Sanitization of Call Detail Records via Differentially-Private Bloom Filters

Mohammad Alaggan, Sébastien Gambs, Stan Matwin, Mohammed Tuhin

Sanitization of Call Detail Records via Differentially-private Bloom Filters

Mohammad Alaggan[1], Sébastien Gambs[2], Stan Matwin[3] and Mohammed Tuhin[3]

[1]Helwan University, [2]Université de Rennes 1 - Inria and [3]Dalhousie University

Abstract. Publishing directly human mobility data raises serious privacy issues due to its inference potential, such as the (re-)identification of individuals. To address these issues and to foster the development of such applications in a privacy-preserving manner, we propose in this paper a novel approach in which Call Detail Records (CDRs) are summarized under the form of a differentially-private Bloom filter for the purpose of privately estimating the number of mobile service users moving from one area (region) to another in a given time frame. Our sanitization method is both time and space efficient, and ensures differential privacy while solving the shortcomings of a solution recently proposed. We also report on experiments conducted using a real life CDRs dataset, which show that our method maintains a high utility while providing strong privacy.

1 Introduction

One of the key ingredients of the digital economy of the future is the opportunity to exploit large amounts of data. In particular, Call Detail Records (CDRs) that are generated by users of mobile devices and collected by telecom operators could potentially be used for the socio-economic development and well-being of populations. For instance, such data can be used for scientific research (*e.g.*, the study of the human mobility), and also for practical objectives that can benefit the society (*e.g.*, to find the best place to build an infrastructure such as a bridge or to create a new bus line). However, learning the location of an individual is one of the greatest threats against his privacy, because it can be used to derive other personal information. For example, from the movements of an individual it is possible to infer his points of interests (such as his home and place of work) [10], to predict his past, current and future locations [9], or to conduct a de-anonymization attack [8]. Thus, it is of paramount importance to develop new methods that can mine CDRs while preserving the privacy of the individuals contained in this data. To counter these threats, we propose a novel data sanitization method based on Bloom filters [4] that produces a privacy-preserving data structure out of CDRs.

2 Preliminaries

Bloom filter. A *Bloom filter* [4] is widely used as a summary data structure originally designed for membership testing (*i.e.*, testing whether a particular

element is contained in the set considered). A Bloom filter can summarize a large dataset using linear space, is very simple to construct and inserting an element or testing its membership can be done in constant time. A Bloom filter is composed of an array of m bits and equipped with a set of k independent hash functions. An element can be inserted into a Bloom filter by passing it as input to each of the hash functions. The k outputs of the hash functions correspond to k positions of the Bloom filter, which are all set to one independently of their previous values. Testing the membership is done in a similar manner by considering that an item is contained in the filter only if the k corresponding bits are all set to 1. Due to the collisions generated by the hash functions, false positives can arise by having an element erroneously considered as being a member of the filter. The accuracy of a query to a Bloom filter depends on the number k of hash functions used, the size m of the filter as well as the number of items inserted. Disclosing directly a plain Bloom filter easily jeopardize privacy. For instance is possible for an adversary observing this Bloom filter to query exhaustively for all items to reconstruct the set encoded in this Bloom filter. Thus, it is necessary to ensure that the summary released also ensure strong privacy guarantees such as differential privacy, which we present hereafter.

Differential privacy [6] aims at providing strong privacy guarantees with respect to the input of some computation by randomizing the output of this computation. In our setting, the input of the computation is a summary of the CDRs observed by a cellular antenna and the randomized output is a perturbed version of this summary (*i.e.*, a Bloom filter). Two databases $\mathbf{x}$ and $\mathbf{x}'$ are said to *differ in at most one element*, or equivalently to be *neighbors*, if they are equal except for possibly one entry.

Definition 1 (Differential privacy [7]). *A randomized function $\mathcal{F} : \mathcal{D}^n \to \mathcal{D}^n$ is ϵ-differentially private, if for all neighbor databases $\mathbf{x}, \mathbf{x}' \in \mathcal{D}^n$ and for all $\mathbf{t} \in \mathcal{D}^n$:*

$$\Pr[\mathcal{F}(\mathbf{x}) = \mathbf{t}] \leqslant e^{\epsilon} \cdot \Pr[\mathcal{F}(\mathbf{x}') = \mathbf{t}] \ .$$

This probability is taken over all the coin tosses of $\mathcal{F}$ and e is the base of the natural logarithm.

The privacy parameter ϵ is public and may take different values depending on the application (for instance it could be 0.1, 0.25, 1.5, 5 or even more) [11]. The smaller the value of ϵ, the higher the privacy but also as a consequence the higher the impact on the utility of the resulting output. Differential privacy is known to compose well. In particular, if two functions F_1 and F_2, that are respectively ϵ_1 and ϵ_2 differentially private, then the function G combining their outputs $G(\mathbf{x}) = (F_1(\mathbf{x}), F_2(\mathbf{x}))$ is $(\epsilon_1 + \epsilon_2)$ differentially private. For a computation applying several differentially private algorithms on the same dataset, the sum of their differential privacy parameters is called the *privacy budget*.

Originally, differential privacy was developed within the context of private data analysis. The main guarantee is that if a differentially private mechanism is applied on a dataset composed of the personal data of individuals, no output would become significantly more (or less) probable whether or not a *single*

participant contributes to the data set. This means that an adversary observing the output of the mechanism only gains negligible information about the presence (or absence) of a particular individual in the database. This statement is a statistical property about the behavior of the mechanism (*i.e.*, function) and holds independently of the auxiliary knowledge that the adversary might have gathered. In our setting, the database that we want to protect is a CDRs dataset and the objective of a differentially private mechanism is to hide the presence or absence of a particular user in these CDRs.

To compute privately the similarity between profiles in a social platform, Alaggan, Gambs and Kermarrec [1] have introduced a privacy-preserving summary technique called BLIP (for *BLoom-then-flIP*). In this context, the profile of each user is represented compactly using a Bloom filter and the main objective of BLIP is to prevent an adversary with unlimited computational power from learning the presence or absence of an item in the profile of a user by observing the Bloom filter representation of this profile. BLIP ensures ϵ-differential privacy [6] by flipping each bit of Bloom filter with some probability before publishing it. BLIP has the advantage of having the same communication cost as a plain Bloom filter, while guaranteeing privacy at the expense of a slight decrease of utility. We use BLIP as a fundamental building block in our approach.

3 Mining CDRs via Differentially-private Summaries

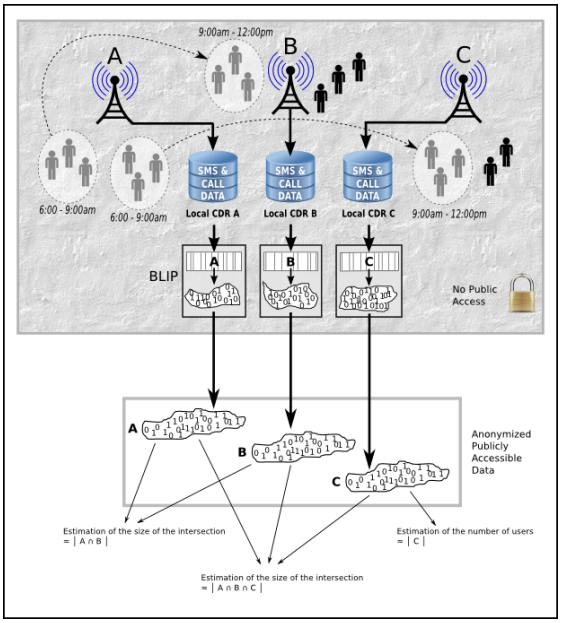

Fig. 1. lllustration of the system model.

System model. Our main objective is to perform data mining operations on differentially-private summaries learnt directly from CDRs. For instance, we would like to be able to count the number of distinct users represented in a particular summary or to compute the size of the intersection between two summaries. To realize this, we will leverage the BLIP mechanism described in the previous section. Our system model is illustrated in Figure 1. More precisely, we consider as our model a network composed of a large number of cellular antennas that are responsible for recording the CDRs related to their neighborhood (basically the events generated by calls or text messages sent or received in their corresponding region). Each of these cellular antennas publishes at a regular interval (*e.g.*, every 6 hours or each day) a summary of the users it has seen during this period. This summary takes the form of a Bloom filter in which each element inserted is actually the identifier of a user associated with a CDR. One of the advantages of using a Bloom filter is that even if a user is inserted several times (*e.g.*, if he has received or made several calls during the same period), his impact on the Bloom filter is the same. In addition, Bloom filter can be updated incrementally and in an online manner. Once the end of the period is reached, all cellular antennas BLIPed their summaries before publishing them and then erase their memories.

Estimating the size of the intersection between two summaries. Broder, Mitzenmacher and Mitzenmacher [5] described a method to approximate the intersection of two sets, S_1 and S_2, given their Bloom filter representation, B_1 and B_2. Basically, they provided a relationship between the inner product of two Bloom filters and the cardinality of the set intersection of the two sets encoded in those Bloom filters. This expected value for the inner product $\sum_i B_1[i] \times B_2[i]$ corresponds to the probability that a particular bucket is simultaneously set to one in both Bloom filters, multiplied by the total number of buckets: $m \times \Pr\left[B_1[i] = 1 \wedge B_2[i] = 1\right]$. We extended their method to the case in which the bits of both Bloom filters are flipped with some probability (such as in BLIP) prior to computing the inner product, which enables to estimate the cardinality of set intersection in a privacy-preserving way.

Our main result is summarized by the following theorem.

Theorem 1 (Size of the intersection of two sets). *Given two sets S_1 and S_2 with respective cardinalities n_1 and n_2, and their flipped Bloom filters B_1 and B_2 whose bits were flipped independently with probability p, let Q be the inner product of B_1 and B_2. Let also*

$$g(x) = -\ln(x/m - C_1)/C_2 + C_3 \ , \tag{1}$$

in which $q = 1 - p$, k is the number of hash functions employed by the Bloom filter, m is its number of bits, $\phi = 1 - 1/m$, $C_1 = (pq - q^2)(\phi^{kn_1} + \phi^{kn_2}) - q^2$, $C_2 = k \ln \phi$ and $C_3 = \ln((p-q)^2)/k \ln \phi + n_1 + n_2$. Then, $W = g(Q)$ is an estimator for $|S_1 \cap S_2|$ with expected value:

$$\mathbb{E}[W] \approx |S_1 \cap S_2| + \mathrm{var}(Q)/(2C_2(\mathbb{E}[Q] - C_1 m)^2) \ . \tag{2}$$

Proof. Consider $\mathbb{E}[W] = \mathbb{E}[g(Q)]$. The expectation of the second degree Taylor expansion (*i.e.,* truncated after the third term[1]) around $\mathbb{E}[Q]$ of $g(Q)$ is

$$g(\mathbb{E}[Q]) + \text{var}(Q)/(2C_2(\mathbb{E}[Q] - C_1 m)^2) \ . \tag{3}$$

We further detail the bias $\text{var}(Q)/(2C_2(\mathbb{E}[Q] - C_1 m)^2)$ later.

We now compute $\mathbb{E}[Q]$ and prove that $g(\mathbb{E}[Q]) = |S_1 \cap S_2|$. Consider two standard unflipped Bloom filter representations of two sets S_1 and S_2. The j-th bit of both Bloom filters will be set to 1 simultaneously if it was set by an element in $S_1 \cap S_2$ or by some element in $S_1 - S_2$ and a different element in $S_2 - S_1$. Consider the probability space of the choice of the hash functions used and let A_j be the event that the j-th bit is set in the first Bloom filter by an element of $S_1 - S_2$, B_j be the event that the j-th bit is set in the second Bloom filter by an element of $S_2 - S_1$, and C_j as the event that the j-th bit in both Bloom filters was set by an element in $S_1 \cap S_2$. The event that the j-th bit is set in both Bloom filters is $(A_j \cap B_j) \cup C_j$ and its probability is [5]:

$$\Pr[C_j] + (1 - \Pr[C_j]) \times \Pr[A_j] \times \Pr[B_j] = \quad 1 - \phi^{kn_1} - \phi^{kn_2} + \phi^{k(n_1+n_2-|S_1 \cap S_2|)} \ .$$

For the case of flipped Bloom filters let $A = 1 - \Pr[A_j], B = 1 - \Pr[B_j]$, and $C = 1 - \Pr[C_j]$, and we obtain

$$\Pr[C_j] + (1 - \Pr[C_j]) \times \Pr[A_j] \ \times Pr[B_j] = \quad (1 - C) + C(1 - A)(1 - B) \ .$$

For a bit to be set to 1 in the flipped Bloom filter it either had to be 1 in the unflipped Bloom filter and remained unchanged despite the probabilistic flipping or the converse. Therefore, it is easy to verify that the probability that the j-th bit is set to 1 is both *flipped* Bloom filters is

$$q^2(1 - C) + C\big[q^2(1 - A)(1 - B) + pqA(1 - B) + \quad pqB(1 - A) + p^2AB\big] \ ,$$

which is equal to

$$(pq - q^2)(\phi^{kn_1} + \phi^{kn_2}) + q^2 + (p - q)^2\phi^{k(n_1+n_2-|S_1 \cap S_2|)} \ . \tag{4}$$

Now, considering that the inner product Q between two flipped Bloom filters is the sum of m binary random variables, each of which has the probability of being 1 given by Equation (4). One can easily verify by simple algebraic manipulation that $g(\mathbb{E}[Q]) = |S_1 \cap S_2|$. By independence, we also have that $\mathbb{E}[Q]$ itself is the same as (4) multiplied by m. Hence, the result follows. □

In the proof of Theorem 1, we truncated the Taylor expansion after the third term, which naturally introduces an error. However, we demonstrate that for standard values of the parameters, the error is negligible. For instance consider the following standard values for the parameters: $\epsilon = 3$, $m = 187500$, $k = 2$ and $n_1 = n_2 = 50000$. With probability at least 99.9%, when Q is in the range

[1] We discuss later the error introduced by this truncation.

$\mathbb{E}[Q] \pm 3\sqrt{m}$ [1], the absolute error introduced by the truncating beyond the third term is at most 0.003 for all values of $|S_1 \cap S_2|$. This error is clearly negligible compared to $|S_1 \cap S_2|$. Moreover, for the same settings, the ratio of the fourth term to the third term is 0.002, which justifies its truncation.

The bias of the estimator as described in Equation (2) can be bounded using the Bhatia-Davis inequality [3]: $\text{var}(Q) \leq (m - \mathbb{E}[Q])\mathbb{E}[Q]$. Using this bound, the absolute ratio of the bias

$$\text{var}(Q)/(2C_2(\mathbb{E}[Q] - C_1 m)^2) \tag{5}$$

to the set intersection $|S_1 \cap S_2|$ is at most 30% for $|S_1 \cap S_2| \leq 12000$, using the same set of values considered in the previous paragraph. We will see in the next section that this value represents a Mean Relative Error (MRE) of 0.3, which is very small.

Counting the number of users in a summary. The computation of the intersection requires knowledge of n_1 and n_2, the true cardinalities of the two sets under consideration. One possibility is to release directly the set sizes by applying the ϵ-differentially private Laplacian mechanism [7] with error $O(1)$. In that case, the total privacy budget will be 2ϵ instead of ϵ. Alternatively, these cardinalities can be estimated directly from the flipped Bloom filters with no cost to the privacy budget using the method introduced recently by Balu, Furon and Gambs in [2]. This method defines a probabilistic relation between the number of items in a set and the number of bits in its Bloom filter, before extending this relation to the flipped Bloom filter. In a nutshell, given an unflipped Bloom filter with m bits, k hash functions and a set of c items, the probability[2] that a particular bit is set to 1 is $\pi_c = 1 - \phi^{ck}$, which corresponds to the probability that at least one hash function of at least one item chooses this bit. Note that this probability is not known in advance since c is precisely the quantity that we want to estimate. This relationship extends to a Bloom filter flipped with probability p, obtaining $\widetilde{\pi}_c = (1-p)\pi_c + p(1-\pi_c)$, the probability that a bit is set to 1 in the flipped Bloom filter. The parameter p is known since ϵ is public and $p = 1/(1 + \exp(\epsilon/k))$, thus the only unknown value is c, the size of the set encoded in the Bloom filter. Their method is based on the observation that the ratio of bits set to 1 in the flipped Bloom filter (*i.e.*, its Hamming weight) to the total number of bits m is an unbiased estimator for $\widetilde{\pi}_c$. Since this ratio can be computed directly from the released flipped Bloom filter, the equation for $\widetilde{\pi}_c$ can be used to derive a value for π_c, from which an estimate for c can be obtained.

4 Experimental Results and Conclusion

Experimental setting. In this section, we report on the results obtained by applying our method on a real dataset from a telecom operator. This large dataset includes coarse-grained phone call data for users in a city divided into a number of neighborhoods. More precisely for each call made by the user, his location

[2] Assuming that the hash functions are independent and uniform.

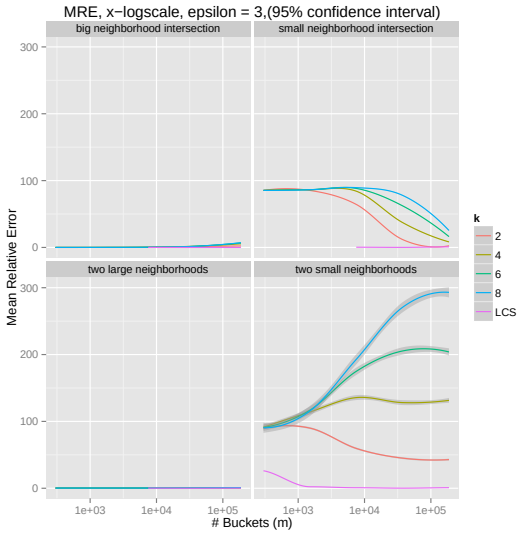

Fig. 2. Mean relative error for different number m of buckets with a differential privacy $\epsilon = 3$. The non-private linear counting sketch (LCS) baseline is also plotted.

as well as the time of the call are recorded. In our experiments, we have first extracted the set of users for each neighborhood and for each month and encode them within a Bloom filter. Afterwards, the Bloom filter was flipped according to the recipe of BLIP before release. Then, we use the algorithm from the previous section to estimate the intersection between each pair of neighborhoods given their flipped Bloom filters. To observe the behavior of our algorithm under different conditions, we specifically selected four sets of neighborhood couples. The choice of these four sets was based on the different number of users in the neighborhoods. For instance, the first set contains two neighborhoods with 57000 and 42000 users, and we call it "two large" in the figures while the next set contains two neighborhoods with 600 and 1000 users, and we call it "two small". For the two other sets, there is a small neighborhood and a large one, but one set has a small intersection between the two neighborhoods while the other one has a (relatively) big intersection. In particular, the first set has 15000 and 1200 users and an intersection of 180 users, thus we call it "small intersection". The second set has 3400 and 39000 users with an intersection of 3339 users and we call it "big intersection". All the experiments reported involving BLIP are averaged over 100 independent trials.

Evaluation. To assess the utility of our method, we compute the Mean Relative Error (MRE), which is defined as the absolute difference between the estimated intersection and the true intersection, divided by the true intersection. We also plot the standard deviation of the MRE for selected parameters. Figure 2 show the MRE obtained for a privacy level $\epsilon = 3$ and for different values of Bloom

filter parameters: k, the number of hash functions, as well as m the number of buckets. The value $\epsilon = 3$ is rather strict and provides high privacy guarantees. The MRE changes based on the properties of the neighborhoods considered. In particular, when the two neighborhoods are big or when the intersection is large, the MRE is less than 12%, regardless of the choice of parameters. However, when the two neighborhoods are small or when the intersection is small, the choice of parameters becomes critical. Generally, a lower value for k and a higher value for m is better, except when the two neighborhoods are small.

Conclusion. In this paper, we have proposed a method by which CDRs are sanitized in the form of a differentially-private Bloom filter for the purpose of privately counting of the number of mobile service users moving from one area (region) to another in a given time frame. The results obtained show that our method achieves - in most cases - a performance in terms of utility similar to another method (linear counting sketch) that does not provide any privacy guarantees. We leave as future work the possibility of performing other operations on differentially-private summaries as well as the design of privacy variants of more complex data structures such as trajectories and mobility models.

Acknowledgments. This work was partially supported by the MSR-INRIA joint lab as well as the INRIA project lab CAPPRIS, and by NSERC Canada.

References

1. M. Alaggan, S. Gambs, and A.-M. Kermarrec. BLIP: Non-interactive differentially-private similarity computation on bloom filters. In *SSS*, pages 202–216, 2012.
2. R. Balu, T. Furon, and S. Gambs. Challenging Differential Privacy:The Case of Non-interactive Mechanisms. In *ESORICS*, pages 146–164. Springer, 2014.
3. R. Bhatia and C. Davis. A better bound on the variance. *The American Mathematical Monthly*, 107(4), April 2000.
4. B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, July 1970.
5. A. Broder, M. Mitzenmacher, and A. B. I. M. Mitzenmacher. Network applications of bloom filters: A survey. In *Internet Mathematics*, pages 636–646, 2002.
6. C. Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.
7. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.
8. S. Gambs, M. Killijian, and M. N. del Prado Cortez. De-anonymization attack on geolocated data. In *TrustCom*, pages 789–797, 2013.
9. M. C. González, C. A. H. R., and A. Barabási. Understanding individual human mobility patterns. *CoRR*, abs/0806.1256, 2008.
10. J. Krumm. Inference attacks on location tracks. In *PERVASIVE*, pages 127–143, 2007.
11. J. Lee and C. Clifton. How much is enough? choosing ϵ for differential privacy. In *ISC*, pages 325–340, 2011.