z-anonymity: Zero-Delay Anonymization for Data Streams

z-anonymity: Zero-Delay Anonymization for Data Streams

Nikhil Jha, Thomas Favale, Luca Vassio, Martino Trevisan, Marco Mellia
Politecnico di Torino
first.last@polito.it

***Abstract*—With the advent of big data and the birth of the data markets that sell personal information, individuals' privacy is of utmost importance. The classical response is anonymization, i.e., sanitizing the information that can directly or indirectly allow users' re-identification. The most popular solution in the literature is the k-anonymity. However, it is hard to achieve k-anonymity on a continuous stream of data, as well as when the number of dimensions becomes high.**

In this paper, we propose a novel anonymization property called z-anonymity. Differently from k-anonymity, it can be achieved with zero-delay on data streams and it is well suited for high dimensional data. The idea at the base of z-anonymity is to release an attribute (an atomic information) about a user only if at least $z-1$ other users have presented the same attribute in a past time window. z-anonymity is weaker than k-anonymity since it does not work on the combinations of attributes, but treats them individually. In this paper, we present a probabilistic framework to map the z-anonymity into the k-anonymity property. Our results show that a proper choice of the z-anonymity parameters allows the data curator to likely obtain a k-anonymized dataset, with a precisely measurable probability. We also evaluate a real use case, in which we consider the website visits of a population of users and show that z-anonymity can work in practice for obtaining the k-anonymity too.

***Index Terms*—Anonymization, data streams, scalability, zero delay, k-anonymity.**

I. Introduction

Big data have opened new opportunities to collect, store, process and, most of all, monetize data. This has created tension with privacy, especially when it comes to information about individuals. We live in the data era, where a big part of our life is readily available in digital format, from our online activity to our location history, from what we buy to how we spend out free time [1]. Recently, legislators have introduced privacy laws to regulate the data collection and market, with notable examples of the General Data Protection Regulation (GDPR) in EU, or the California Consumer Privacy Act (CCPA) in the US.

The classical approach to publish personal information is anonymization, i.e., generalizing or removing data of the most sensitive fields. Thanks to this, Privacy-Preserving Data Publishing (PPDP) has gained attention in the last decade [2]. It is now even more popular (and critical) with the birth of data markets where data buyers can have access to large collections of data about individuals. Removing the user's *identifiers* (name, social security number, phone number, etc.) is not sufficient to make a dataset anonymous. Indeed, an attacker can link a user's apparently harmless attributes (such as gender, zip code, date of birth, etc.) called *quasi-identifiers* (QIs) to a (possibly even public) background knowledge. In this way, the attacker can re-identify the person and gain access to further sensible information from the dataset (disease, income, etc.) called *sensitive attributes* (SAs) [3]. Famous is the de-anonymization of Netflix public dataset [4] based on the study of QIs.

Researchers proposed several properties that anonymized data should respect to avoid re-identification, the most popular of which is the k-anonymity [5], or k-anon for short. Despite its limits, it remains the golden standard for anonymization. k-anon imposes that the information of each person contained in the release cannot be distinguished from at least $k-1$ individuals whose information also appears in the release. k-anon is conceived for tabular and static data. In other words, the dataset must be completely available at anonymization-time. Extensions to a streaming scenario have been proposed, where continuously incoming records are processed, typically using sliding windows [6]. In this case, the new records are temporarily stored, processed and released after an unavoidable delay. However, for specific applications it is fundamental to avoid any processing delay. For example, for network traffic, where it is unfeasible to store packets for a long time, or location history, if a real-time (but anonymous) stream shall be used for, e.g., mobility optimization.

This paper proposes a novel anonymization property called z-anonymity, or z-anon for short. It is designed to work with data streams and can be achieved with zero-delay (hence the choice of the letter z instead of k). We assume to observe a raw stream of data, in which users' new attributes are published in real-time as they are generated. For instance, a new transaction in their credit card, a new position of their car, or a new website they visit. These attributes are QIs, and, when accumulated over time, may allow users' re-identification.

z-anon builds on the same idea of k-anon. When a new attribute arrives, it is released only if at least $z-1$ individuals have presented the same attribute in the past window Δt. Otherwise, it is blurred. z-anon is weaker than k-anonymity since it cannot guarantee that at least $k-1$ users present the same *combinations* of QIs (i.e., the aggregated record). Implementing z-anon in real-time at high speed requires

The research leading to these results has been funded by the European Union's Horizon 2020 research and innovation program under grant agreement No. 871370 (PIMCity) and the SmartData@PoliTO center for Big Data technologies.

ingenuity, especially considering the large number of attributes the system deals with - i.e., the high-dimensional data problem, which is one of the problems hampering k-anon too [7]. In this paper, we show that z-anon can be obtained both with zero-delay and in an efficient way when employing a scalable implementation and appropriate data structures. Lastly, we present a probabilistic framework to map z- into k-anon properties. We find out that z-anon can provide k-anon with desired probability, for appropriate values of z.

There are various examples of application of z-anon. For instance, we originally proposed it for internet traffic analysis, where high-speed passive monitors process packets that contain QIs (e.g., hostnames of visited websites) in real time [8]. Similarly, the user browsing history, the credit card history, and the location history offer rich information that companies want to access as quickly as possible, i.e., datum after datum, without waiting for records to be aggregated. For instance, recent credit card transactions can be useful for fraud detection or shopping recommendations; the browsing history for personalized advertisements or market intelligence; the location history to promptly optimize the mobility, or study patterns of real-time traffic.

In the remainder of the paper, after presenting the related work (Section II), we formalize the z-anon property and present an approach to implement it efficiently and in real-time (Section III). We then propose a probabilistic model to derive k-anon properties from z-anonymized streams (Section IV), and study the effect of the different parameters (Section V). We then apply the model to the browsing history use case (Section VI). Finally, we discuss the limitations of our approach and future work (Section VII) and draw the conclusions (Section VIII).

II. Related work

The problem of providing anonymization guarantees to dynamic datasets arose together with the increasing attention towards PPDP. Several approaches have been proposed during the years, that we can roughly group in microaggregation, input and output perturbation, generalization and suppression, clustering-based and tree-based techniques.

Microaggregation techniques ([9], [10]) group the data and release an aggregated version of them, so that the user's sensitive attributes are not released as is. Input perturbation ([10], [11]) methods aim at adding noise to the incoming data, while output perturbation techniques ([12], [13]) generally modify the output so that it is not possible to link a user by means of a sensitive attribute with high confidence.

Other methods directly emerged from the k-anon concept where a user is indistinguishable from at least other $k-1$ users in the release. Authors of [14] propose two algorithms using suppression and generalization to avoid a correlation analysis from items of a transaction. Achieving k-anon is not trivial with high-dimensional data where the number of possible combinations of attributes explodes. Popular approaches are based on trees ([15]–[17]) or clustering ([6], [18]–[21]). The rationale is the same: firstly load the incoming records in a structure (either a tree or a cluster) and secondly release those tuples when k-anon is achieved, while maintaining a trade-off with information loss.

The majority of the previous methods, however, works with the concept of sliding window, i.e., the incoming data is accumulated, then processed, and finally released with a certain delay. Some efforts have been spent to reduce the delay as far as possible: authors of [16] include the delay in the concept of output quality, with a trade-off between data quality and batch size.

To the best of our knowledge, the only work that approached the problem of zero-delay anonymization is [12], where the authors propose an output perturbation approach. When a sensitive attribute arrives, it is published along with other $s-1$ other different sensitive attributes, so that the attacker can find it with probability not higher than $1/s$. Here, differently, we propose that an attribute is published only if at least other $z-1$ users exhibit the same attribute in the past Δt. In the following, we formalize the concept of z-anon, that we previously empirically adopted in the context of live packet stream monitoring [8]. We generalize our approach and present a probabilistic framework to observe to what extent the z-anon property allows to satisfy also k-anon.

III. z-anonymity

A. Requirements

Our goal is to define an anonymization property for data organized in streams that can be achieved with zero delay. Concisely, we seek at defining an anonymization strategy with the following requirements:

- **Data streams:** we assume that observations arrive continuously in a stream. As such, we shall anonymize them based on a limited view. We do not know the future data, and we can only keep a (limited) memory of the past.
- **Zero delay:** it shall be possible to achieve the anonymization property without any delay for publishing the anonymized stream. In other words, we want to make an atomic decision. All approaches based on the processing of batches of observations are not applicable, as they need to store and process the entire batch before the release.
- **Efficient algorithm for high dimensional data:** the anonymization property shall be achieved with an efficient algorithm, allowing deployment on high speed and a large volume of data with off-the-shelf computing capabilities. It is important to carefully build an algorithm working with efficient data structures too, in order to obtain the necessary information as quickly as possible. Moreover, users might expose a large set of attributes, whose number is not known *a priori*.

B. The z-anonymity approach

We work on a data stream, in which we continuously receive observations that associate users with a value of an attribute. We define an observation as (t, u, a), which indicates that, at time t, the user u exposes an attribute-value pair

a.[1] For example, if *Sex* is the attribute, and *Female* is the value assumed by the attribute of user u at time t, then a is the pair *(Sex, Female)*. Attributes can be related to whatever field: a visit to a web page, a GPS location, a purchase, etc. We consider attributes a as *quasi-identifiers*, while *sensitive-attributes* are not present. We want to keep private those values of attributes associated with a small group of users. We define the property of z-private attribute-value as follows:

Definition 1. An attribute-value pair a is z-private at time t if it is associated with less than z users in the past Δt time interval.

Notice that the same attribute a can be both z-private and not z-private at different time t.

If the anonymized dataset hides all z-private attribute-value pairs, it achieves z-anon.

Definition 2. A stream of observations is z-anonymized if all z-private attribute-value pairs are obfuscated, given z and Δt.

In other words, the attributes that are associated with less than z users in the past Δt shall be obfuscated, i.e., removed or replaced with an empty identifier. The goal is to prevent rare values of attributes to be published, thus reducing the possibilities of an attacker to re-identify a user through unusual attributes.

We exemplify a data stream and the z-anon mechanism in Figure 1. Assume $z = 3$. At time t_0 user u_0 is the first to expose the attribute-value a_0. The attribute a_0 is z-private at time t_0, hence it shall be obfuscated. Still, the information that u_0 exposed the attribute a_0 is kept in memory for a time equal to Δt. At time t_1, user u_1 also exposes a_0. Since the number of observations in Δt is still smaller than 3, the observation is not released. At time t_2 user u_0 re-expose again a_0, extending the lifetime of the observation, but not changing the number of unique users having exposed a_0. At time t_3, user u_2 exposes a_0, making the total users in the past Δt equal to 3. Thus the attribute-value pair a_0 is not z-private at time t_3 and the observation (t_3, u_2, a_0) can be be released. At time $t_1 + \Delta t$ the attribute a_0 related to user u_1 expires, hence the total user count decreases back to 2. The same happens when u_0 observation expires (at $t_2 + \Delta t$), so that when u_3 exposes a_0 at t_4 the observation cannot be released.

In a nutshell, in a stream of incoming data, an observation is released if and only if at least $z - 1$ other users had an observation for the same attribute-value pair in the past Δt time interval.

z and Δt are system parameters that can be tuned to regulate the trade-off between data utility and privacy. This allows z-anon to adapt to the needs of the desired use case, resulting in a flexible paradigm that can be used in many different fields. A large z and a small Δt result in the majority of attributes to be anonymized, while a small z or a large Δt allows rare values to be possibly released. Δt regulates the memory of the system.

[1]Here we will use attribute and attribute-value pair interchangeably.

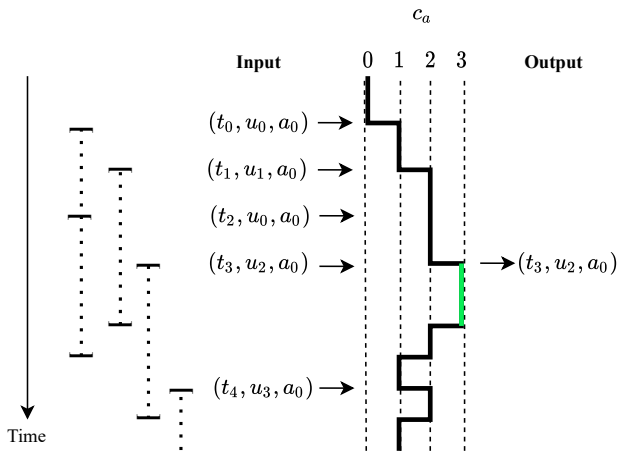


Fig. 1: A graphical example of z-anon concept with $z = 3$: a tuple is released only if other $z - 1 = 2$ different users have exposed the same attribute-value pair in the previous Δt.

It is important to recall that z-anon acts in an attribute-by-attribute fashion, not considering their combinations as in the k-anon property. Hence, it is interesting to study which guarantees the z-anon algorithm offers in a global perspective, i.e., which assumptions it is possible to make on the overall privacy properties (e.g., it terms of k-anon) of the output.

C. Implementation and complexity

The z-anon property can be achieved in real-time with zero delay using a simple algorithm based on efficient data structures. We propose to generalize the approach presented in our previous work [8]: the attribute-value pairs a are stored as a hash table $\mathcal{H}$, with linked lists to manage collisions. Each value $\mathcal{H}(a)$ in the hash table contains three elements:

- metadata about a;
- a Least Recently Used list LRU_a of tuples (t, u);
- a hash table $\mathcal{V}_a$ for the users.

The idea is to minimize the time spent searching into the data structures, therefore reducing the memory accesses. By assuming that the number of attributes a has the same order of magnitude of the hash structure dimension, collisions are infrequent, and consequently, the total computational cost is $O(1)$ for each incoming observation.

The $\mathcal{H}(a)$'s metadata include the counter c_a and the reference for the LRU_a first and last attribute. Referring to Algorithm 1, once an observation (t, u, a) arrives, the value a should be inserted in the hash table, if not already present (lines 2-6), otherwise an update should be performed (lines 7-21). The hash value is calculated and the access to the table is done in $O(1)$.

If the user u comes with attribute a for the first time in the previous Δt, the user u is inserted into $\mathcal{V}_a$ in $O(1)$, c_a is increased by one and the tuple (t, u) is inserted on top of the LRU_a in $O(1)$ thanks to the aforementioned references (lines 8-11). If u was instead already present in $\mathcal{V}_a$ and in LRU_a with value (t', u), we replace t' with t and the tuple (t, u) is

moved on the top of the LRU_a. Again all is done in $O(1)$ (lines 12-14).

Last, to evict old entries and consequently decrease c_a, we traverse the LRU in reverse order: we remove each tuple (t', u') where $t' < t - \Delta t$, and we decrease c_a accordingly (lines 17-21). At last, if $c_a \geq z$ the observation (t, u, a) is released (lines 23-24).

k-anon has been proved [22] an *NP-Hard* problem. Differently, z-anon property can be achieved for each observation with $O(1)$ complexity with properly sized hash-tables.

Algorithm 1 Pseudo code of the algorithm to implement z-anon.

```
1:  Input: (t, u, a)
2:  if a ∉ H then
3:      H ← H ∪ a //new attribute: insert it for the first time
4:      V_a ← {u} //insert new user u
5:      LRU_a ← (t, u)
6:      c_a = 1
7:  else
8:      if u ∉ V_a then
9:          V_a ← V_a ∪ {u} //insert new user u
10:         c_a ← c_a + 1 //add new user
11:         LRU_a ← (t, u)
12:     else
13:         (t', u) ← (t, u) //update timestamp of user u
14:         move (t, u) on top of LRU_a
15:     end if
16: end if
17: //Always evict old users
18: for ((t', u') = last(LRU_a); t' < t − Δt; (t', u')=next) do
19:     remove (t', u') from LRU_a
20:     remove (u') from V_a
21:     c_a ← c_a − 1
22: end for
23: if (c_a ≥ z) then
24:     OUTPUT (t, u, a)
25: end if
```

IV. Modeling z-anonymity and k-anonymity

We now study the relationship between the z-anon and k-anon properties. In particular, we quantify how a z-anonymized dataset could result in a k-anon release with a certain probability. Intuitively, z-anon ensures that each published value of an attribute a is associated at least with z users in the past time interval, while, with k-anon, any given record (i.e., the combinations of all user's attributes) appears in the published data at least k times. Recall that with high-dimensional data, the set of attribute-value combinations becomes extremely high, thus making k-anon tricky to guarantee. Here we show that with a proper choice of z, it is possible to release data in which users are *k*-anonymized.

We define a simple model where users generate a stream of attributes. Each attribute has a given probability of appearance that reflects its different popularity. We assume few attributes are very popular, with a long tail of infrequent attributes that may seldom appear. This often happens in real-world systems that are governed by power-law distributions [23].

A. User and attribute popularity model

We consider a system in which a set of $\mathcal{U}$ users can access a catalog of $\mathcal{A}$ attributes. Let $U = |\mathcal{U}|$ and $A = |\mathcal{A}|$.

Users generate a stream of information, exposing in real-time the attribute they have just accessed. For instance, this reflects a location tracking system in which black boxes installed on a fleet of vehicles periodically exports each car location; or operating system telemetry that periodically reports which application is running; or network meters reporting which website a user is visiting. The system collects *reports* in the form of the tuple (t, u, a), i.e., at time t, the user $u \in \mathcal{U}$ exposes the attribute $a \in \mathcal{A}$. For simplicity, we assume that users are homogeneous and all reports are independent, so that the probability of getting a report, only depends on the value assumed by a.[2] In particular, we assume any user u exposes the attribute a with a given rate λ_a, with exponential inter-arrival time. Hence, given the time interval Δt, the number of times a user exposes an attribute a is modeled as a Poisson random variable R_a with parameter $\lambda_a \cdot \Delta t$ ($R_a \sim Poisson(\lambda_a \cdot \Delta t)$).

We denote as X_a the random variable describing whether a user exposed at least once attribute a in a time interval Δt. X_a assumes value 1 if a user exposes a in Δt, 0 otherwise. We note that $X_a \sim Bernoulli(p_a^X)$, where p_a^X is the probability that a user exposes attribute a at least once in the past Δt. It is straightforward to compute p_a^X given λ_a and Δt as:

$$p_a^X = P[R_a \geq 1] = 1 - P[R_a = 0] = 1 - e^{-\lambda_a \cdot \Delta t} \quad (1)$$

B. Applying z-anon

We study how a stream of data modeled as above appears when released respecting z-anon. With z-anon, z-private attributes at time t are removed. Namely, if less than other $z-1$ users are associated with a in the previous Δt, the current association is blurred. We here define the event of a report (t, u, a) to be published when exposed as a random variable O_a. We have that O_a is a Bernoulli random variable with parameter p_a^O.

$$p_a^O = P[O_a = 1] = P\left[\sum_{v \in \mathcal{U} \setminus u} X_a \geq z - 1\right] \quad (2)$$

Given our assumption of independent and homogeneous users, we are summing $U - 1$ times the same random variable X_a. We remove one user since we are checking the z-anon for the report (t, u, a). Hence one user is already involved by construction. Since X_a is a Bernoulli with success probability p_a^X, its sum results in a Binomial distribution, measuring the number of occurrences in a sequence of $U - 1$ independent experiments $\sum_{v \in \mathcal{U} \setminus u} X_a \sim \mathcal{B}(U - 1, p_a^X)$.

Starting from Equation (2) and using the probability mass function of the Binomial distribution we can derive p_a^O as:

$$p_a^O = 1 - \sum_{i=0}^{z-2} \binom{U-1}{i} \left(p_a^X\right)^i \left(1 - p_a^X\right)^{U-1-i} \quad (3)$$

[2] We can relax this assumption, e.g., by considering classes of users. We leave this for future work.

Similar to what we did in Section IV-A, we denote as Y_a the random variable describing if a user published at least once attribute a in a time interval Δt. We note that $Y_a \sim Bernoulli(p_a^Y)$, where p_a^Y is simply:

$$p_a^Y = P[X_a = 1] \cdot P[O_a = 1] = p_a^X \cdot p_a^O$$

The set of the random variables describing the presence or absence for all the possible attribute-value pairs $a \in \mathcal{A}$ for a user is denoted as $\bar{Y} = \{Y_a\}_{a \in \mathcal{A}}$. Again this is equal for all users, being them homogeneous.

C. The attacker point of view

We assume an attacker observes the z-anonymized output streams for all users $u \in \mathcal{U}$ for a time $N\Delta t$ with $N \in \mathbb{R}^+$ (for simplicity, in our model we considered $N \in \mathbb{N}, N \geq 1$). Hence, in our scenario, the attacker can accumulate the output for a time span possibly much larger than the parameter Δt. Similarly to Y_a, we can thus define the random variable Y_a^N, that models whether a user exposed *and* published attribute a at least once during the total observation period $N\Delta t$. It is clear that Y_a and Y_a^N are strongly related. In fact we have $Y_a^N \sim Bernoulli(p_a^N)$, where the parameter p_a^N can be computed as follows:

$$p_a^N = [1 - (1 - p_a^Y)^N]$$

This is because for a user u to expose and publish an attribute a in the period $N\Delta t$, (s)he has to be associated with a value 1 of Y_a at least in one of the N periods Δt long. At the end of the period $N\Delta t$, the attacker has observed U users hence obtaining U realizations $\overline{y^N}$ of the random variable $\overline{Y^N} = \{Y_a^N\}_{a \in \mathcal{A}}$ including all the possible attributes.

The attacker will not know the random variable $\overline{Y^N}$, and will observe only realizations of it. Let us denote as y_a^N a realization of the random variable Y_a^N and as $\overline{y^N} = \{y_a^N\}_{a \in \mathcal{A}}$ a realization of the random variable $\overline{Y^N}$.

D. Getting to k-anon

We want to check to what extent a z-anonymized stream of a user satisfies also k-anonymity property in the whole stream of U users. Given a specific realization $\overline{y^N}$ of a user, our goal is to derive the probability to observe at least other $k-1$ users in $\mathcal{U}$ having the same realization $\overline{y^N}$. If this happens, the system lets k users release the same attributes and thus they cannot be uniquely re-identified, resulting k-anonymized.

Let us consider first the probability that two realizations of Y_a^N are equal. Let us denote the two realizations, related to two users u and v, as $y_a^N(u)$ and $y_a^N(v)$. The probability is simply $(p_a^N)^2 + (1 - p_a^N)^2$ because either both take the values of 1, or both take the value of 0. Remind that the users are assumed to act independently. The probability that two users have the same realization of $\overline{Y^N}$ is then the following:

$$p^Q = P\left[\overline{y^N(u)} = \overline{y^N(v)}\right] = \prod_{a \in \mathcal{A}} \left((p_a^N)^2 + \left(1 - p_a^N\right)^2\right)$$

TABLE I: Terminology used to model z-anon and k-anon.

$\mathcal{U}, U$	Set and number of users
$\mathcal{A}, A$	Set and number of attribute-value pairs
Δt	The time interval length used for evaluating z-anon
N	Length of the data stream, in multiples of Δt on which we test the k-anonymity
λ_a	Exposing rate for attribute a
R_a	Random variable counting number of times a user exposes attribute a in Δt. $R_a \sim Poisson(\lambda_a \cdot \Delta t)$
X_a	Random variable representing whether a user exposes attribute a in Δt. $X_a \sim Bernoulli(p_a^X)$
O_a	Random variable representing whether a report (t, u, a) is published when exposed. $O_a \sim Bernoulli(p_a^O)$
Y_a	Random variable representing whether a user published at least once attribute a in Δt. $Y_a \sim Bernoulli(p_a^Y)$
Y_a^N	Random variable representing whether a user published at least once attribute a in $N\Delta t$. $Y_a^N \sim Bernoulli(p_a^N)$
$\overline{Y^N}$	Set of random variables $\{Y_a^N\}_{a \in \mathcal{A}}$
Q	Random variable representing whether two realizations of $\overline{Y^N}$ are equal. $Q \sim Bernoulli(p^Q)$
p_{k-anon}	Probability that a realization of $\overline{Y^N}$ satisfies k-anonymity property

where $\overline{y^N(u)}$ and $\overline{y^N(v)}$ are the two realizations of $\overline{Y^N}$. The parameter p^Q can be seen as the parameter of a Bernoulli random variable Q describing whether two realizations are equal (assuming value 1) or not (assuming value 0).

Finally we define the probability that a given realization $\overline{y^N(u)}$ satisfies the k-anonymity property. Hence, it means that there are at least $k-1$ other users with the same realization. We denote this probability as p_{k-anon}.

$$p_{k-anon} = P\left[\sum_{v \in \mathcal{U} \setminus u} Q \geq k - 1\right]$$

Then p_{k-anon} is the probability that at least other $k-1$ realizations are equal to the one studied. Again, as in Equation (2), $\sum_{v \in \mathcal{U} \setminus u} Q$ follows a Binomial distribution of $U-1$ experiments with probability p^Q. Then we can derive p_{k-anon} as in Equation (3):

$$p_{k-anon} = 1 - \sum_{i=0}^{k-2} \binom{U-1}{i} \left(p^Q\right)^i \left(1 - p^Q\right)^{U-1-i}$$

In summary, our model describes the probability that a data stream undergoing z-anon results in dataset which respects the k-anon property. Although we can only provide a probabilistic guarantee that the released data will be k-anonymized, we can study and control this probability as a function of the parameters.

V. Comparing z-anonymity and k-anonymity

In the following, we show the impact of the system parameters on the k-anon and z-anon properties. In our model, we assume a small set of popular attributes and a large tail of infrequent ones. This allows us to catch the nature of systems where users are more likely to expose top-ranked attributes, but there exist a large catalog. As such, we choose that the

TABLE II: The default values used for the model.

Variable	Default Value
U	50 000
A	5 000
λ_{a_r}	0.05 / r
N	24
z	20
k	2

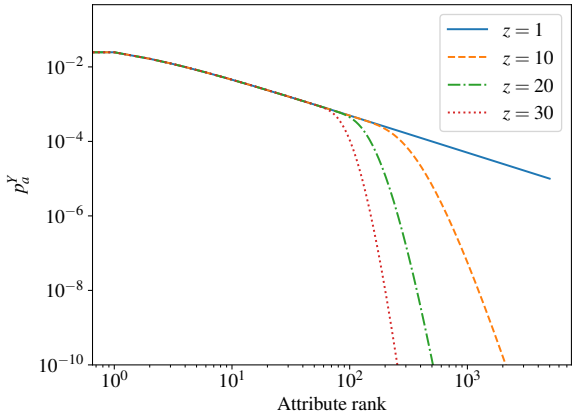


Fig. 2: The probability p_a^Y for a user to publish attribute a in Δt, according to its rank.

λ_a for all attributes follow a power law in function of their rank. Let us suppose attributes are sorted by rank, and the most popular attribute is a_1 and the least popular a_A. We impose $\lambda_{a_1} = 0.05$ and set the remaining λ_a as the power-law function $\lambda_{a_r} = {}^{0.05}/_{r}$, where r is the rank of attribute a_r. The p_a^X value is evaluated as described in Equation (1) - for the sake of simplicity, we consider $\Delta t = 1$ unit of time. Notice that the different attributes are independent and $p_{a_r}^X$ is not a distribution probability mass function, hence it does not have to sum to 1.

We have defined a model that describes the probability that in the released data, satisfying z-anon, a user has at least $k-1$ other users with the same set of associated attributes. Formally speaking, $p_{k-anon} = \mathcal{F}(U, A, \lambda, N, z, k) \rightarrow [0, 1]$. As such, $\mathcal{F}$ gives the probability a generic user is k-anonymized in the released data. Each of the above parameters has an impact on the output probability p_{k-anon}. Here, we study the impact of different combinations of parameters. Where not otherwise noted, the default parameters listed in Table II are used.

A. The impact of the attribute rank

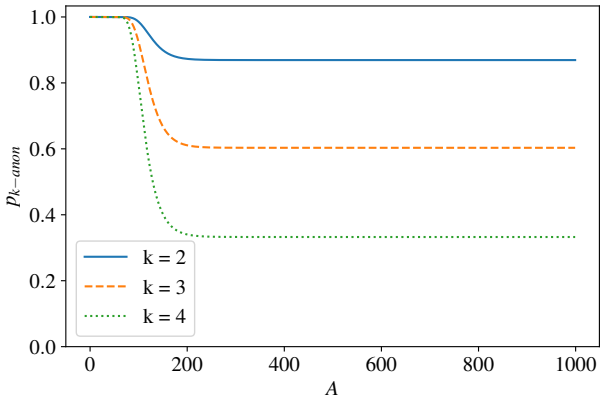
We first focus on the p_a^Y, i.e., the probability of observing at least once the attribute a in a Δt, for a given user, in the released data, after z-anonymization. Figure 2 shows the p_a^Y in function of the attribute rank. Remind that the popularity of attributes follows a power law, since $\lambda_{a_r} \approx r^{-1}$. Indeed, the blue solid line in the figure shows the probability of observing an attribute in case $z = 1$, i.e., no anonymization (equal to p_a^X). The curve appears as a straight line, representing a power law on the log-log plot. When enabling z-anon ($z > 1$), we notice that the probability of observing uncommon attributes abruptly decreases with an evident knee. For example, if we observe the curve for $z = 20$ (green dashed line in the figure), already the 300^{th}-ranked attribute is observed with a probability below 10^{-6}, while it appears on the original stream with 10^{-3}. A higher z moves the knee of the curve closer to the top-ranked attributes. In other words, the figure shows how z-anon operates in preventing uncommon attributes from being released. Indeed, those attributes are released only when enough users are exposing them, hence very rarely.

B. The impact of A

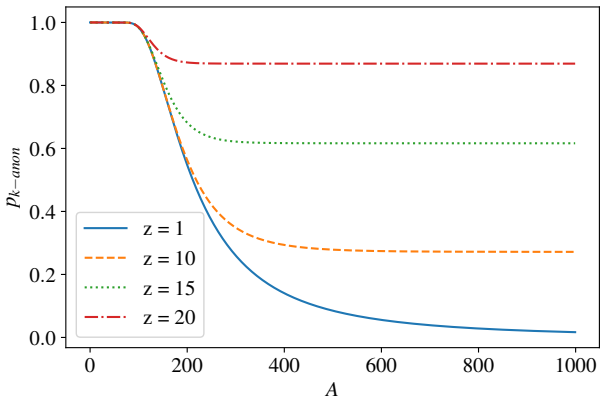
In Figure 3, we study the impact of the size of the catalog of attributes $\mathcal{A}$. In Figure 3a we show in a z-anon dataset how the probability p_{k-anon} of a user being k-anonymized varies with $\mathcal{A}$. To this end, we perform different simulations with increasing numbers of attributes A. We consider a system where only the top A ranked attributes exist. Intuitively, with a large number of attributes, it is harder to find users with the same output attribute set $\overline{y^N}$. However, our assumption of a long tail of infrequent attributes plays with us. indeed, the probability of observing them rapidly goes to to 0 (see Figure 2), and, as such, these attributes rarely appear in the users' released sets. Figure 3a shows this behavior with $k = 2, 3, 4$, while keeping constant values of z and U. With a very small catalog of top-100 or less attributes, users are k-anonymized with reasonable certainty, being very likely to observe multiple users with the same set $\overline{y^N}$. When A increases, we start releasing less-popular attributes. The number of possible attribute combinations thus explodes exponentially[3], and z-anon starts showing its effects. Focusing, for example on the orange dashed curve for $k = 2$, when A exceeds 100, the probability of finding 1 or more identical users to a given one suddenly decreases. However, it settles to approximately 0.9 with $A > 100$, clearly showing the effect of z-anon. The infrequent attributes are not released, and, as such, this limits the explosion of the possible combinations. Further enlarging A does not affect p_{k-anon}, as the attributes in the tail are anyway not published. Increasing the value of k results in lower probability of satisfying k-anon property.

For comparison, in Figure 3b we report the effect of finding at least an identical user to a given one with different values of parameter z of z-anon. Similarly to the other cases, p_{k-anon} starts at 1, when few attributes are present, and the number of their possible combinations is low. When A increases, less frequent attributes start to appear. The possible combinations of attributes explode exponentially. With $z = 1$, i.e., no z-anon in place, the probability of finding identical users rapidly goes to 0. Enabling z-anon, we prevent rare attributes to be released, thus reducing the possible combinations. The higher z, the higher the p_{k-anon}.

[3]The attribute combinations increase as 2^A.



(a) p_{k-anon} changing k ($z = 20$).



(b) p_{k-anon} changing z ($k = 2$).

Fig. 3: The impact of A on p_{k-anon}, considering both different k and z values.

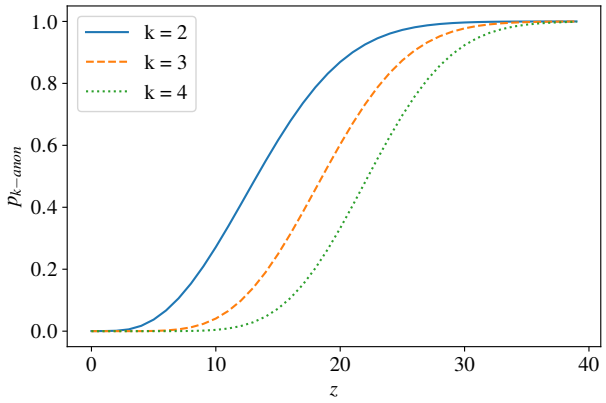


Fig. 4: The impact of z on p_{k-anon} for different k values.

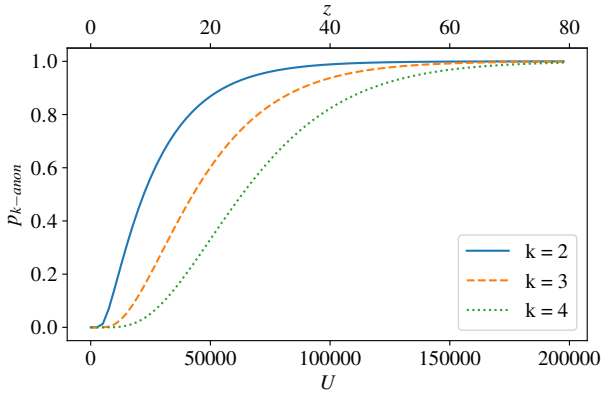


Fig. 5: The impact of U and z on p_{k-anon} for different k values ($z = 20$).

In summary, z-anon allows k-anonymity to be satisfied with a non-zero probability, even with a long tail of attributes.

C. The impact of z

We now evaluate the impact of z on the p_{k-anon}. In Figure 4, we report how different values of z result in different probabilities for a given user to be k-anonymized, i.e., there are at least $k-1$ other users with the same set of released attributes. The other parameters are fixed to the values shown in Table II, and different lines correspond to different values of k. Intuitively, the larger is z, the higher is p_{k-anon}. Focusing on $k = 2$ (blue solid line), p_{k-anon} increases starting from $z = 4$. With $z = 20$, the probability of finding at least a user with an identical set of released attributes is already 0.8. When $k > 35$, p_{k-anon} approaches 1 for the three curves, giving the almost certainty that the whole release is k-anonymized (for $k = 2, 3, 4$). In other words, it is possible to choose a proper z to enforce a desired k and p_{k-anon} on the released data.

D. The impact of U

Next, we study in Figure 5 how the number of users U impacts the privacy of the released data. If we only increase the number of users U, not shown in the Figure, there is a higher chance that some users have even rare attributes released, breaking thus k-anon. This would happen because a large number of users would cause even less-popular attributes to overcome the z threshold, increasing the number of possible combinations, and decreasing p_{k-anon}. Hence, for a fair comparison, z is set proportional with U, and we report it on the upper x-axis of Figure 5. Again, A is fixed to $5\,000$. Focusing on $k = 2$ (blue solid line), we notice how p_{k-anon} grows quickly with U. With $U = 22\,000$ (and $z = 9$), the probability of a user of having another user with identical attributes is already 0.5. p_{k-anon} keeps growing, even if at a lower pace, reaching value very close to 1 with $U = 100\,000$. This result shows that a large number of users leads to better guarantees of k-anon as far as z is set proportionally to U.

E. The impact of N

Finally, Figure 6 shows the impact of the observation time of the attacker (N), defined for simplicity in multiples of Δt. The figure quantifies how increasing N affects p_{k-anon}. In Figure 6a N varies on the x-axis, while different lines represent different k. Intuitively, having a larger observation time makes it more difficult for users to be k-anonymized, since the probability that rare attributes are released increases, and, thus, the number of attribute combinations. When an

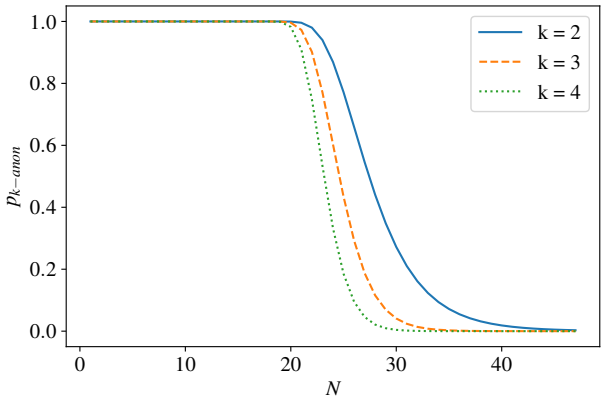
attacker can access enough z-anonymized data, p_{k-anon} drops. Looking at the blue solid line for $k = 2$, after $N = 22$ periods of Δt, the probability of finding identical users starts falling, reaching 0 with $N = 45$. We observe a similar behavior with higher values of k (dashed lines), for which the decrease starts earlier and it is steeper.

Figure 6b shows different insights, observing the impact of the attacker obtaining data in a longer time window. Here, we fix $k = 2$, and we draw different lines for different z, with N up to 400. With $z = 1$, no k-anon can be guaranteed as soon as the attacker observes the data for $N > 3$. z-anon preserves k-anon for longer time (e.g., up to $N = 70$ for $z = 120$). This suggests to use z-anon in combination with other privacy preserving approaches, e.g., user ID rotation or randomization after $N\Delta t$ time. Interestingly, with larger values of z, p_{k-anon} grows again as the observation time increases. This happens because, sooner or later, the most popular attributes will be exposed *and* published by almost every user. Hence, the observations $\overline{y^N(u)}$ will be mostly composed of 1s, and thus most likely be equal to others. For this phenomenon to occur within a reasonable observation time, z must be large enough to just consider most popular attributes, that will take less time to be exposed by almost every user.

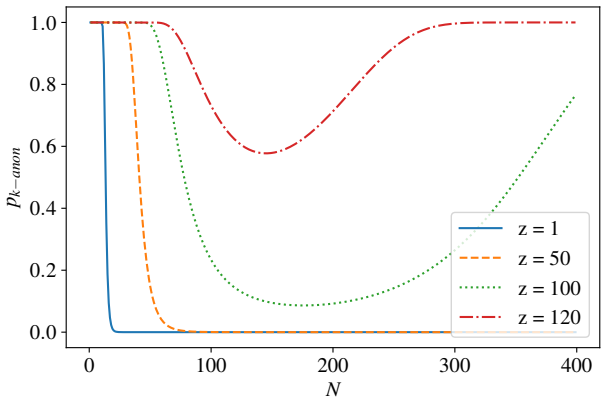
VI. A PRACTICAL USE CASE: THE VISITS TO WEBSITES

In this section, we explore a practical use case for the z-anon: the users' navigation data. To this end, we use the data gathered on a real network to set the parameters of our model. We build on passive measurements collected by Tstat [24], a passive meter that collects rich flow-level records, including hundreds of statistics on the monitored traffic. Essential to our analysis, Tstat builds a log entry for each TCP connection observed on the network, and, for each, it reports, among other statistics, the IP address of the client, a timestamp and the domain name of the server as indicated on the HTTP or TLS headers.[4] We use the entries collected over one day in 2018 in a Point Of Presence of a European ISP aggregating the traffic of approximately 10 000 households. To filter those websites carrying very little information, such as content delivery networks, cloud providers or advertisement, we keep only those websites included in the top-1 Million rank by Alexa[5] and not belonging to the aforementioned categories. For privacy reasons, we encrypted the client identifiers, i.e., the IP addresses, with the Crypto-PAn [25] algorithm, rotating the encryption keys every day.

We use 1 day of collected data to estimate values of the parameters. We assume $\Delta t = 1\,hour$ and $N = 24$. We obtain $A = 27\,482$ and $U = 9\,670$, and we estimate directly the p_a^X for each attribute (a website in this case).[6] Then, we setup our analysis with these obtained parameters, running our probabilistic framework and showing the results we obtain.



(a) p_{k-anon} changing k ($z = 20$).



(b) p_{k-anon} changing z ($k = 2$).

Fig. 6: The impact of observation time N on p_{k-anon}, considering both different k and z values.

In Figure 7, we show the probability p_a^Y of observing the attribute a, for a given user, in the released z-anon data. The solid blue line corresponds to $z = 1$, i.e., no anonymization, thus reporting the popularity of websites in the dataset. The most popular website is *google.com*, which has $p^X_{google.com} = 0.34$, meaning that in 1 hour any of the users will visit this website at least once with this probability. There are some very popular websites, with the top-7 ranked having $p_a^X > 0.1$. In the tail, we find 15 464 websites accessed by only one user on the considered day. When running z-anon with $z > 1$, these uncommon websites are not released, as they are associated with less than z for most of Δt. Focusing on the orange dashed line for $z = 10$, starting from the 200^{th}-ranked website, the probability of observing it in the released data falls rapidly (notice the log scale). Higher values of z (green and red dashed lines) result in earlier and steeper decrease of p_a^Y. We can compare this figure with Figure 2, which shows the same results for the previous case. We first notice that the dashed lines (for $z > 1$) move away from the solid blue line ($z = 1$) in the same range $10^2 - 10^3$. Secondly, we

[4] In case of HTTP transactions, the domain name is extracted from the `Host` HTTP header, while in case of the TLS from the SNI header in the Client Hello message.

[5] https://www.alexa.com/topsites

[6] We opt to extract directly the p_a^X rather than λ_a since these were directly available in the collected data.

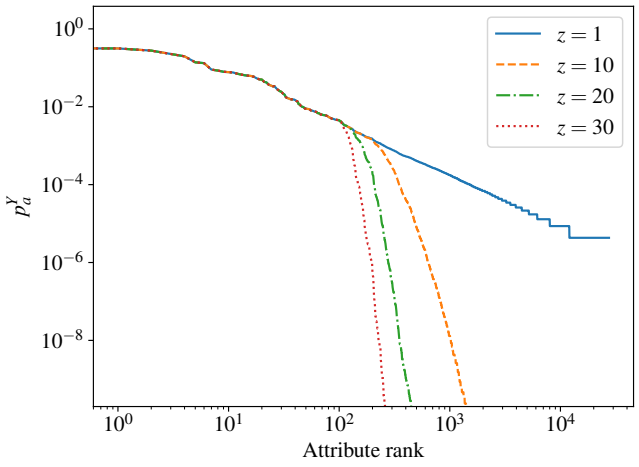


Fig. 7: The probability p_a^Y to publish attribute a in a $\Delta t = 1\,hour$, according to its rank, as estimated from the users' navigation data ($U = 9\,670$, $A = 27\,482$).

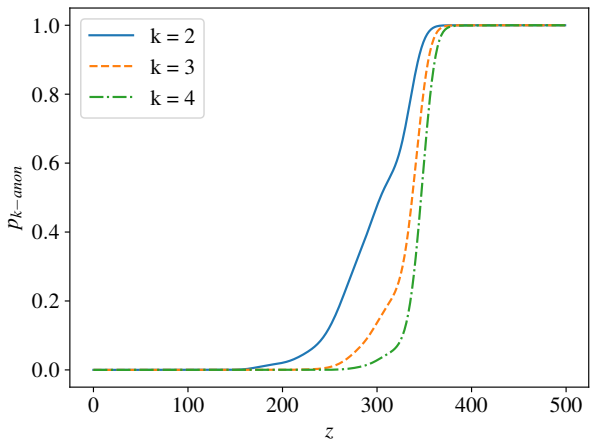


Fig. 8: The relation between z-anon and k-anon in the users' navigation data ($U = 9\,670$, $A = 27\,482$).

notice that the top-ranked attributes have higher p_a^Y than the previous case, with 70 websites having $p_a^Y > 10^{-2}$. This is a peculiarity of the web ecosystem, characterized by a few tens of very popular websites, including popular search engines, news portals and productivity suites, and a long tail of niche websites. In the following, we show that z-anon also works for this scenario, despite the large number of popular websites boosting the number of possible attribute combinations.

We now evaluate the impact of z-anon on the released data in terms of the k-anon property. Running the probabilistic framework described in Section IV, we can derive the probability p_{k-anon} that a given user has at least $k-1$ other users with the same attribute set. We show the results in Figure 8, where we report how p_{k-anon} varies with z, for different values of k. Focusing for example on the blue solid line (for $k = 2$), we notice that z must exceed 200 for p_{k-anon} to move away from 0. p_{k-anon} reaches 1 when z is 350. When considering higher k (dashed lines), larger z are necessary for p_{k-anon} to get close to 1. However it is not necessary a drastic increase of z; for $k = 4$ (green dashed line), $z = 380$ is already enough. Interesting is the comparison of the website visits with the previous case study in Figure 4: here z shall reach 380 to obtain k-anon almost certainly, while $z = 35$ is already enough for the previous case. Two reasons are behind this. Firstly, we have only 9 760 users for the website visits, while $U = 50\,000$ in the previous case, decreasing the probability of finding users with the same set of attributes. Secondly, the probability p_a^X to expose an attribute is quite different for the two cases, with the most popular websites being visited by a large portion of users on a hourly basis. z-anon can provide reasonable guarantees of k-anon even in this case, provided it is properly tuned. However, this guarantees come at the cost of publishing a small number of attributes. This exemplifies the tension between data usefulness and privacy.

VII. Limitations and Future Work

With z-anon, we only prevent users' re-identification if an attacker leverages uncommon attributes, by hiding z-private ones. It is designed uniquely to avoid such kind of re-identification, and, so far, we do not consider other kinds of attacks, e.g., targeting the timing or order at which users' entries appear in the data stream. Moreover, z-anon does not consider combinations of z-anonymized attributes, treating them independently. Still, we provided a probabilistic framework that shows that users can be also k-anonymized with a controllable probability even in case an attacker knows the entire set of released attributes. With this, we provide guidelines to properly tune the system parameters to also guarantee k-anon. This allows the data curator to understand the properties of the released data and manage the trade-off between privacy and data utility.

Future work goes in manifold directions. First, our probabilistic framework can be employed not only to assess how z-anon results into k-anon, but also to *dynamically* tune z to achieve a desired k. The probabilistic framework assumes all users behave the same. Clearly, this is a simple and strong assumption and it can be refined considering classes of users with different rates of activity as well as diverse behaviors. Moreover, in z-anon, we only considered blurring z-private attributes. Alternatively, we could generalize the attributes so that they pass the z-threshold. For example, we could generalize a website to its second level domain or its content category. Moreover, we argue that we can achieve better data utility while avoiding users' re-identification at the same time even if some z-private items are released. This can be obtained by introducing perturbations in the released data, e.g., by inserting noise in the data stream or modifying some of the associations between users and attributes. Such an approach melds concepts from the classical k-anonymity with the ideas of differential privacy, where the addition of noise is the means

to achieve users' privacy.

VIII. Conclusion

In this paper, we presented z-anon, a novel anonymization property suitable for data streams. We designed it to operate with high dimensional data, organized in transactions (atomic information about users) and with the constraint of zero-delay processing. The idea at the base of z-anon is to hide z-private users' attributes, i.e., those associated with less than $z-1$ other users, which could be used by an attacker for re-identification. We show that z-anon can be achieved with an efficient algorithm if using suitable data structures. A data stream undergoing z-anon is immediately anonymized and is available with zero delay to the consumer.

z-anon is weaker than k-anon, as it operates on users' attributes independently without considering their combination. However, we provided a probabilistic framework to map z-anonymity into k-anonymity, using which the data curator can tune the trade-off between privacy and data utility. We show a practical use case, in which we evaluate z-anon using the characteristics of a real dataset of users accessing websites. We show that it is possible to tune the system parameters to obtain k-anon with a controllable probability also in this scenario.

References

[1] L. Vassio, H. Metwalley, and D. Giordano, "The exploitation of web navigation data: Ethical issues and alternative scenarios," in *Blurring the Boundaries Through Digital Innovation*, pp. 119–129, Springer, 2016.

[2] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," *ACM Comput. Surv.*, vol. 42, June 2010.

[3] L. Sweeney, "Guaranteeing anonymity when sharing medical data, the Datafly System," in *Proceedings of the AMIA Annual Fall Symposium*, p. 51, American Medical Informatics Association, 1997.

[4] A. Narayanan and V. Shmatikov in *2008 IEEE Symposium on Security and Privacy (sp 2008)*.

[5] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[6] J. Cao, B. Carminati, E. Ferrari, and K. Tan, "CASTLE: Continuously Anonymizing Data Streams," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, pp. 337–352, 2011.

[7] C. C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality," in *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005* (K. Böhm, C. S. Jensen, L. M. Haas, M. L. Kersten, P. Larson, and B. C. Ooi, eds.), pp. 901–909, ACM, 2005.

[8] T. Favale, M. Trevisan, I. Drago, and M. Mellia, "α-MON: Anonymized Passive Traffic Monitoring," in *to appear in the 32th International Teletraffic Congress*, 2020.

[12] S. Kim, M. K. Sung, and Y. D. Chung, "A framework to preserve the privacy of electronic health data streams," *Journal of Biomedical Informatics*, vol. 50, pp. 95 – 106, 2014. Special Issue on Informatics Methods in Medical Privacy.

[9] M. Khavkin and M. Last, "Preserving Differential Privacy and Utility of Non-stationary Data Streams," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 29–34, 2018.

[10] J. Domingo-Ferrer, J. Soria-Comas, and R. Mulero-Vellido, "Steered Microaggregation as a Unified Primitive to Anonymize Data Sets and Data Streams," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3298–3311, 2019.

[11] M. Chamikara, P. Bertok, D. Liu, S. Camtepe, and I. Khalil, "An efficient and scalable privacy preserving algorithm for big data and data streams," *Computers & Security*, vol. 87, p. 101570, 2019.

[13] S. A. Abdelhameed, S. M. Moussa, and M. E. Khalifa, "Restricted Sensitive Attributes-based Sequential Anonymization (RSA-SA) approach for privacy-preserving data stream publishing," *Knowledge-Based Systems*, vol. 164, pp. 1 – 20, 2019.

[14] J. Wang, C. Deng, and X. Li, "Two Privacy-Preserving Approaches for Publishing Transactional Data Streams," *IEEE Access*, vol. 6, pp. 23648–23658, 2018.

[15] J. Li, B. C. Ooi, and W. Wang, "Anonymizing Streaming Data for Privacy Protection," in *2008 IEEE 24th International Conference on Data Engineering*, pp. 1367–1369, 2008.

[16] B. Zhou, Y. Han, J. Pei, B. Jiang, Y. Tao, and Y. Jia, "Continuous Privacy Preserving Publishing of Data Streams," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, (New York, NY, USA), p. 648–659, ACM, 2009.

[17] J. Zhang, J. Yang, J. Zhang, and Y. Yuan, "KIDS:K-anonymization data stream base on sliding window," in *2010 2nd International Conference on Future Computer and Communication*, vol. 2, pp. 311–316, 2010.

[18] J. Tekli, B. Al Bouna, Y. B. Issa, M. Kamradt, and R. Haraty, "(k, l)-Clustering for Transactional Data Streams Anonymization," in *International Conference on Information Security Practice and Experience*, pp. 544–556, Springer, 2018.

[19] A. B. Sakpere and A. V. D. M. Kayem, "Adaptive buffer resizing for efficient anonymization of streaming data with minimal information loss," in *2015 International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 1–11, 2015.

[20] A. Otgonbayar, Z. Pervez, K. Dahal, and S. Eager, "K-VARP: K-anonymity for varied data streams via partitioning," *Information Sciences*, vol. 467, pp. 238–255, Oct. 2018.

[21] A. Otgonbayar, Z. Pervez, and K. Dahal, "Toward Anonymizing IoT Data Streams via Partitioning," in *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 331–336, 2016.

[22] A. Meyerson and R. Williams, "On the Complexity of Optimal K-Anonymity," in *Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '04, (New York, NY, USA), p. 223–228, Association for Computing Machinery, 2004.

[23] L. A. Adamic, B. A. Huberman, A. Barabási, R. Albert, H. Jeong, and G. Bianconi, "Power-law distribution of the world wide web," *science*, vol. 287, no. 5461, pp. 2115–2115, 2000.

[24] M. Trevisan, A. Finamore, M. Mellia, M. Munafo, and D. Rossi, "Traffic Analysis with Off-the-Shelf Hardware: Challenges and Lessons Learned," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 163–169, 2017.

[25] J. Fan, J. Xu, and M. H. Ammar, "Crypto-PAn: Cryptography-based Prefix-preserving Anonymization," *Computer Networks*, vol. 46, no. 2, 2004.