bloom filters to publish differentially private summaries of call detail records from different cellular towers in order to estimate the intersection of such summaries. The estimation could in turn be used to identify user movement patterns without violating individual user privacy.

They extend the set intersection estimation on two (non-flipped) Bloom filters method proposed by Broder et al. [13] to flipped Bloom filters. Specifically, they provide a relationship between the inner product of Bloom filters and the cardinality of the set intersection of the two sets encoded in those Bloom filters. Alaggan et al. proposes a similar estimator for flipped Bloom filters. Now we summarize their main results below.

Let k be the number of hash functions employed by Bloom filters. Note that we replace the flipping probability p with q in their original equations. We believe that the original equations are in error and we in fact observe unimaginably huge relative errors using the original estimation. The variables ϕ, C_1, C_2 and C_3, are defined as follows:

$$\phi = 1 - \frac{1}{L}$$

$$C_1 = (qp - p^2)(\phi^{k|K_1|} + \phi^{k|K_2|}) - p^2$$

$$C_2 = k \ln \phi$$

$$C_3 = \frac{\ln((q-p)^2)}{k \ln \phi} + |K_1| + |K_2|$$

The estimator proposed by Alaggan et al. [4] for the set intersection $|K_1 \cap K_2|$ is the following expression:

$$|K_1 \hat{\cap} K_2| = -\frac{\ln(\frac{m_{11}}{L} - C_1)}{C_2} + C_3$$

The estimator has a quite large bias term:

$$\frac{Var(m_{11})}{2C_2(E[m_{11}] - C_1 L)^2}$$

In Section V-D we demonstrate that the error of the above estimator can be substantial (sometimes orders of magnitude larger than the error of the estimator we propose in this paper) and is very sensitive to the choice of L.

V. Empirical Evaluation

In this section we empirically evaluate our approach and the impact different variables have on the overall accuracy of the union estimator. Hereafter we focus on set union and note that most of qualitative findings would directly apply to set intersection/difference and are omitted here.

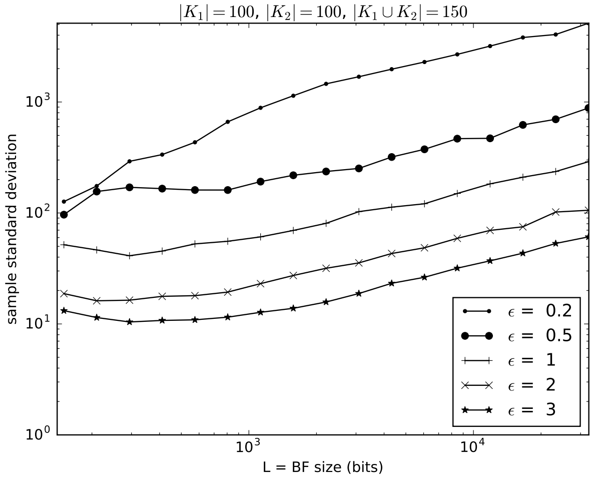

Fig. 3. Sample standard deviation of the union estimator without budget split on 100 flipping runs.

A. Accuracy of the Basic Union Estimator

We start the section by examining how different budgets, ϵ, and Bloom filter sizes, L, affect the accuracy of the basic estimator of the set union cardinality (10) in the case where all privacy budget is devoted to sharing the flipped bloom filter (and thus no explicit information about set cardinality is shared with the data collector). In Figure 3, we consider two sets of size 100, whose union has 150 elements. We vary L from 100 to 30K, and ϵ in the range from 0.2 to 3. For each pair L and ϵ we take 100 runs of bit flipping and report the sample standard deviation averaged over these 100 runs. We can observe several interesting properties of the basic estimators. First, in case when privacy budget is large ($\epsilon \geq 2$), one can expect coefficient of variations (CoV is a standard metric of dispersion defined as the ratio of the standard deviation and the mean) of 10% or less for a range of BF sizes; for small privacy budgets $\epsilon \leq 0.5$ the coefficient of variations are very large ($> 100\%$) and our approach is unlikely to be useful for the applications which require low errors of the estimators. Secondly, for given ϵ the observed coefficient of variation is not a monotone function of the BF size and achieves minimum when $L \approx 2|K_1 \cup K_2|$; this non-monotonicity is not observed for $\epsilon \leq 0.5$ since in that case the 100 runs do not allow for faithful estimate of the standard deviation. Finally, for $\epsilon \geq 1$ the coefficient of variations do not appear to be very sensitive to the choice of L, as long as L is within the order of magnitude of the $\max(|K_1|, |K_2|)$, or in the range of $[100, 1000]$ in our case.

B. Coefficient of Variation and the Set Cardinality

As we discussed earlier the variance of the estimators grow linearly with L (for large L), the size of the Bloom filter. However, from the previous paragraph we know that for a given range of set cardinalities we want to estimate, choosing L to be within one order of magnitude greater than the cardinality

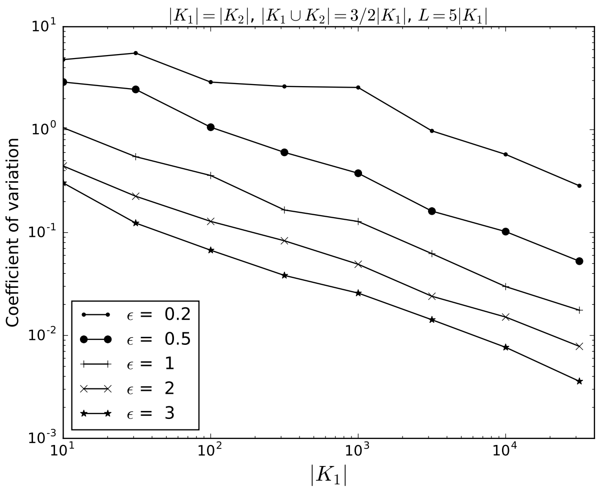

Fig. 4. Sample coefficient of variation of the union estimator (without budget split) on 100 flipping runs for a range of ϵ and set cardinalities. Bloom filter size set proportional to the set cardinality.

of the larger of two sets, the variance of the union estimator is relatively insensitive to the choice of L while once L is larger than that variance, it starts to grow. Hence, in this section we ask the following question: if K_1 and K_2 are of similar sizes, and L is within one order of magnitude greater than their cardinality how does the coefficient of variation of the union estimator (10) vary with the cardinality of the sets? In Figure 4, we report sample coefficient of variation of the union estimator over 100 flipping runs for a range of set cardinality and privacy budgets. Similar to the previous paragraph, we take sets K_1 and K_2 to have the same cardinality, overlapping over half of the elements, and choose $L = 5|K_1|$. From Figure 4, we can observe that for small set sizes, the coefficient of variations (which closely approximates expected error) are rather large even for generous privacy budgets. However, for larger sets, the CoV decay as $1/\sqrt{|K_1|}$ and can be reasonably small even for modest privacy budgets.

C. *Effect of Budget Split*

In Section IV-E, we discussed the idea of budget splitting in order to boost the accuracy of the union estimators. Namely, if we share (perturbed) information about the cardinality of the underlying set along with the flipped BF, we can use it to get additional estimates of the cardinality of the union and hence potentially reduce the variance. We choose three different set cardinalities spanning 3 orders of magnitude, set $L = 5|K_1|$ and choose privacy budget $\epsilon = 1$. We vary the ϵ_2, privacy budget devoted to sharing the set cardinality information, in the range of $[0, 0.2]$[1]. Again, we evaluate the coefficient of variation for the combined estimator and report the results in Figure 5. We observe that allocating relatively small privacy budget for sharing the set cardinality can reduce the variance of the set estimator from 7%, for $|K_1| = 100$ (corresponding to the decrease of CoV from 0.30 to 0.28) to 20% for $|K_1| = 10000$ (corresponding to the decrease of CoV from 0.028 to 0.023). It is not surprising that the gains of budget splitting are larger for larger sets. Namely, for a given ϵ_2, Laplace noise $Lap(1/\epsilon_2)$ has variance independent of $|K_1|$ and offers smaller relative errors for larger $|K_1|$ and consequently greater improvement of budget splitting.

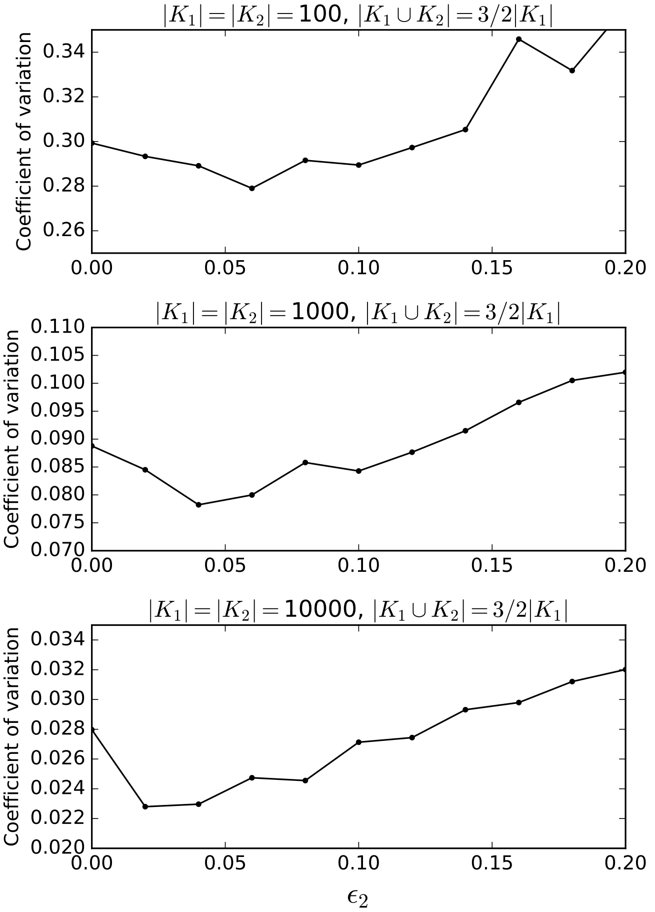

Fig. 5. Sample standard deviation of the union estimator with budget split on 10000 flipping runs for $\epsilon = 1$ and $\epsilon_2 \in [0, 0.2]$. Three choices for $|K_1|$ spanning 3 orders of magnitude.

Remark. Note that variance of the intersection and union estimators are of similar magnitude and hence relative errors of the union estimator are smaller than of the intersection estimator. Additionally, for very small (or empty) intersections the intersection estimator would inevitable yield large relative errors.

D. *Comparison with Alaggan et al.'s Approaach [4]*

We provide a brief description of Alaggan et al.'s approach and their key results in Section IV-F. In this section, we empirically evaluate their approach compared to ours. Their estimation of set intersection cardinalities is based on an approximation of set sizes when multiple hash functions are used in a Bloom filter, which, unfortunately turned out to be a quite coarse estimation. For example, consider the setting of two

[1] Note that $\epsilon_2 = 0$, translates to the case where only flipped BF is shared with the server.

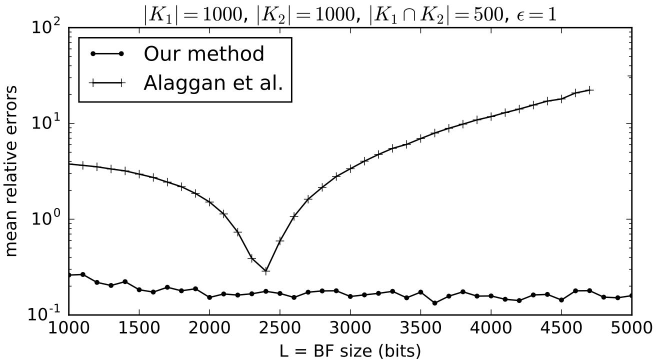

Fig. 6. Mean relative error of the intersection estimator without budget split on 100 flipping runs for $\epsilon = 1$. $|K_1| = |K_2| = 1000$, $|K_1 \cap K_2| = 500$. Comparison of our method and Alaggan et al.

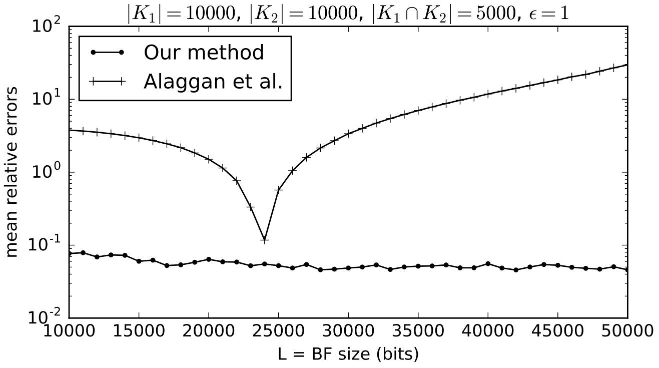

Fig. 7. Mean relative error of the intersection estimator without budget split on 100 flipping runs for $\epsilon = 1$. $|K_1| = |K_2| = 10000$, $|K_1 \cap K_2| = 5000$. Comparison of our method and Alaggan et al.

sets, each with 1000 elements and with intersections of 500 elements with privacy budget $\epsilon = 1$. Our method consistently estimates the cardinality of intersection with relative error of around 16% for a wide range of BF sizes. On the other hand the method from Alaggan et al. [4] is extremely sensitive on the choice of the BF size and produces errors in the range of 30% to over 3000%; see Figure 6. Similarly, for sets of with sizes $|K_1| = |K_2| = 10000$, and $|K_1 \cap K_2| = 5000$ the intersection cardinality error of our method is in the range of $4 - 6\%$ while Alaggan et al. method results in errors which are 3 to 600 times larger; see Figure 7.

VI. Related Work

In this section, we critically analyze and compare our contributions with the previous related work. Broadly, our work is related to data aggregation with untrusted aggregators, differential privacy for private data mining, randomized response and local differential privacy, secure multiparty computation, privacy preserving set cardinality estimation over Bloom filters and dimensionality reduction techniques for privacy preservation. We analyze each of these related work areas below.

Untrusted Data Aggregators: Our work is broadly related to the category of recently explored problems [41], [44], [14], [22] where an untrusted data aggregator, i.e. server in our case, runs some algorithm over aggregated data from data owners while guaranteeing privacy of each contributing user. However, these solutions do not readily suitable for private set intersection/union cardinality computation under a stronger and formal privacy model of differential privacy, which is a main contribution of this work, due to the following limitations. Most of the previous untrusted servers perform arithmetic operations such as sum [41], [44] or frequency estimation over categorical data [22], [8]. Some previous work relies on either peer-to-peer communication or prior communications among data owners [21], [1] to carry out the protocol which we believe is not feasible in some of the example applications such as traffic monitoring and analysis we consider in this work as mentioned in Section I. Further, some of these approaches require not only expensive but extremely vulnerable key pre-distribution among data owners [28]. Thus, unlike previous work, we propose novel algorithms to estimate set intersection/union cardinality with rigorous privacy guarantees and with only one way communication from data owners to the server.

Differential Privacy for Private Data Mining (DP): The idea of perturbing the output of statistical queries in order to preserve the privacy of database records has been studied since late '70s [2]. In such approaches, the database computes a noisy version of the exact result for a given query and returns the noisy result to the querying party [31], [42]. However, it was not until recently the data privacy of such systems was analyzed under a firm theoretical foundations [16]. In fact, the rigorous notion of privacy, differential privacy, which was introduced by Dwork et. al [19], has roots in such theoretical foundations laid earlier. Differential privacy has been adopted widely in order to computing statistics over a population without revealing individual data [25]. The concept was initially proposed for a setting where a trusted server, who possesses a database of records from participants, publishes noised statistics by applying a randomized mechanism to this database. However, this approach does not work when the server is untrusted as in our setting where the privacy of the users' data must be preserved from the server. A notable exception to this line of work is proposed by McGregor et. al [33] where they investigate the error bounds on two party differentially private hamming distance computations. Similar to Secure Multiparty Computation protocols, they rely on multiple rounds of communications achieve mutual differential privacy whereas our protocols are designed to work under single round local differential privacy setting.

Randomized Response (RR) and Local Differential Privacy (LDP): Randomized response is a decades old surveying technique used for collecting statistics on sensitive topics (e.g. Are you HIV positive?) where the privacy of survey users is preserved [47]. While initial work supported binary data, more sophisticated techniques were built to deal with complex data and advanced statistics [22], [8], [39], [23]. It has in fact been shown that randomized response techniques can be reduced to ϵ-differential privacy [22] under the local model, i.e., local differential privacy (LDP) [40], [26], [17]. Under LDP, each data owner perturbs their data locally before sending it to the server to perform statistical operations on the aggregated perturbed data. Our work builds on the idea of randomized response to perturb the data locally at each data owner, but extends the technique to efficiently and accurately compute set intersection/union cardinality at the server. It should be noted that data perturbation to achieve such differential privacy is usually carried out via methods such as the Laplace mechanism [18], the Exponential mechanism [34] or the geometric mechanism [27]. However, such mechanisms are not suitable to address our problem as we are dealing with categorical values and there is no trusted third party to perform the perturbation. Thus, similar to RAPPOR [22], we adapt randomized response via bit flipping to construct differentially private Bloom filters. However, the problem settings are different: RAPPOR estimates the item frequencies whereas our approach estimates set intersection/union cardinality.

Secure Multiparty Computation (SMC): SMC allows two or more participants to compute the value of a public function using their private values as input, but without revealing their individual private values to other participants. Specific to our problem setting, privacy preserving set intersection/union cardinality has been studied under the SMC model. For example, Freedman et. al [24] proposed a private set intersection cardinality interactive protocol using encrypted polynomials. Interactive protocols are not applicable for our problem setting where one way communication is required between data owners and the server.

Set Cardinality Estimation over Bloom Filters: Prior research on private computation of set intersection and union cardinality is carried out primarily in two different ways: (1) applying cryptographic techniques on perturbed client data [46], [15] and (2) applying non-cryptographic techniques such as generalization and sharding over succinct data structures such as Bloom filters [6], [20]. The schemes based on the former approach are interactive in nature and exhibit similar issues as in SMC based protocols mentioned earlier. The schemes based on the latter approach, especially the ones utilizing Bloom filters, are attractive as they usually scale well to large data sets possessed by clients. In fact, Bloom filter based set cardinality estimation methods are extensively employed in distributed applications, especially in distributed database Bloom joins, in order to boost performance and reduce communication overhead [12]. Papapetrou et. al [38] presented a detailed analysis of probabilistic cardinality estimation over disjunctive and conjunctive Bloom filters with tight error bounds. While the statistical techniques used in their approach has some similarities, their approach does not apply to flipped Bloom filters which our approach uses to provide ϵ-differential privacy. Some recent line of work [6], [20] utilizes sharded Bloom filters to estimate set cardinality in a privacy preserving manner using a collaborative protocol among clients. In our work, we assume that data owners are unable to communicate with each other due to various reasons. Further, it is not clear what privacy guarantees these sharding based Bloom filters provide. Thus, such approaches are not suitable to solve the problem we are addressing in this work.

Dimensionality Reduction for Privacy: Dimensionality reduction techniques have been used in the past in order to perform privacy preserving computations while providing acceptable utility [35], [32], [29], [8]. While our motivation behind using Bloom filters is to provide a succinct representation of sets with goal to reduce the variance of cardinality estimators, we believe one could further improve privacy guarantees by careful selection of Bloom filter parameters [10]. We leave this analysis as future work.

VII. Conclusion

In this paper we study the problem of estimating cardinality of set operations when the input sets are perturbed for privacy protection. We apply the random response technique to publish set data and achieve differential privacy. We show in detail how the server could estimate set union/intersection cardinality from the perturbed vectors. Our theoretical analysis reveals the significant negative impact of the high-dimension of bit vectors on estimation accuracy. As an optimization technique, we leverage Bloom filters to control the dimension of the published data, and analyze formally how to select the optimal Bloom filter size to minimize the variance of cardinality estimators. We further consider another optimization technique that spends some of the privacy budget to report noisy cardinality of each input set. Our empirical evaluation shows that the proposed technique could achieve quite accurate cardinality estimation of set operations when the Bloom filter size and privacy budget split are set appropriately. A natural venue built on our technique is to tailor it for specific application domains (e.g., traffic monitoring systems). Another interesting problem is to investigate how to conduct more complex data analysis tasks (e.g., similarity measure and community discovery) utilizing our technique as a basic building block.

References

[1] Aydin Abadi, Sotirios Terzis, and Changyu Dong. O-psi: Delegated private set intersection on outsourced datasets. In Hannes Federrath and Dieter Gollmann, editors, *Proceedings of the 30th IFIP TC 11 International Conference on ICT Systems Security and Privacy Protection*, pages 3–17. Springer International Publishing, 2015.

[2] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Survey*, 21:515–556, 1989.

[3] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. Blip: Non-interactive differentially-private similarity computation on bloom filters. In *Proceedings of the 14th International Conference on Stabilization, Safety, and Security of Distributed Systems*, pages 202–216. Springer-Verlag, 2012.

[4] Mohammad Alaggan, Sébastien Gambs, Stan Matwin, and Mohammed Tuhin. Sanitization of call detail records via differentially-private bloom filters. In *Data and Applications Security and Privacy XXIX - 29th Annual IFIP WG 11.3 Working Conference, DBSec 2015, Fairfax, VA, USA, July 13-15, 2015, Proceedings*, pages 223–230, 2015.

[5] W. Albazrqaoe, J. Huang, and G. Xing. Practical bluetooth traffic sniffing: Systems and privacy implications. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 333–345. ACM, 2016.

[6] V. G. Ashok and R. Mukkamala. A scalable and efficient privacy preserving global itemset support approximation using bloom filters. In *Proceedings of the 28th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy*, pages 382–389. Springer-Verlag New York, Inc., 2014.

[7] J. Barceló, L. Montero, L. Marqués, and C. Carmona. Travel time forecasting and dynamic origin-destination estimation for freeways based on bluetooth traffic monitoring. *Transportation Research Record: Journal of the Transportation Research Board*, (2175):19–27, 2010.

[8] R. Bassily and A. Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, pages 127–135. ACM, 2015.

[9] H. Benaroya, S. M. Han, and M. Nagurka. *Probabilistic Models for Dynamical Systems*. CRC Press, 2013.

[10] G. Bianchi, L. Bracciale, and P. Loreti. "better than nothing" privacy with bloom filters: To what extent? In *Proceedings of the 2012 International Conference on Privacy in Statistical Databases*, pages 348–363. Springer-Verlag, 2012.

[11] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[12] A. Broder, M. Mitzenmacher, and A. Mitzenmacher. Network applications of bloom filters: A survey. In *Internet Mathematics*, pages 636–646, 2002.

[13] Andrei Broder, Michael Mitzenmacher, and Andrei Broder I Michael Mitzenmacher. Network applications of bloom filters: A survey. In *Internet Mathematics*, pages 636–646, 2002.

[14] T.-H. H. Chan, M. Li, E. Shi, and W. Xu. Differentially private continual monitoring of heavy hitters from distributed streams. In *Proceedings of the 12th International Conference on Privacy Enhancing Technologies*, pages 140–159. Springer-Verlag, 2012.

[15] E. De Cristofaro, P. Gasti, and G. Tsudik. Fast and private computation of cardinality of set intersection and union. In *Proceedings of the 11th International Conference on Cryptology and Network Security*, pages 218–231. Springer Berlin Heidelberg, 2012.

[16] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 202–210. ACM, 2003.

[17] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE Computer Society, 2013.

[18] C. Dwork. Differential privacy: A survey of results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer-Verlag, 2008.

[19] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, pages 265–284. Springer-Verlag, 2006.

[20] R. Egert, M. Fischlin, D. Gens, S. Jacob, M. Senker, and J. Tillmanns. Privately computing set-union and set-intersection cardinality via bloom filters. In *Proceedings of 20th Australian Conference on Information Security and Privacy*, pages 413–430. Springer International Publishing, 2015.

[21] Fabienne Eigner, Aniket Kate, Matteo Maffei, Francesca Pampaloni, and Ivan Pryvalov. Differentially private data aggregation with optimal utility. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 316–325. ACM, 2014.

[22] U. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014.

[23] G. Fanti, V. Pihur, and U. Erlingsson. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. *Proceedings on Privacy Enhancing Technologies (PoPETS)*, issue 3, 2016, 2016.

[24] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Proceedings of Advances in Cryptology - EUROCRYPT*, page 119. Springer Berlin Heidelberg, 2004.

[25] A. Friedman and A. Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–502. ACM, 2010.

[26] A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, pages 803–812. ACM, 2011.

[27] M. Hardt and K. Talwar. On the geometry of differential privacy. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, pages 705–714. ACM, 2010.

[28] W. He, X. Liu, H. V. Nguyen, K. Nahrstedt, and T. Abdelzaher. Pda: Privacy-preserving data aggregation for information collection. *ACM Transactions on Senor Networks*, 8(1):6:1–6:22, 2011.

[29] Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the johnson-lindenstrauss transform. *CoRR*, abs/1204.2606, 2012.

[30] M.-J. Lesot, M. Rifqi, and H. Benhadda. Similarity measures for binary and numerical data: a survey. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 1(1):63–84, 2008.

[31] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems*, 10:395–411, 1985.

[32] B. Liu, Y. Jiang, F. Sha, and R. Govindan. Cloud-enabled privacy-preserving collaborative learning for mobile sensing. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 57–70. ACM, 2012.

[33] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil P. Vadhan. The limits of two-party differential privacy. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:106, 2011.

[34] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103. IEEE Computer Society, 2007.

[35] D. Mir, S. Muthukrishnan, A. Nikolov, and R. N. Wright. Pan-private algorithms via statistics on sketches. In *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 37–48. ACM, 2011.

[36] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 111–125. IEEE Computer Society, 2008.

[37] Information Commisioner's Office. Wi-fi location analytics. 2016.

[38] O. Papapetrou, W. Siberski, and W. Nejdl. Cardinality estimation and dynamic length adaptation for bloom filters. *Distributed and Parallel Databases*, 28(2):119–156, 2010.

[39] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren. Heavy hitter estimation over set-valued data with local differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 192–203. ACM, 2016.

[40] S. Raskhodnikova, A. Smith, H. K. Lee, K. Nissim, and S. P. Kasiviswanathan. What can we learn privately? In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, pages 531–540. IEEE Computer Society, 2008.

[41] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. ACM, 2010.

[42] S. P. Reiss. Practical data-swapping: The first steps. *ACM Transactions on Database Systems*, 9(1):20–37, 1984.

[43] N. Schenker and T. E. Raghunathan. Combining information from multiple surveys to enhance estimation of measures of health. *Statistics in Medicine*, pages 1802–1811, 2007.

[44] E. Shi, R. Chow, T-H. H. Chan, D. Song, and E. Rieffel. Privacy-preserving aggregation of time-series data. In *Proceedings of Network and Distributed Systems Security Symposium*, 2011.

[45] L. Sweeney. Weaving technology and policy together to maintain confidentiality. *Statistics in Medicine*, pages 98–110, 1997.

[46] J. Vaidya and C. Clifton. Secure set intersection cardinality with application to association rule mining. *Journal of Computer Security*, 13(4):593–622, 2005.

[47] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60:63–69, 1965.

[48] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.