Differentially Private Queries in Crowdsourced Databases for Net Neutrality Violations Detection

Maria Silvia Abba Legnazzi[†], Cristina Rottondi[*] and Giacomo Verticale[†]
[†] Politecnico di Milano, Department of Electronics, Information and Bioengineering
mariasilvia.abba@polimi.it, giacomo.verticale@polimi.it
[*] Dalle Molle Institute for Artificial Intelligence (IDSIA)
University of Lugano (USI) - University of Applied Science and Arts of Southern Switzerland (SUPSI)
cristina.rottondi@supsi.ch

***Abstract*—Lawmakers and regulatory bodies around the world are asserting Network Neutrality as a fundamental property of broadband Internet access. Since neutrality implies a comparison between different users and different ISPs, this opens the question of how to measure net neutrality in a privacy-friendly manner.**

This work describes a system in which users convey throughput measurements for the different services they use to a crowdsourced database and submit queries testing their measurements against the hypothesis of a neutral network. The usage of crowdsourced databases poses potential privacy problems, because users submit data that may possibly disclose information about their own habits. This leaves the door open to information leakages regarding the content of the measurement database.

Randomized sampling and suppression of small clusters can provide a good tradeoff between usefulness of the system, in terms of precision and recall of discriminated users, and privacy, in terms of differential privacy.

***Index Terms*—Differential Privacy; Network Neutrality;**

I. Introduction

The Network Neutrality principle aims at protecting and maintaining open, uninhibited access to legal online content without broadband Internet access providers being allowed to block, impair, or establish fast/slow lanes to lawful content.

In case of non-equal treatment of data traffic transmitted over the Internet, it is possible that an ISP is discriminating a user with respect to a service; it means that the performance experienced by the user is worse than the network performance experienced by other users accessing the same service with different providers.

Several policy-makers, most notably the FCC in the USA and the European Parliament in the EU, forbid discrimination by user or by service. Although the definition of Network Neutrality implies some measurement mechanism, there is no consensus on what are the mechanisms to detect whether some form of discrimination is in place. Researchers have proposed various approaches based on passive measurements campaigns by large content providers, on active measurements by public or private entities [1][2], or on crowdsourced measurements collected by the users [3][4][5]. In these last approaches, users submit measurement reports to a central server, which runs algorithms to verify the presence of discrimination by the ISP.

The usage of crowdsourced databases poses potential privacy problems, because users submit data that may possibly disclose information about their own habits. Therefore, it is necessary to put in place mechanisms that limit how much the collected data can be used to infer information beyond the intended meaning of the data collection.

This paper considers a scenario in which a user agent running on the user device collects passive measurements of the user activity and sends reports to a server in the form of a tuple of attributes. Periodically, the server makes a snapshot of the collected data and stores them in a database. Upon reception of a new tuple, the server provides the binary answer, "false" if there is no evidence of net neutrality violation, or "true" if that specific measurement might be due to net neutrality violation. Multiple measurements are then necessary to

achieve statistical significance.

There are two interfaces which require sanitization for privacy protection: the data collection step, in which users submit their data, and the query response, in which the answer is calculated and provided to the users. This paper focuses on the second step and provides the following new contributions: (1) the proof that a compliance test over a clustered database of subsampled data provides privacy in a differential sense; (2) the evaluation of the tradeoff between privacy and effectiveness in identifying net neutrality violations.

The rest of the paper is organized as follows. Section II provides a literature review on works about net neutrality and differential privacy; Section III introduces definitions and assumptions made to develop our system and provides a description of the database construction, the sanitization algorithm and the attack scenario; Section IV evaluates the privacy bounds provided by our proposed system in general and in case of Gaussian model; Section V discusses the validation system and the obtained results in terms of precision and recall; Section VI sums up the main contributions of the paper.

II. Related Work

A few systems issuing the evaluation of network neutrality have recently been proposed in the scientific community: The principle of the NANO system [5] is to establish a causal relationship between an ISP policy and the observed degradation of performance for a service using only passively collected data. It introduces the problem of privacy but it does not define the concept theoretically proving its guarantee. Unlike NANO, we provide a formal definition of differential privacy in a crowdsourced scenario.

Neubot [2] is an open source application, voluntarily installed by the users, that measures the characteristics of transmissions across the Internet. This tool does not detect explicitly net neutrality violations, but collects results from clients and continuously controls their performance. In addition, Neubot focuses on confidentiality that is guaranteed by the encryption of data. Instead, we focus on privacy preservation through an anonymization mechanism.

Glasnost [6] is a system that improves network transparency by enabling ordinary Internet users to detect whether their ISPs apply differentiated treatments to flows of specific applications. The aim of Glasnost is making any differentiation transparent to users using throughput as measure of flow performance like in our system, but it does not deal with privacy issues.

DiffProbe [7] is an active probing method that aims to detect discrimination when it actually affects user traffic. Also in this work there is no privacy definition and the considered metrics for assessing discriminations are delay and loss, but not connection throughput.

The notion of differential privacy was first introduced by the seminal work by Dwork et al. in [8]. Differential privacy aims at guaranteeing that the removal or addition of a single item in a statistical database has negligible impact on the outcome of any query on that database. The author gives a formal definition of differential privacy as a measure of the tradeoff between the precision of the aggregate data and the probability of identifying the contributions of individual data inside the aggregate.

In this paper, we will apply a sanitization approach similar to the one proposed in [9], which shows that a k-anonymity-based sanitization algorithm can satisfy differential privacy when preceded by a random sampling of the database entries.

III. System Model

A. Basic Mechanism

We assume that each user sends a tuple q containing the following attributes: date and time, location, type of application and/or server, ISP, subscribed broadband service tier, and one or more measurements evaluating the service quality (e.g. throughput, latency, or jitter).

We adopt the basic assumption of NANO [5], i.e. that, all other things being equal, the majority of ISPs complies with net neutrality rule. Therefore, a net neutrality violation can be detected by comparing the performance received by the subscribers of a given ISP to the performance received by the subscribers of all other ISPs, after taking into account the effect of any confounding factors.

Many factors other than differentiated treatment may affect the performance of a particular service

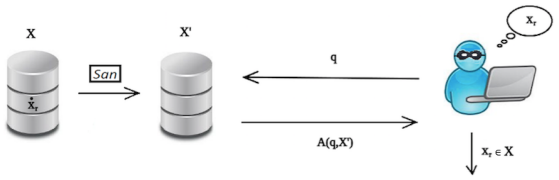

Fig. 1. The attack scenario

or application. For example, a service may be slow due to overload at a particular time of day or it might be supplied in a location characterized by worse performance. Similarly, the performance might depend on software or hardware, or other network peculiarities.

Consequently, we consider ISP as the *treatment variable*, any performance measurement, in particular the throughput, as an *outcome variable*, and any other parameters such as time, location and network speeds as the *confounding variables*. A confounding variable (or simply *confounder*) is one that correlates both with the considered treatment variable (i.e., the ISP) and the outcome variable (i.e., the performance).

Similarly to [5], we use stratification to gather confounding variables together. Stratification places measurements into clusters such that all the samples in each cluster have "similar" values for the confounding variables. Inside each cluster, the treatment and the outcome variables can be considered independent of the confounding variables. The procedure that maps samples into clusters is called generalization. In this work, we consider data independent generalization, meaning that the clusters are defined before the data are collected. In particular, we define upper and lower thresholds for each confounding attribute and we define a cluster as the set of all instances whose confounding attributes all fall within the same threshold bounds.

We consider a semi-honest adversary $\mathcal{A}$. The adversary adheres to the protocol rules, but it can freely choose its input and store all received messages with the aim of inferring additional information w.r.t. what is implied by the knowledge of the query answers.

In particular, as depicted in Figure 1, the adversary wants to ascertain whether an arbitrarily chosen tuple x_r is present or not in the database X. Conversely, the database is assumed to be an honest entity.

B. Database Construction and Sanitization

A database X is a collection of N data rows $x_1, \ldots, x_N$ containing elements drawn from a public universe $\mathcal{U}$. We assume that the content of X does not change in a given time frame. Each data row consists of a set of L values taken from a domain $D = D_1 \times \cdots \times D_L$.

The database users interact with the database by submitting queries $q_1, \ldots q_Q$, which are themselves drawn from $\mathcal{U}$. The database answer is a binary value representing whether the submitted tuple is compatible or not with the tuples already in X. Similarly to [5], the parameters in domain D are divided in three classes: treatment, confounder, and outcome. For the sake of simplicity, we will consider the case in which there is a single treatment variable D_1 and a single outcome variable D_L. The other variables are the confounders.

A Data Independent Generalization (DIG) function $g(x)$ takes as input a tuple from $\mathcal{U}$ and associates it to a cluster of similar tuples. For the sake of simplicity, we assume that each cluster can be labeled with a natural number. It is worth noting that the generalization function $g(x)$ takes into account neither the treatment nor the outcome variables. We assume that the clustering parameters are given.

Each cluster is also associated to a compliance interval, calculated from the outcome field of the tuples in X that are part of the same cluster and have a different treatment variable. In practice, we compare the user own measurements to measurements from similar users having a different ISP.

What is the most accurate way to calculate a compliance interval is a matter of study and, in general, it is necessary that a significant number of repeated measurements fall outside the compliance interval before one can conclude that some kind of non-neutral traffic treatment is in place. In this paper we calculate the compliance interval for cluster i as: $[M_i(X) - \sigma, M_i(X) + \sigma]$ where $M_i(X)$ denotes the sample mean of the outcome field of the tuples in X that fall in cluster i (i.e., with equal values of confounders), and σ is a system parameter controlling the tradeoff between detection, precision and recall. The query q returns False (0) if the outcome

field of q falls inside the compliance interval, True (1) otherwise.

We consider the following sanitization building blocks:

- a β-sampling mechanism that samples database X with probability β;
- a Data Independent Generalization (DIG) function $g(x)$ that divides into intervals the confounding variables in order to obtain clusters in which samples have equal values for the generalized attributes;
- a k-suppression mechanism, which eliminates from X all the clusters having fewer than k tuples.

First of all the database X is sampled through the β-sampling algorithm, i.e. each tuple is dropped with probability $1-\beta$. After that, the generalization mechanism $g(x)$ takes as input the sampled version of X and produces a new dataset linking user's measures to a cluster based on common confounding variables. Finally we apply the k-suppression algorithm to remove any cluster containing less than k tuples and we obtain the new database X'.

C. Privacy Definition

We evaluate privacy in the Differential Privacy model [10]. Let $\mathcal{A}(q, X)$ the result of submitting the query q to X. The system consisting of the database and the sanitization algorithm provides (ϵ, δ)-differential privacy if, for all q, x_r and b, the following holds with probability no smaller than δ:

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(q, X) = b]}{\Pr[\mathcal{A}(q, X \backslash x_r) = b]} \leq e^{\epsilon} \tag{1}$$

where x_r is a tuple from X and $X \backslash x_r$ is the database X with the tuple x_r removed.

IV. Privacy Evaluation

The effectiveness of the Sanitization Algorithm in providing privacy depends on the underlying data model. In this Section, we identify the conditions that guarantee (ϵ, δ)-differential privacy under suitable assumptions. We consider the scenario of a single query. In case of multiple queries, exploiting the composition theorem [10], the sanitization algorithm provides $(Q\epsilon, Q\delta)$-differential privacy, with Q the number of queries submitted to X.

A. General Case

In the general case, we assume that, for all the tuples in each cluster, the outcome variable is drawn from an unknown distribution with probability density function $f_{\gamma_i}(x)$, where i is the cluster label. We state the following theorem. In the appendix, we provide the main reasoning. The full proof is available in an extended version of the paper.

Theorem 1. *The system consisting of the database and the sanitization algorithm provides (ϵ, δ)-differential privacy with*

$$\epsilon = \ln\left(N_i(1-\beta) + \mathcal{K}'\right) \qquad \forall\ N_i \geq k \tag{2}$$

$$\delta = \left[1 - \sum_{l=0}^{k-1} Bi(N_i, \beta, k)\right] \sum_{j=0}^{k-1} Bi(N_i - 1, \beta, k) + \sum_{l=0}^{k-1} Bi(N_i, \beta, k) \left[1 - \sum_{j=0}^{k-1} Bi(N_i - 1, \beta, k)\right] \tag{3}$$

The constant N_i is the size of cluster i and the constant k is the minimum cluster size set by the suppression algorithm. The constant $\mathcal{K}'$, defined in (8)*, depends on $f_{\gamma_i}(x)$ and approaches zero as N_i grows. The function $Bi(N_i, \beta, k)$ is the binomial cumulative density function for k successes out of N_i trials with success probability β.*

As N_i grows, $\mathcal{K}'$ becomes negligible and ϵ can be bounded as:

$$\epsilon = \ln\left(N_i(1-\beta)\right) \quad \forall\ N_i \geq k \tag{4}$$

Figure 2 shows the upper bound of ϵ in (4) for different values of β and N_i, which assumes the value of minimum cluster size k.

Figure 3 depicts the value of δ in (3) with increasing level of anonymization and cluster size $N_i = 100$. The probability δ keeps small values for significantly small sampling probability $\beta \leq 1\%$. However, when $N_i \sim k/\beta$ we underline that δ assumes higher values, in this case it is necessary to force a limit in the number of queries Q otherwise the privacy is hardly guaranteed.

B. Gaussian Model

Theorem 1 proves that the sanitization algorithm provides privacy, but the ϵ bound resulting from (4) and plotted in Figure 2 is of limited practical use,

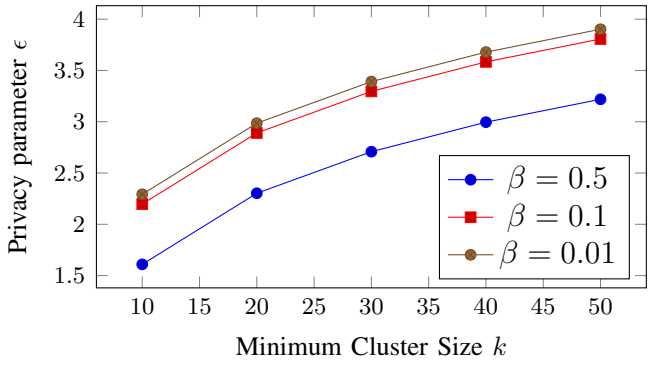

Fig. 2. Upper bound of the privacy parameter ϵ in the general case.

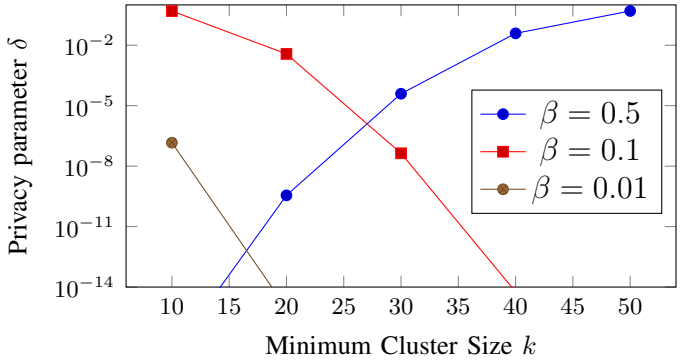

Fig. 3. Privacy parameter δ in the general case for a cluster of $N_i = 100$ elements

since it is too large and becomes even larger as k grows. A more useful estimation can be obtained by making stronger assumptions on the distribution of the data in the cluster. We replace in (5) the well known probabilistic functions of a Gaussian distribution with mean value μ and standard deviation σ, once both concerning downloading and once both concerning video streaming, the two service types introduced in Section V.

We use Monte Carlo method based on random sampling in order to obtain numerical results. Thus we generate multiple realizations of X and $X \backslash x_r$ and we apply the sanitization mechanism with different value of β and k in order to obtain the epsilon bounds in Figure 4. We average the different simulations and we verify the ratio in (1) for several queries q choosing the maximum value.
At this point, in Figure 4 we plot the decreasing trend of the value of epsilon and the resulting gain in privacy level versus the increasing anonymization level.

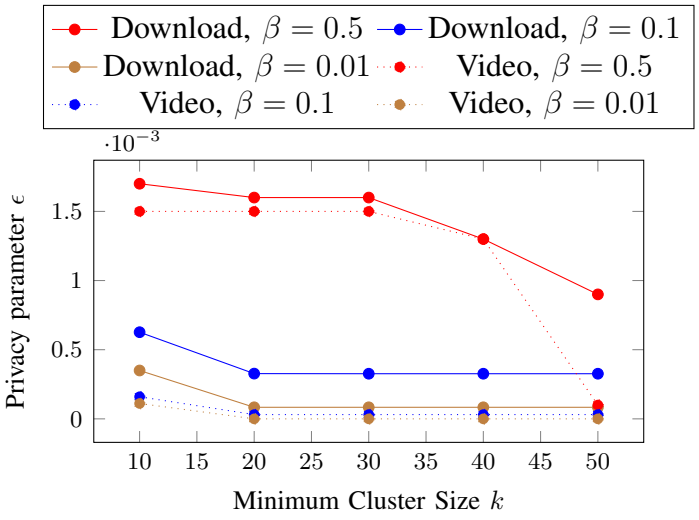

Fig. 4. Privacy parameter ϵ in the case of normally distributed data and cluster size $N_i = 100$

V. Numerical Assessment

A. Validation Method

We evaluate the privacy-precision tradeoff by applying the technique described in this paper to the measurements provided by SamKnows during the project *Measuring Broadband America* [1].
The raw data collected from the measurement database for each active metric are made available by month in tarred gzipped files. We use the September 2013 data validated by SamKnows through a work of anomalies' removal, such as IP address out of valid range or throughput inconsistent with the service tier provisioned by the ISP.
We exploit files related to download speed, web browsing, video streaming and voice over IP. For all these services we consider the following attributes: unit profile identifier, time test finished and running total of throughput. Then, information regard service tiers (ISP, download speed, upload speed) and regard position (longitude, latitude) is included in unit profile and unit census block files, respectively.

Based on the above mentioned attributes, fifty millions of database entries are classified in different categories, as shown in Table I.

We try to identify when service performance differs across ISPs but confounding factors are equal. A big challenge in designing such a system is to identify the confounding factors and create an

TABLE I
VARIABLES AND GENERALIZATION RULES

Treatment variable	
ISP	–
Confounder variables	
time	hour and day of the week
longitude	areas of five degrees
latitude	areas of five degrees
up	steps of ten megabit per second
down	steps of ten megabit per second
service	four different services
Outcome variable	
throughput	–

environment where all confounding factors are equal or independent of the ISP or service performance.

In order to validate our detection mechanism we randomly select an ISP and trim the corresponding throughput measurements similarly to a policing algorithm. We set at one Megabit per second the threshold at which throughput is truncated according to a feasible policing procedure.

Then, we randomly partition the dataset in training set (which includes 85% of the available tuples) and a test set. The training set becomes the database X, which is then divided in clusters and sanitized in order obtain the new database X'. We set a boolean variable *ground truth* to True in case of alteration or to False in case of neutral network. Conversely, the attribute *category* assumes value True in case of detection of a probable net neutrality violation or value False when the measure falls within the compliance interval.

B. Results

We start applying the algorithm to the full database, with no sanitization, in order to assess the baseline performance of the detection technique. We label this case as "baseline" in the following figures. The resulting precision, defined as the ratio of the correct detections of net neutrality violations to the total detections, depends on the service. In case of the download service, it is about 58%, while for the video streaming service it is about 29%. These figures, which are not very good in absolute terms, refer to a single query. In order to declare that a violation is indeed occurring, it is necessary to collect multiple queries over multiple days. In turn, the number of necessary queries depends on the extent of the violation. The correct identification of the observation interval is out of the scope of this paper.

The recall in the baseline scenario, defined as ratio of the number of detections by the number of violations in the data, is more than 99% for the downloading service and about 55% for the video streaming service. The relatively low performance for the video streaming service is due to the fact that the service comprises heterogeneous streams with different speeds that depend on the receiver device and on the resolution of the video stream. For a more accurate detection of violations in the video streaming service it is therefore necessary to consider other confounding variables such as the resolution of the stream. Unfortunately, these variables are difficult to obtain and were not available in our data. Nevertheless, since our the goal is to study how sanitization impacts on the recall, we will show the impact of the various sanitization parameters with respect to the baseline scenario, which represents the best result.

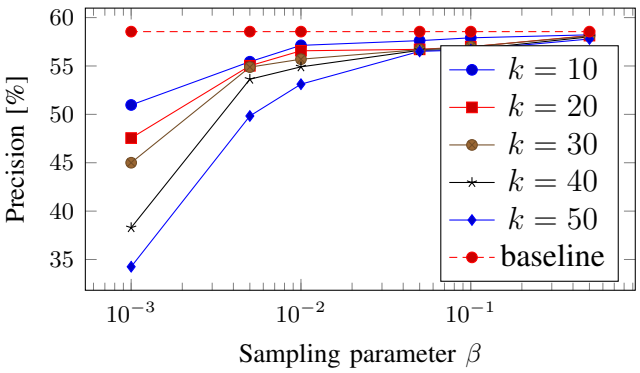

Fig. 5. Precision versus the sanitization parameters for the downloading service

Figures 5 and 6 show the precision for the downloading and the video streaming services, respectively, versus the sanitization parameters β and k. The figures also show the baseline precision. As the sampling parameter β grows, the precision also grows, with bad performance for $\beta \leq 0.1\%$ and with results very similar to the baseline for $\beta \geq 10\%$ for all the values of k.

Figures 7 and 8 show the recall for the downloading and in the video streaming services, respectively,

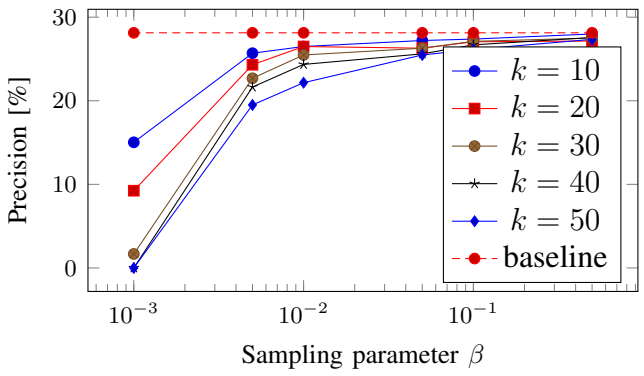

Fig. 6. Precision versus the sanitization parameters for the video streaming service

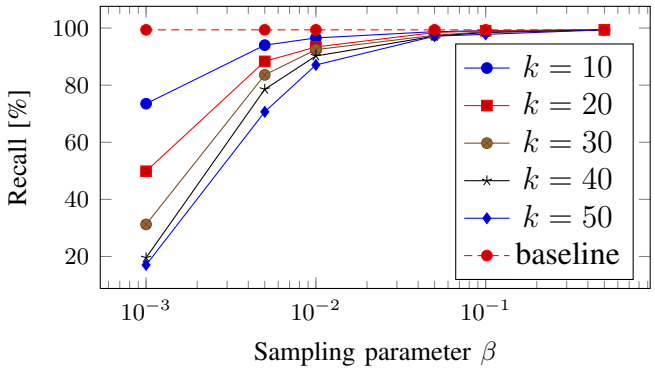

Fig. 7. Recall versus the sanitization parameters for the downloading service

versus the sanitization parameters β and k. The performance of the sanitization algorithm in terms of recall is similar to the performance in terms of precision, with good results for $\beta \geq 10\%$ and significant recall loss for smaller β. In addition, for large β the impact of k is negligible, but for small values of β we notice increasingly worse results for larger k. This is mainly due to the somehow arbitrary decision to declare that a measurement is not a violation if the relevant cluster has been suppressed and the number of suppressed clusters is large if β is small or k is large. A different assumption, however, would negatively impact the precision, which is already critical.

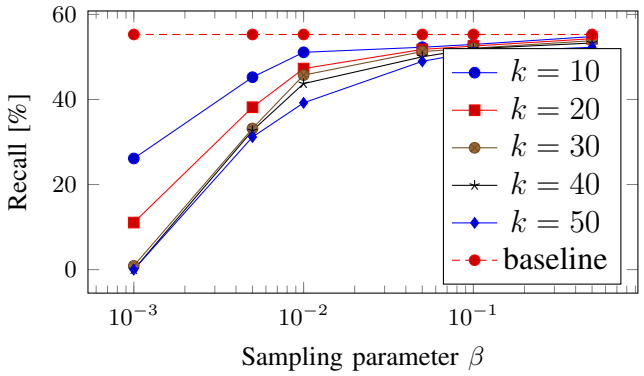

Fig. 8. Recall versus the sanitization parameters for the video streaming service

Finally, with data at our disposal, we prove the existence of finite values of β, between $10\% - 1\%$, and finite values of k, near 30, that make possible to reach both good quality parameters, in terms of precision and recall, and good privacy bounds, ϵ and δ, of the detection technique.

VI. Conclusion

We describe an algorithm for the crowdsourced detection of possible net neutrality violations. We also consider the application of a sanitization technique of the collected measurements in order to protect sensitive data of users exploiting such a system.

We formally prove that data independent generalization, subsampling, and suppression of small clusters, make it possible to achieve privacy under the differential privacy model.

We also assess the effectiveness of the algorithm in detecting neutrality violations by using a large dataset of measurements of broadband traffic by home users. We show that a small subsampling along with the elimination of very small clusters is capable of providing minimal performance loss and, at the same time, provide a good degree of privacy.

Appendix
Proof of Theorem 1

Let N_i be the number of tuples of X belonging to the same cluster i as x_r. We assume that these tuples are drawn from an unknown distribution with probability density function $f_{\gamma_i}(x)$. Let Y and Z be the number of tuples selected by the sampling algorithm over the databases X and $X \backslash x_r$. Clearly Y and Z are drawn from a binomial distribution with parameters (N_i, β) and $(N_i - 1, \beta)$ respectively. Let $y_1, \ldots, y_Y$ be the tuples sampled from X and $z_1, \ldots, z_Z$ be the tuples sampled from $X \backslash x_r$. We distinguish three different cases:

1) if $Y < k$ and $Z < k$, then the cluster is removed from both databases X and $X \backslash x_r$. This case is trivial.
2) if $Y \geq k$ and $Z \geq k$, then no cluster is removed;
3) otherwise, the cluster is removed only in one database, in X or in $X \backslash x_r$.

Case 2. No cluster is removed: We prove first the case with $b = 0$; the case with $b = 1$ is similar. We have that $\mathcal{A}(q, X) = 0$ and $\mathcal{A}(q, X \backslash x_r) = 0$ if and only if

$$-\sigma < \tfrac{1}{Y} \textstyle\sum_{j=k}^{Y} y_j - q < \sigma \quad \forall q$$
$$-\sigma < \tfrac{1}{Z} \textstyle\sum_{j=k}^{Z} z_j - q < \sigma \quad \forall q$$

Let S_{N_i} be the mean of independent random variables with probability density function $f_{\gamma_i}(x)$ sampled with probability β from a population of N_i. Let $f_{S_{N_i}}(x)$ be its probability density function and $F_{S_{N_i}}(x)$ be its cumulative distribution function. We have:

$$e^{-\epsilon} \leq \frac{\int_{q-\sigma}^{q+\sigma} f_{S_{N_i}}(x)dx}{\int_{q-\sigma}^{q+\sigma} f_{S_{N_i-1}}(x)dx} \leq e^{\epsilon} \quad \forall q \tag{5}$$

We consider the right inequality of (5), for which we have:

$$F_{S_{N_i}}(q+\sigma) - F_{S_{N_i}}(q-\sigma) - e^{\epsilon} F_{S_{N_i-1}}(q+\sigma) + e^{\epsilon} F_{S_{N_i-1}}(q-\sigma) \leq 0 \tag{6}$$

Let $\phi_{\gamma_i}(\omega)$ be the characteristic function of X's tuples and $\phi_{S_{N_i}}(\omega)$ be the characteristic function of S_{N_i}. Equation (6) can be rewritten as:

$$\frac{1}{2\pi}\left\{\int_{-\infty}^{\infty} \frac{e^{-j(q-\sigma)\omega} - e^{-j(q+\sigma)\omega}}{j\omega} \beta^{N_i} \phi_{\gamma_i}\left(\frac{\omega}{N_i}\right)^{N_i} d\omega \; + \right.$$
$$\int_{-\infty}^{\infty} \frac{e^{-j(q-\sigma)\omega} - e^{-j(q+\sigma)\omega}}{j\omega} \sum_{j=k}^{N_i-1} \phi_{\gamma_i}\left(\frac{\omega}{j}\right)^{j} \beta^{j}(1-\beta)^{N_i-1-j} \cdot$$
$$\left.\binom{N_i - 1}{j}\left[\frac{N_i(1-\beta)}{N_i - j} - e^{\epsilon}\right] d\omega\right\} \leq 0 \tag{7}$$

We observe that the last term in the summation, with index $j = N_i - 1$, is the largest. In addition, we define the constant $\mathcal{K}'$ as:

$$\mathcal{K}' = \frac{\beta \int_{-\infty}^{\infty} \frac{e^{-j(q-\sigma)\omega} - e^{-j(q+\sigma)\omega}}{j\omega} \phi_{\gamma_i}\left(\frac{\omega}{N_i}\right)^{N_i} d\omega}{(N_i - 1)\int_{-\infty}^{\infty} \frac{e^{-j(q-\sigma)\omega} - e^{-j(q+\sigma)\omega}}{j\omega} \phi_{\gamma_i}\left(\frac{\omega}{N_i - 1}\right)^{N_i - 1} d\omega} \tag{8}$$

So we can obtain the new equation:

$$e^{\epsilon} \geq N_i(1-\beta) + \mathcal{K}' \quad \forall \; N_i \geq k \tag{9}$$

A similar bound can be found for the left inequality of (5), but can be ignored since (9) always provide a stricter condition.

Case 3. The cluster is removed only in one database: In this scenario it is possible to find a condition under which (ϵ, δ)-differential privacy can be satisfied with the same ϵ calculated in previous case and with δ reflecting the probability that this case occurs, which is:

$$\delta = \; Pr\{Y \geq k, Z < k\} + Pr\{Y < k, Z \geq k\}$$

After simple substitutions, we obtain (3).

REFERENCES

[1] FCC. (2014) Validated data - Measuring broadband America 2014. https://www.fcc.gov/general/validated-data-measuring-broadband-america-2014.

[2] J. C. De Martin and A. Glorioso, "The neubot project: A collaborative approach to measuring internet neutrality," in *2008 IEEE International Symposium on Technology and Society*. IEEE, 2008, pp. 1–4.

[3] Z. Zhang, O. Mara, and K. Argyraki, "Network neutrality inference," *SIGCOMM Comp. Comm. Rev.*, vol. 44, no. 4, pp. 63–74, Aug. 2014.

[4] D. Miorandi, I. Carreras, E. Gregori, I. Graham, and J. Stewart, "Measuring net neutrality in mobile internet: Towards a crowdsensing-based citizen observatory," in *2013 IEEE International Conference on Communications Workshops (ICC)*, June 2013, pp. 199–203.

[5] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar, "Detecting network neutrality violations with causal inference," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. ACM, 2009, pp. 289–300.

[6] M. Dischinger, M. Marcon, S. Guha, P. K. Gummadi, R. Mahajan, and S. Saroiu, "Glasnost: Enabling end users to detect traffic differentiation." in *NSDI*, 2010, pp. 405–418.

[7] P. Kanuparthy and C. Dovrolis, "Diffprobe: detecting isp service discrimination," in *INFOCOM, 2010 Proceedings of*. IEEE, 2010.

[8] C. Dwork, "Differential privacy," in *Automata, languages and programming*. Springer, 2006, pp. 1–12.

[9] N. Li, W. H. Qardaji, and D. Su, "Provably private data anonymization: Or, k-anonymity meets differential privacy," *Arxiv preprint*, 2011.

[10] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.