Noname manuscript No.
(will be inserted by the editor)

Secure and Differentially Private Detection of Net Neutrality Violations by means of Crowdsourced Measurements

Maria Silvia Abba Legnazzi · Cristina Rottondi · Giacomo Verticale

the date of receipt and acceptance should be inserted later

Abstract Evaluating Network Neutrality requires comparing the quality of service experienced by multiple users served by different Internet Service Providers. Consequently, the issue of guaranteeing privacy-friendly network measurements has recently gained increasing interest. In this paper we propose a system which gathers throughput measurements from users of various applications and Internet services and stores it in a crowdsourced database, which can be queried by the users themselves to verify if their submitted measurements are compliant with the hypothesis of a neutral network. Since the crowdsourced data may disclose sensitive information about users and their habits, thus leading to potential privacy leakages, we adopt a privacy-preserving method based on randomized sampling and suppression of small clusters. Numerical results show that the proposed solution ensures a good trade-off between usefulness of the system, in terms of precision and recall of discriminated users, and privacy, in terms of differential privacy.

Keywords Differential Privacy; k-Anonymity; Network Neutrality

1 Introduction

The role of Network Neutrality (NN) in today's Internet is a topic of debate in lawmaker and policy-maker councils. The principle of NN, also referred to as Open

Authors are listed in alphabetical order

Maria Silvia Abba Legnazzi
EY Italy (work was done while at Politecnico di Milano)
E-mail: mariasilvia.abba@mail.polimi.it

Cristina Rottondi
Dalle Molle Institute for Artificial Intelligence (IDSIA) – University of Lugano (USI) – University of Applied Science and Arts of Southern Switzerland (SUPSI)
E-mail: cristina.rottondi@supsi.ch

Giacomo Verticale
Politecnico di Milano, Department of Electronics, Information and Bioengineering
E-mail: giacomo.verticale@polimi.it

Internet paradigm [10], affirms that access to legitimate content published online must be guaranteed by Internet Service Providers (ISPs) without any form of inhibition achievable by blocking, impairing or delaying the transmission of certain categories of data streams through their network infrastructure [3]. Discriminatory treatment of user generated traffic translates into a lower quality of service experienced by the targeted users and is forbidden by several regulation and governmental bodies such as the European Parliament in the EU.

Though the scientific community has yet not reached a general consensus on metrics and detection approaches to individuate and quantify a potential NN violation, the establishment of suitable measurement mechanisms is a necessary requirement and several approaches based either on passive measurements campaigns by large content providers [1], on active measurements by public or private entities [19][5], or on crowdsourced measurements collected by the users [22][16][21] have been investigated. The latter approaches require users to submit measurement reports to a central server, which stores the collected data in a database and runs detection algorithms to identify potential traffic discriminations actuated by ISPs.

The popularity of such crowdsourced approaches is rapidly growing: they are used in several contexts, such as public funding and market development, and also by well-known sites as Wikipedia and Facebook. Unfortunately, the entries of crowdsourced databases may disclose sensitive information about users' habits, interests and preferences, thus raising potential privacy issues. Therefore, privacy-preserving approaches need to be adopted to limit the amount of additional information that can be extracted from the gathered measurements, beyond that directly related to the scope of the collection.

In this paper, we consider a framework in which passive measurements about the user's activities are collected by an agent installed on the user's device and stored in a local database as tuples of numerical and/or categorical attributes. The server periodically receives client measurements, aggregates them and calculates a compliance interval. Then, it is available to answer client queries about the compliance of their measurements. A client receiving multiple indications that their service is not compliant might conclude that NN was violated.

Ensuring privacy preservation in NN violation detection requires the application of sanitization approaches in two distinct phases: the measurement collection phase, when user agents submit their measurements, and the query response phase, in which the result of the NN violation check is elaborated and communicated to the user. This paper focuses on the latter phase and provides two novel contributions:

- it proves that the proposed NN compliance test over a clustered database of subsampled data ensures privacy guarantees under the differential privacy framework;
- it evaluates the trade-off between privacy level and correct identification of NN violations.

The remainder of this paper is organized as follows: in Section 2 we review the recent literature on NN and differential privacy; in Section 3 we introduce basic definitions and assumptions necessary for the development of our system and describe the database construction, the sanitization algorithm and the attack scenario; Sections 4 and 5 evaluate the security, overehad and privacy bounds

provided by our proposed system; Section 6 discusses the validation system and the obtained results in terms of precision and recall; Section 7 concludes the paper.

2 Related Work

2.1 Network Neutrality

We now briefly review some recently proposed systems for NN violation detection. The NANO system [21] aims at establishing a causal relationship between an ISP traffic policy and the performance degradation experienced by users of a given service, by using only passively collected data. Though the authors of [21] mention the existence of privacy issues, they do not provide a theoretical formalization of the privacy notion nor quantify achievable privacy guarantees. In this paper, we provide a formal definition of differential privacy in a crowdsourced scenario.

Neubot [5] is an open-source application measuring transmission-related parameters, which can be installed by the users on voluntary basis. It is not explicitly aimed at NN violation detection, but simply monitors the network performance experienced by the users. Neubot ensures confidentiality by means of data encryption techniques, but does not address privacy issues, whereas in this contribution we focus on privacy preservation through an anonymization mechanism.

The Glasnost system [6] adopts throughput measurements to detect whether the user's ISP applies differentiated treatments to data flows generated by specific applications. In this paper, we adopt the same type of throughput measurements. Conversely, DiffProbe [12] applies an active probing method to detect NN violations based on packet delay and loss measurements. Both systems do not deal with privacy-related issues.

A crowdsensing-based hybrid active/passive network monitoring framework is proposed in [11], which leverages measurement agents embedded in the devices of distributed systems deployed in the wild. However, no discussion on possible privacy implications of such approach is provided.

A methodology for acquisition, analysis and performance comparison of throughput statistics by video hosting services aimed at identifying NN violations is discussed in [4]. Though we also focus on the video-streaming service in our performance analysis, our proposed framework is more general and can be applied to any type of traffic.

Among the recent studies addressing the issues of NN, the authors of [15] show that network performance measurements can be easily manipulated by defensive ISPs to hide their non-neutral behavior and propose a stealth neutrality measurement tool which exploits covert channels.

2.2 Differential Privacy

The differential privacy framework was proposed in the seminal work by Dwork et al. [7], which provides a theoretical approach to quantify the probability of identifying the presence/absence of a single entry within a statistical database. To ensure differential privacy, adding or removing one database item should not significantly impact on the result of any query issued on the whole database content.

Differential privacy provides a semantically-flavoured privacy definition, since such definition is independent of the adversarial knowledge [13]. Therefore, it overcomes the inherent limitations of syntactic privacy definitions, which assume a specific attack model and a limited adversarial knowledge, thus failing in providing privacy guarantees in case such knowledge exceeds the limits assumed in the model [2]. Nevertheless, some syntactic privacy definitions such as k-anonymity [20] have been proven to satisfy differential privacy under specific assumptions: in particular, the authors of [14] show that a k-anonymity-based sanitization algorithm can satisfy differential privacy when preceded by a random sampling of the database entries. In this paper, we adopt the same sanitization approach.

3 System Model

3.1 Basic Mechanism

We assume that each user sends to the server a tuple 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 [21], i.e. that, all other things being equal, the majority of ISPs complies with the NN paradigm. Therefore, a NN 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 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 the 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 [21], we use stratification to gather confounding variables together. Stratification places measurements into clusters such that all the samples in each cluster have "similar" values of the confounding variables. Inside each cluster, the treatment and the outcome variables can thus 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 fall within the same threshold bounds.

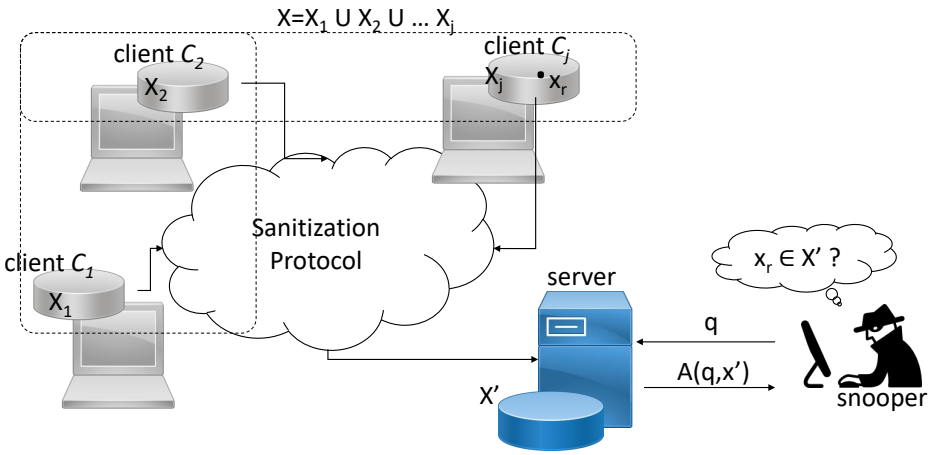

Fig. 1 System Model. The virtual database X is the union of the databases stored at each node. The Sanitization Protocol is executed for each cluster i. The database can be queried by client nodes or by malicious snoopers. The test tuple x_r may be present or not.

3.2 System Architecture

The system architecture is shown in Figure 1. It comprises a server and a set of client nodes C_j each holding a database of local measurements X_j. The full database X is the union of the local databases and is not centrally stored. Each data row consists of a set of L values taken from a domain $\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_L$. The clients interact with the database by submitting queries $q_1, \ldots q_Q$ to the server, whose values are themselves drawn from $\mathcal{D}$. The database answer is a binary value indicating whether the submitted tuple is compatible or not with the tuples already in X.

Similarly to [21], the parameters in domain $\mathcal{D}$ 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 $\mathcal{D}_1$ and a single outcome variable $\mathcal{D}_L$. The other variables are the confounders.

A Data Independent Generalization (DIG) function $g(x)$ takes as input a tuple from X 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. We consider a generalization function $g(x)$ that takes into account neither the treatment nor the outcome variables and whose clustering parameters are given and need not be extracted from the data. The DIG function is public.

The server runs an instance of the Sanitization Protocol and builds a new database X', which contains, for each cluster, a compliance interval, calculated from the outcome field of the tuples in X that are part of the same cluster. In addition to data generalization, the sanitization protocol implements a β-sampling mechanism, which randomly sumbsamples the database by a factor $1/\beta$, and a k-suppression mechanism, which eliminates from X' all the clusters containing fewer than k tuples.

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:

$$\left[M_i - \kappa\sqrt{S_i^2}, M_i + \kappa\sqrt{S_i^2}\right]$$

where M_i 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), S_i^2 is the sample variance, and κ is a system parameter controlling the tradeoff between privacy, precision and recall. For the sake of simplicity, we calculate the compliance interval for each cluster over the tuples in X that fall inside the cluster, therefore ignoring the treatment variable.

3.3 Security Assumptions and Goals

We make the following assumptions:

1. Each client reports at most one measurement per cluster.
2. Clients are semihonest: they execute the protocol honestly, but may use any collected information to infer information about other clients' databases.
3. The server is semihonest: it executes the protocols honestly, but may use any collected information to infer information about the clients' databases.
4. The snooper is semihonest: it executes the protocols honestly, but it can freely choose its input and may use any collected information to infer information about the content of the virtual database X.
5. Nodes do not collude with one another.
6. the Diffie-Hellman Key Exchange protocol generates a secure, pseudorandom shared secret.

The possible attackers in the system are the clients, the server, and the snooper, which can either be a client or an external entity. With respect to the clients, the sanitization protocol must guarantee confidentiality of the local measurement databases: clients should not learn information about other clients' local databases, and the server should learn only the information needed to calculate the compliance intervals. These assumptions leave out the case of a dishonest client that also behaves as a snooper. Such client could inject into the database a set of bogus measurements and then cleverly craft queries to infer specific information about other clients. This is a special case of the more general problem of how to prevent clients from polluting the database with incorrect measurements. In general, it is very hard to prevent this from happening; however, the problem can be mitigated by limiting the number of measurements per cluster conveyed by any single client.

In the following we will consider the concrete security model, in which a feasible algorithm is defined as any algorithm that can be executed in, at most, a given number of CPU cycles and a negligible value ϵ is defined as some very small number.

Definition 1 (Security against Malicious Clients) The system is secure against malicious clients if any feasible client has negligible advantage in guessing $b' = b$ in Algorithm 1; for any M_0 and M_1, we have

$$|\Pr(\textsc{Client Attack}(0) = 1) - \Pr(\textsc{Client Attack}(1) = 1)| \leq \epsilon$$

Algorithm 1 Client's Attack Experiment

function CLIENT ATTACK(b) ▷ $b \in \{0, 1\}$
M_0 and M_1 are two sets of possible client measurements such that M_0 and M_1 have the same cardinality. The Attacker's own measurement is present both in M_0 and in M_1.
The protocol is executed with M_b
The attacker executes any feasible algorithm and calculates b'
return b' ▷ Attacker's guess
end function

Definition 2 (Security against Malicious Server) The system is secure against malicious server if any feasible server has negligible advantage in guessing $b' = b$ in Algorithm 2; for any M_0 and M_1, we have

$$|\Pr(\text{SERVER ATTACK}(0) = 1) - \Pr(\text{SERVER ATTACK}(1) = 1)| \leq \epsilon$$

Algorithm 2 Server's Attack Experiment

function SERVER ATTACK(b) ▷ $b \in \{0, 1\}$
M_0 and M_1 are two sets of possible client measurements such that M_0 and M_1 have the same cardinality, the same mean and the same variance.
The protocol is executed with M_b
The attacker executes any feasible algorithm and calculates b'
return b' ▷ Attacker's guess
end function

With respect to the snooper, the server should protect the privacy of the user conveying their information the the database. In particular the snooper wants to ascertain whether an arbitrarily chosen tuple x_r is present or not in the database X. We evaluate privacy in the Differential Privacy model [8].

Definition 3 (Privacy against Snoopers) 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 $(1 - \delta)$:

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

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

3.4 Sanitization Protocol

The distributed protocol comprises an initialization step, in which the server publishes public parameters, a first phase, in which the clients commit to send a measurement, and a second phase in which the clients send the masked measurements. The two phases of the protocol are shown in Figure 2 and are repeated for each measurement.

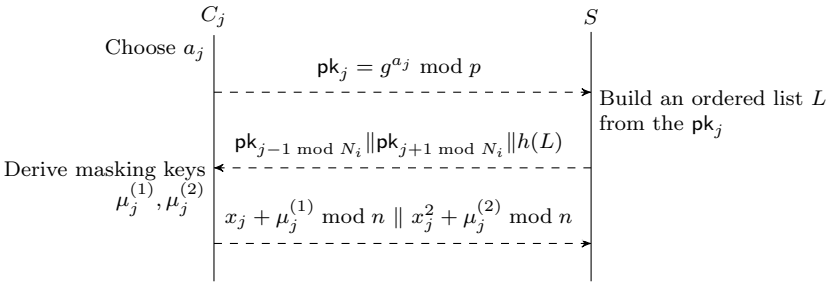

Fig. 2 Message flow of the sanitization protocol executed for cluster i.

3.4.1 Initialization

The server publishes:

- Parameters β, k, and the function DIG, which maps the set of confounding variables to a cluster, identified with a unique integer number.
- Diffie-Hellmann parameters g and p.
- An integer number n larger than the square of any possible measurement, times the maximum number of measurements in a cluster. Measurements are encoded as integer numbers.
- A key derivation function $\mathsf{KDF}\colon \{0,1\}^* \to \mathbb{Z}_n$.
- A cryptographically secure hash function h.

3.4.2 Phase 1

Suppose a node holds a measurement. It randomly decides whether to proceed with the protocol, with probability β, or to stop. The same node can later make a query to the server to verify whether the measurement is compliant or not.

Let j be a client that continues with the protocol. Let x_j be the measurement, which belongs to some cluster i. The node generates a Diffie-Hellmann private key a_j and an ephemeral public key $\mathsf{pk}_j = g^{a_j} \bmod p$, which it sends to the server along with the cluster identifier i.

At some time, the server declares that the cluster i is complete and stops accepting new submissions for the cluster. If the cluster contains fewer than k public keys, the protocol stops and the cluster is suppressed and the protocol stops, otherwise the server publishes the list of received public keys:

$$L = \mathsf{pk}_1 \| \ldots \| \mathsf{pk}_j \| \ldots \| \mathsf{pk}_{N_i}$$

where N_i is the number of collected keys. The order of the keys is arbitrary; for the sake of simplicity, we will assume that the key of node j is at position j in the list.

3.4.3 Phase 2

Node j receives from the server the keys preceding and following the node's key in the list, as well as a hash of the list itself, $h(L)$. Then it calculates two pairs of

masking values, by executing DH with the preceding node and with the following node.

The masking value for the mean is calculated as:

$$\mu_j^{(\nu)} = \mathsf{KDF}\left[(\mathsf{pk}_{j+1 \bmod N_i})^{a_j} \bmod p \| i \| h(L) \| \nu\right] - \mathsf{KDF}\left[(\mathsf{pk}_{j-1 \bmod N_i})^{a_j} \bmod p \| i \| h(L) \| \nu\right]$$

with ν being 1, for masking the sum, or 2, for masking the sum of the squared values. Finally, it sends the message $x_j + \mu_j^{(1)} \bmod n \parallel x_j^2 + \mu_j^{(2)} \bmod n$.

The server calculates the sample mean as

$$M_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_i + \mu_i^{(1)} = \frac{1}{N_i} \sum_{j=1}^{N_i} x_i$$

Note that the equality holds because all the masking terms $\mu_i^{(1)}$ cancel out in pairs.

The server proceeds similarly for the sum of the squared values $M_i^{(2)}$ and then calculates the sample variance of the cluster, S_i^2, with the well-known formula:

$$S_i^2 = \frac{\left(\sum_{j=1}^{N_i} x_i^2 + \mu_i^{(2)}\right) - \left(\sum_{j=1}^{N_i} x_i + \mu_i^{(1)}\right)^2 / N_i}{N_i - 1}$$

4 Protocol Evaluation

4.1 Security Analysis

Security against client's attack is trivial. The client receives only a list of DH public keys, which are independent of the measurements. No algorithm can make a guess on b better than a random choice. Consequently, the protocol guarantees security according to Definition 1 with $\epsilon = 0$.

We prove security against server's attack under the assumption that function KDF is a Random Oracle. We discuss the case of the mean; the case for the mean squared value is similar.

Let call $x_{0,1} \dots x_{0,N_i}$ the set of measurements in M_0 and $x_{1,1} \dots x_{1,N_i}$ the set of measurements in M_1. We already know from Definition 2 that the two sets have the same sum. Let call this sum M_Σ, it clearly holds that the measurement for the last client is uniquely determined as $x_{b,N_i} = M_\Sigma - \sum_{j=1}^{N_i-1} x_{b,j}$.

We also know that the masking values for the first $N_i - 1$ clients $\mu_1 \dots \mu_{N_i-1}$ can be considered mutually independent, uniformly distributed random variables. Instead the masking value for the last client is uniquely determined as $\mu_{N_i} = -\sum_{j=1}^{N_i-1} \mu_j \bmod n$.

Phase 2 of the sanitization protocol can be thought as a Vigenère cipher, with each client sending one "letter" of the ciphertext, $c_j = x_j + \mu_j \bmod n$. From the above relations, it is clear that the messages from the first $N_i - 1$ clients are the same as a one-time pad, thus carrying no information about b. On the other hand,

the message of last client is uniquely determined given the messages of the other clients and does not depend on b, in fact:

$$c_{N_i} = x_{b,N_i} + \mu_{N_i} = M_\Sigma - \sum_{j=1}^{N_i-1} c_j$$

Consequently, also the message from the last client does not carry information about b. We can thus say that no algorithm can make any better guess about b than choosing randomly. Since there is a negligible, but not zero, probability that a server could find the masking values by breaking the Diffie-Hellman Key Exchange protocol, we conclude that security is guaranteed according to Definition 2 with some negligible ϵ.

4.2 Overhead Evaluation

There are on the market several platforms for the collection of crowdsourced measurements of network performance (Ookla[18], SamKnows –which is also used by the Measuring Broadband America program of the FCC –[19], or the Misurainternet platform endorsed by the Italian regulatory agency[17]. These platforms provide applications for various operating systems and already include mechanisms for delivering measurements from clients anywhere on the Internet to a central collection database. The proposed distributed sanitization protocol could be integrated in any of those. In this Section, we calculate the expected overhead w.r.t. simple measurement reporting.

We consider two types of overhead with respect to a protocol that simply coveys the measurements from the clients to the server: computational overhead, which increases the amount of operations that the nodes must perform, and the message overhead, which increases the bandwidth consumed by the protocol.

In term of computational overhead, each client, for each reported measurement, must:

1. generate a Diffie-Hellman key pair,
2. calculate two Diffie-Hellman secrets,
3. calculate KDF twice,
4. perform one integer squaring and four sums modulo n.

Step (1) can be reused for multiple measurements and its cost can be amortized. Steps (3) and (4) involve simple operations over integer numbers that are cheap to execute. Consequently, the main protocol cost consists in two modular exponentiations for every measurement of each client. The server only performs simple operations and thus bears small costs. Finally, the querying protocol involves simple comparisons and has negligible costs.

In terms of message overhead:

1. each client sends a Diffie-Hellman public key,
2. each client receives a list of Diffie-Hellman public keys,
3. each client sends two masked measurements modulo n.

The Diffie-Hellman modulo should be at least 2048 bits. Additionally, we assume that the maximum reported measurement is 10^9, for example 1 Tbit/s measured in steps of 1 kbit/s, and the maximum number of reported measurements

in a cluster is 10^5. Therefore, the minimum value for n is 10^{23}, requiring about 80 bits. The hash function h should be at least 256 bit.

Consequently, for each reported measurement, each client sends about 2.2 kilobits and receives about 4.3 kilobits.

5 Privacy Evaluation

This section discusses the conditions guaranteeing (ϵ, δ)-differential privacy, assuming that a single query is submitted to the virtual database X. The case of multiple queries can be handled by applying the composition theorem proved in [8], which states that, in presence of Q consecutive queries, the system guarantees $(Q\epsilon, Q\delta)$-differential privacy.

5.1 General Case

The most general assumption is that, for every tuple belonging to each cluster, the outcome variable is characterized by an unknown probability distribution with density function $f_{\gamma_i}(x)$, being i the cluster label.

We now state the following theorem.

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

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

$$\delta = \left[1 - F_{\mathrm{Bi}}(N_i, \beta, k-1)\right] F_{\mathrm{Bi}}(N_i - 1, \beta, k-1) + F_{\mathrm{Bi}}(N_i, \beta, k-1)\left[1 - F_{\mathrm{Bi}}(N_i - 1, \beta, k-1)\right] \tag{3}$$

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 (12), depends on $f_{\gamma_i}(x)$ and approaches zero as N_i grows. The function $F_{\mathrm{Bi}}(N_i, \beta, k)$ is the binomial cumulative density function for k successes out of N_i trials with success probability β.

Proof The proof is provided in the Appendix, under the slightly simplifying assumption that the variance of the data in the cluster is known and is not calculated from the data themselves.

Note that, when N_i grows, $\mathcal{K}'$ becomes negligible, which allows ϵ to be bounded as:

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

In Figure 3 we report the lower bound of ϵ in (4) for multiple values of β and N_i, under assumption that the minimum cluster size is k, i.e. $N_i \geq k\ \forall i$.

In Figure 4 we plot the value of δ in (3) for different values of β, assuming cluster sizes of $N_i = 100$, $N_i = 1000$, and $N_i = 10000$. Results show that the probability δ depends on the product βN_i, i.e. on the expected number of elements that survive in the cluster after sampling, and is negligible when βN_i is either much smaller or much larger than k. Conversely, when $N_i \sim k/\beta$, δ rises consistently: in this case, a limit must be imposed on the maximum number of queries Q to guarantee a sufficient privacy level.

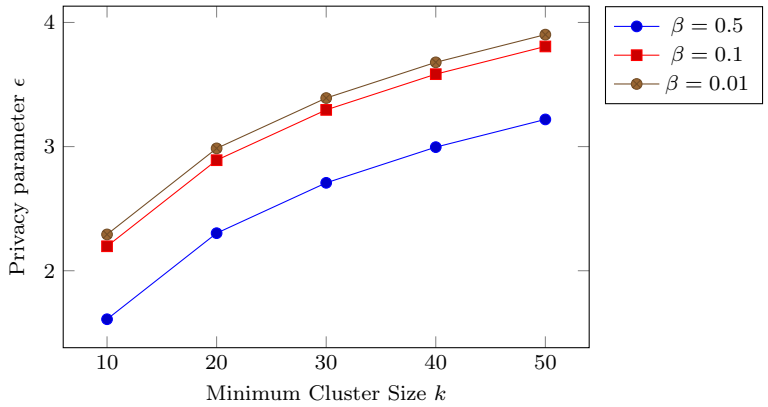

Fig. 3 Upper bound of the privacy parameter ϵ in the general case.

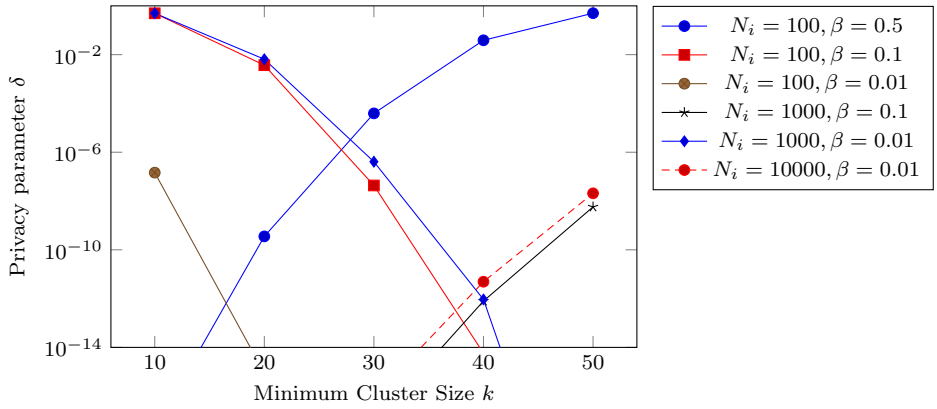

Fig. 4 Privacy parameter δ in the general case for various values of the sampling factor β and for clusters of sizes $N_i = 100$, $N_i = 1000$, and $N_i = 10000$ elements

5.2 Gaussian Model

With Theorem 1 we have proved that our proposed sanitization algorithms ensures differential privacy, but the bounds on the value of ϵ obtained with Equation (4) and reported in Figure 3 are of little practical use, as they are often too loose, especially for larger clusters (high values of N_i). A tighter and more useful estimation can be computed under stronger assumptions on the statistical distribution of the clustered data. Therefore, in the remainder of the section, we re-evaluate equation (5) assuming that data are normally distributed, i.e., using Gaussian distribution functions with mean value μ and standard deviation σ. We use realistic parameter values associated to two widely diffused services: file downloading and video streaming, which will also be considered in the numerical assessment provided in Section 6.

In order to numerically assess the performance of the sanitization algorithm, we generate multiple realizations of X and $X \backslash x_r$ by randomly subsampling the available data. Then, we apply the sanitization mechanism with different value of

β and k in order to obtain the ϵ bounds in Figure 5, which reports the resulting gain in privacy level versus increasing anonymization levels. We average the different simulations and verify the ratio in (1) for several queries q, always choosing the maximum value over all the queries. Results show that the obtained bounds are at least three order of magnitude lower than those obtained in the case of unknown probabilistic distributions of the clustered data.

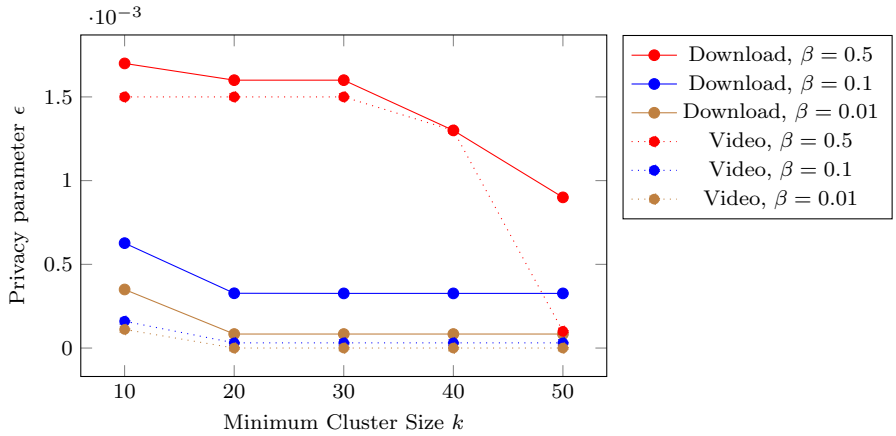

Fig. 5 Privacy parameter ϵ in the case of normally distributed data and cluster size $N_i = 100$

6 Numerical Assessment

6.1 Validation Method

For the numerical assessment of the privacy-precision trade-off achieved by our proposed sanitization approach we use the September 2013 measurement dataset collected by SamKnows in the project *Measuring Broadband America* [9]. Data had already been subjected to an anomaly removal procedure (e.g. by eliminating out of range IP addresses or throughput measurement that were inconsistent with the service tier provisioned by the ISP). We consider the following services: web browsing and video streaming. For each service, the tuple of considered attributes includes service tiers (i.e., ISP, download speed, upload speed). position (i.e., longitude and latitude), time and total throughput. Based on the above mentioned attributes, 50 millions database entries are classified in different categories, according to the granularities reported in Table 1, where we differentiate the attributes as treatment, confounder and outcome variables.

For the validation procedure, we randomly select an ISP and simulate the application of a traffic filtering policy by imposing a 1 Mbit/s threshold on the maximum throughput. A boolean variable is used as *ground truth* and set to True if traffic discrimination is applied (i.e., if the throughput value in the tuple exceeds the imposed threshold), to False otherwise (i.e., if the throughput value in the tuple is below the threshold value). The entries of the measurement database

Table 1 Variables and Generalization Rules

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

are randomly partitioned in a training set (which includes 85% of the available entries) and a test set (including the remaining 15%). The data belonging to the training set are clustered and sanitized via our proposed method, thus obtaining the sanitized database X'. We consider a sampling rate β equal to 0.1% or larger, resulting in a surviving data set of about 400,000 flows, before the suppression of small clusters. This is consistent with the work by Tariq *et al.* [21], which considers a subsample of 100,000 flows. We expect that our approach is less effective than Tariq *et al.* because elimination of small clusters is data-dependent and because we make decisions for each ISP-client pair, instead of just per-ISP. Numerical results in the next section show that, in order to obtain results close to the baseline, we need at a sampling rate β at least 1%.

6.2 Results

We first assess a benchmark performance of the proposed NN violation detection mechanism by applying it to the full measurement dataset, without previous sanitization. Results obtained for this scenario are reported as "baseline"in the following Figures. The obtained precision, computed as the ratio of the correct NN violation detections to the total number of detections, depends on the type of service: it reaches 58% for the file download services, whereas it is only 29% for the video streaming service. The recall, defined as the ratio of the number of detections to the total number of violations, exceeds 99% for the file downloading service but only reaches 55% for the video streaming service. This is motivated by the fact that the latter service includes heterogeneous video contents characterized by different resolutions, and that the streaming speed also depends on the receiver device. Other confound variables, not available in the Measuring Broadband America dataset, should therefore be considered in order to increase the probability of correct detection for the video streaming service. Note also that the above reported results, which may appear unsatisfactory, are obtained considering a single query, whereas real implementations of the detection mechanism should consider multiple queries over a suitably long time period, to ensure statistical significance. However, the choice of the observation interval and of the number of queries is out of the scope of this paper.

We then apply our proposed NN violation detection mechanism after performing database sanitization: the precision obtained for different values of the

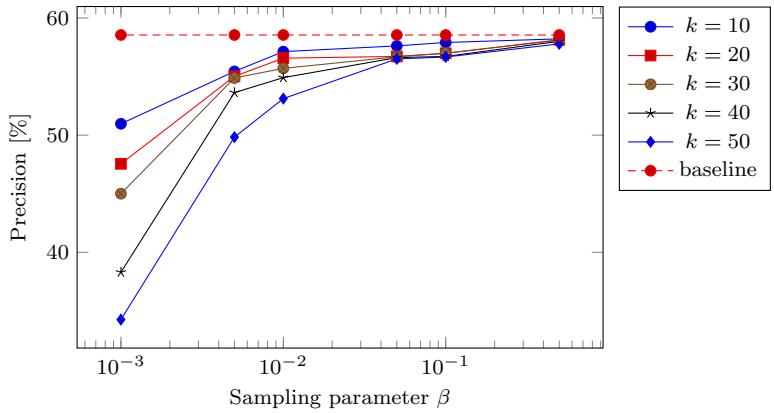

Fig. 6 Precision versus the sanitization parameters for the downloading service

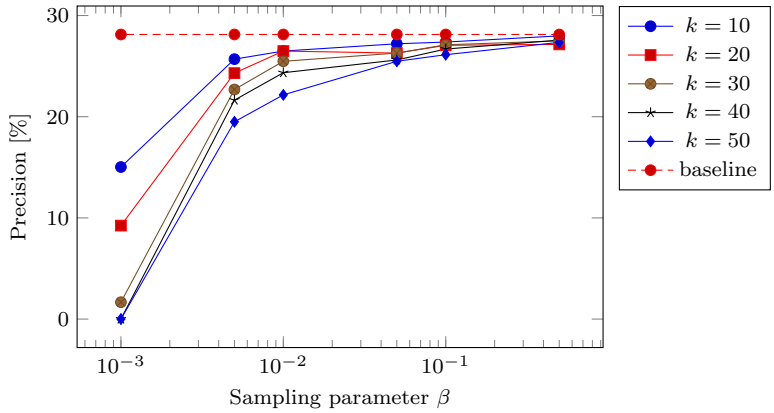

Fig. 7 Precision versus the sanitization parameters for the video streaming service

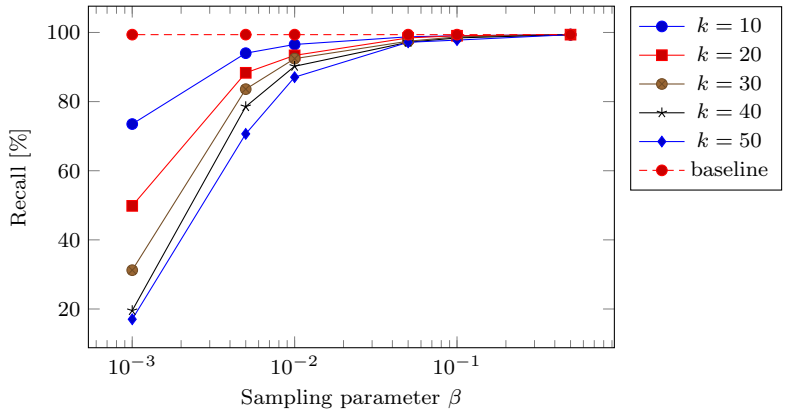

Fig. 8 Recall versus the sanitization parameters for the downloading service

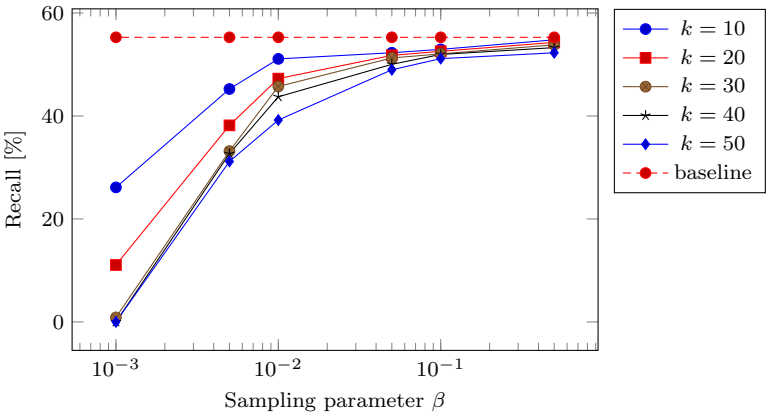

Fig. 9 Recall versus the sanitization parameters for the video streaming service

sanitization parameters β and k is plotted in Figures 6 (for file downloading) and 7 (for video streaming) and compared to the baseline results. The Figures show that precision increases when the sampling parameter β grows and the cluster size k diminishes: precision is heavily reduced when $\beta \leq 0.1\%$ but closely approaches the baseline values when $\beta \geq 10\%$, for all the considered values of k.

Conversely, recall results for different values of the sanitization parameters β and k are reported in Figures 8 and 9, respectively for the file downloading and video streaming services. In both cases, the trend is analogous to that of precision. We conclude that values of β between $1\% - 10\%$ and values of k around 30 ensure the best tradeoff between accuracy of NN violation identifications (in terms of precision and recall) and privacy bounds (in terms of ϵ and δ).

It is worth noting that, in our experiment design, we arbitrarily assume that the violation detection check always returns a negative outcome to queries for measurements belonging to suppressed clusters. A different choice would have lead to worse results in terms of precision, especially when β is small or k is large (i.e., when the number of suppressed clusters is large).

7 Conclusion

We describe an algorithm for the crowdsourced detection of possible net neutrality violations and a protocol to collect the necessary data. The protocol applies cryptographic encryption to prevent the server from obtaining per-user measurements and applies data sanitization 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 evaluate the tradeoff between effectiveness of the detection algorithm and the achieved privacy level by using a large dataset of measurements of broadband traffic by home users. A little data subsampling along with the elimination of very

small clusters is capable of providing minimal accuracy 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 to which x_r belongs. 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 $X \backslash x_r$;
2. if $Y \geq k$ and $Z \geq k$, then no cluster is removed;
3. otherwise, the cluster is removed only in one database, either in X or in $X \backslash x_r$.

Case 1. The cluster is removed from both databases Then $\mathcal{A}(q, X) = \mathcal{A}(q, X \backslash x_r) = 0$ and inequality (1) is always true.

Case 2. No cluster is removed We first prove 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

$$-\sigma < \frac{1}{Y} \sum_{j=k}^{Y} y_j - q < \sigma \quad \forall q$$

$$-\sigma < \frac{1}{Z} \sum_{j=k}^{Z} z_j - q < \sigma \quad \forall q$$

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:

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

First, 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}. The latter can be written as:

$$\phi_{S_{N_i}}(\omega) = \sum_{j=k}^{Y} \left[\phi_{\gamma_i}\left(\frac{\omega}{j}\right)\right]^j \binom{N_i}{j} \beta^j (1-\beta)^{N_i - j}$$

Equation (6) can be rewritten as:

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{-j(q-\sigma)\omega}-e^{-j(q+\sigma)\omega}}{j\omega}\left[\phi_{S_{N_i}}(\omega)-e^{\epsilon}\phi_{S_{N_i-1}}(\omega)\right]d\omega\leq 0 \tag{7}$$

Note that we can write:

$$\phi_{S_{N_i}}(\omega)-e^{\epsilon}\phi_{S_{N_i-1}}(\omega)=\beta^{N_i}\phi_{\gamma_i}\left(\frac{\omega}{N_i}\right)^{N_i}+ \sum_{j=k}^{N_i-1}\phi_{\gamma_i}\left(\frac{\omega}{j}\right)^{j}\beta^{j}(1-\beta)^{N_i-1-j}\binom{N_i-1}{j}\left[\frac{N_i(1-\beta)}{N_i-j}-e^{\epsilon}\right] \tag{8}$$

Substituting (8) into (7), we obtain:

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{-j(q-\sigma)\omega}-e^{-j(q+\sigma)\omega}}{j\omega}\left[\beta^{N_i}\phi_{\gamma_i}\left(\frac{\omega}{N_i}\right)^{N_i}+ \sum_{j=k}^{N_i-1}\phi_{\gamma_i}\left(\frac{\omega}{j}\right)^{j}\beta^{j}(1-\beta)^{N_i-1-j}\binom{N_i-1}{j}\left[\frac{N_i(1-\beta)}{N_i-j}-e^{\epsilon}\right]\right]d\omega\leq 0 \tag{9}$$

$$\frac{1}{2\pi}\left\{\int_{-\infty}^{\infty}\frac{e^{-j(q-\sigma)\omega}-e^{-j(q+\sigma)\omega}}{j\omega}\beta^{N_i}\phi_{\gamma_i}\left(\frac{\omega}{N_i}\right)^{N_i}d\omega\ + \int_{-\infty}^{\infty}\frac{e^{-j(q-\sigma)\omega}-e^{-j(q+\sigma)\omega}}{j\omega}\sum_{j=k}^{N_i-1}\phi_{\gamma_i}\left(\frac{\omega}{j}\right)^{j}\beta^{j}(1-\beta)^{N_i-1-j}\cdot \binom{N_i-1}{j}\left[\frac{N_i(1-\beta)}{N_i-j}-e^{\epsilon}\right]d\omega\right\}\leq 0 \tag{10}$$

We set $\mathcal{K}=\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{-j(q+\sigma)\omega}-e^{-j(q-\sigma)\omega}}{j\omega}\beta^{N_i}\phi_{\gamma_i}\left(\frac{\omega}{N_i}\right)^{N_i}d\omega$ that is always positive. Then, for the purposes of the inequality less or equal to zero, the last case $j=N_i-1$ of the summation gives the most significant contribution with respect to the previous terms, which assume lower values.
We can therefore simplify (10) as:

$$(N_i-1)\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{-j(q-\sigma)\omega}-e^{-j(q+\sigma)\omega}}{j\omega}\beta^{N_i-1}\phi_{\gamma_i}\left(\frac{\omega}{N_i-1}\right)^{N_i-1}\cdot \left[N_i(1-\beta)-e^{\epsilon}\right]d\omega\leq-\mathcal{K} \tag{11}$$

We set then the constant $\mathcal{K}'$ as:

$$\mathcal{K}'=\frac{\mathcal{K}}{\frac{(N_i-1)}{2\pi}\int_{-\infty}^{\infty}\frac{e^{-j(q-\sigma)\omega}-e^{-j(q+\sigma)\omega}}{j\omega}\beta^{N_i-1}\phi_{\gamma_i}\left(\frac{\omega}{N_i-1}\right)^{N_i-1}d\omega} =\frac{\beta\int_{-\infty}^{\infty}\frac{e^{-j(q-\sigma)\omega}-e^{-j(q+\sigma)\omega}}{j\omega}\phi_{\gamma_i}\left(\frac{\omega}{N_i}\right)^{N_i}d\omega}{(N_i-1)\int_{-\infty}^{\infty}\frac{e^{-j(q-\sigma)\omega}-e^{-j(q+\sigma)\omega}}{j\omega}\phi_{\gamma_i}\left(\frac{\omega}{N_i-1}\right)^{N_i-1}d\omega} \tag{12}$$

So we can obtain the new equation:

$$e^{\epsilon} \geq N_i(1-\beta) + \mathcal{K}' \quad \forall\, N_i \geq k \tag{13}$$

Second, we consider the left inequality of (5), for which we have:

$$F_{S_{N_i}}(q+\sigma) - F_{S_{N_i}}(q-\sigma) - e^{-\epsilon}F_{S_{N_i-1}}(q+\sigma) + e^{-\epsilon}F_{S_{N_i-1}}(q-\sigma) \geq 0 \tag{14}$$

Following the same steps as before, we get:

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e^{-j(q-\sigma)\omega} - e^{-j(q+\sigma)\omega}}{j\omega}\big[\phi_{S_{N_i}}(\omega) - e^{-\epsilon}\phi_{S_{N_i-1}}(\omega)\big]d\omega \geq 0 \tag{15}$$

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e^{-j(q-\sigma)\omega} - e^{-j(q+\sigma)\omega}}{j\omega}\left[\beta^{N_i}\phi_{\gamma_i}\left(\frac{\omega}{N_i}\right)^{N_i} + \sum_{j=k}^{N_i-1}\phi_{\gamma_i}\left(\frac{\omega}{j}\right)^{j}\beta^{j}(1-\beta)^{N_i-1-j}\binom{N_i-1}{j}\left(\frac{N_i(1-\beta)}{N_i-j} - e^{-\epsilon}\right)\right]d\omega \geq 0 \tag{16}$$

Keeping constants $\mathcal{K}$ and $\mathcal{K}'$ as in (12), we obtain:

$$e^{-\epsilon} \leq N_i(1-\beta) + \mathcal{K}' \quad \forall\, N_i \geq k \tag{17}$$

From (13) and (17) we can finally obtain the ϵ bounds:

$$\epsilon \geq \log\big(N_i(1-\beta) + \mathcal{K}'\big) \quad \forall N_i \geq k \tag{18}$$

$$\epsilon \geq -\log\big(N_i(1-\beta) + \mathcal{K}'\big) \quad \forall N_i \geq k \tag{19}$$

The first bound is stricter than the second one, which can be ignored.

Case 3. The cluster is removed only in one database Reminding that the system is set up to respond False (0) if it is impossible to find the cluster to which the measure under review belongs, if $b = 0$ the proof is the same of *Case 1.*

Instead, the proof is more elaborated when $b = 1$, i.e., if:

$$\frac{1}{Y}\sum_{j=k}^{Y} y_j - q > \sigma \qquad \text{or} \qquad \frac{1}{Y}\sum_{j=k}^{Y} y_j - q < -\sigma$$

$$\frac{1}{Z}\sum_{j=k}^{Z} z_j - q > \sigma \qquad \text{or} \qquad \frac{1}{Z}\sum_{j=k}^{Z} z_j - q < -\sigma$$

In case the cluster is removed in only one of the two databases, we can have either $\int_{q-\sigma}^{q+\sigma} f_{S_{N_i}}(x)dx = 1$ or $\int_{q-\sigma}^{q+\sigma} f_{S_{N_i-1}}(x)dx = 1$, because the output of the algorithm will always be False. This results in:

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

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

In this scenario the ϵ-differential privacy cannot be satisfied. It is possible to relax the privacy definition assuming that with a small probability δ the inequality (1) can be violated.
The (ϵ, δ)-differential privacy may be satisfied with the same ϵ calculated in the previous cases and with a small value of δ that reflects the probability of dropping a cluster in one of the two databases, X or $X \backslash x_r$, given the response True.
The probability of dropping a cluster of size N_i is the binomial probability of taking at most k elements over N_i:

$$F_{\mathrm{Bi}}(N_i, \beta, k) = \sum_{j=0}^{k} \binom{N_i}{j} \beta^j (1-\beta)^{N_i - j}$$

Since the probability of dropping a cluster only in one database and the probability of getting output True are independent, we can define δ as:

$$\begin{aligned}
\delta = Pr\{Y \geq k, Z < k\} + Pr\{Y < k, Z \geq k\} = \\
\left[1 - \sum_{l=0}^{k-1} \binom{N_i}{l} \beta^l (1-\beta)^{N_i - l}\right] \sum_{j=0}^{k-1} \binom{N_i - 1}{j} \beta^j (1-\beta)^{N_i - 1 - j} + \\
\sum_{l=0}^{k-1} \binom{N_i}{l} \beta^l (1-\beta)^{N_i - l} \left[1 - \sum_{j=0}^{k-1} \binom{N_i - 1}{j} \beta^j (1-\beta)^{N_i - 1 - j}\right] = \\
[1 - F_{\mathrm{Bi}}(N_i, \beta, k-1)] F_{\mathrm{Bi}}(N_i - 1, \beta, k-1) + \\
F_{\mathrm{Bi}}(N_i, \beta, k-1)[1 - F_{\mathrm{Bi}}(N_i - 1, \beta, k-1) \quad (20)
\end{aligned}$$

References

1. Akamai: Real user monitoring. www.akamai.com/uk/en/resources/real-user-monitoring.jsp
2. Anjum, A., Anjum, A.: Differentially private k-anonymity. In: Frontiers of Information Technology (FIT), 2014 12th International Conference on, pp. 153–158 (2014). DOI 10.1109/FIT.2014.37
3. Antonopoulos, A., Kartsakli, E., Perillo, C., Verikoukis, C.: Shedding light on the internet: Stakeholders and network neutrality. IEEE Communications Magazine **55**(7), 216–223 (2017)
4. Botta, A., Avallone, A., Garofalo, M., Ventre, G.: Internet streaming and network neutrality: Comparing the performance of video hosting services. In: ICISSP, pp. 514–521 (2016)
5. De Martin, J.C., Glorioso, A.: The neubot project: A collaborative approach to measuring internet neutrality. In: 2008 IEEE International Symposium on Technology and Society, pp. 1–4. IEEE (2008)
6. Dischinger, M., Marcon, M., Guha, S., Gummadi, P.K., Mahajan, R., Saroiu, S.: Glasnost: Enabling end users to detect traffic differentiation. In: NSDI, pp. 405–418 (2010)
7. Dwork, C.: Differential privacy. In: Automata, languages and programming, pp. 1–12. Springer (2006)
8. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science **9**(3-4), 211–407 (2014)
9. FCC: Validated data – Measuring broadband America 2014. www.fcc.gov/general/validated-data-measuring-broadband-america-2014 (2014)
10. FCC: Open internet. https://www.fcc.gov/general/open-internet (2015)
11. Garrett, T., Dustdar, S., Bona, L.C., Duarte, E.P.: Ensuring network neutrality for future distributed systems. In: Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, pp. 1780–1786. IEEE (2017)

12. Kanuparthy, P., Dovrolis, C.: Diffprobe: detecting isp service discrimination. In: INFOCOM, 2010 Proceedings IEEE, pp. 1–9. IEEE (2010)
13. Kasiviswanathan, S.P., Smith, A.: A note on differential privacy: Defining resistance to arbitrary side information. CoRR abs/0803.3946 (2008)
14. Li, N., Qardaji, W.H., Su, D.: Provably private data anonymization: Or, k-anonymity meets differential privacy. Arxiv preprint (2011)
15. Maltinsky, A., Giladi, R., Shavitt, Y.: On network neutrality measurements. ACM Trans. Intell. Syst. Technol. **8**(4), 56:1–56:22 (2017)
16. Miorandi, D., Carreras, I., Gregori, E., Graham, I., Stewart, J.: Measuring net neutrality in mobile internet: Towards a crowdsensing-based citizen observatory. In: 2013 IEEE International Conference on Communications Workshops (ICC), pp. 199–203 (2013). DOI 10.1109/ICCW.2013.6649228
17. Misurainternet. Il progetto italiano per la valutazione della qualità dell'accesso a internet da postazione fissa. www.misurainternet.it (in Italian) (2018)
18. Ookla. The definitive source for global internet metrics. www.ookla.com (2018)
19. Samknows. www.samknows.com (2018)
20. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05), 557–570 (2002)
21. Tariq, M.B., Motiwala, M., Feamster, N., Ammar, M.: Detecting network neutrality violations with causal inference. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pp. 289–300. ACM, New York, NY, USA (2009). DOI 10.1145/1658939.1658972
22. Zhang, Z., Mara, O., Argyraki, K.: Network neutrality inference. SIGCOMM Comput. Commun. Rev. **44**(4), 63–74 (2014). DOI 10.1145/2740070.2626308