IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 January 2024, accepted 5 February 2024, date of publication 21 February 2024, date of current version 15 March 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3368852*

RESEARCH ARTICLE

# A Taxonomy of Syntactic Privacy Notions for Continuous Data Publishing

**ADRIÁN TOBAR NICOLAU, JAVIER PARRA-ARNAU, AND JORDI FORNÉ**
Department of Network Engineering, Universitat Politècnica de Catalunya—BarcelonaTech (UPC), 08034 Barcelona, Spain

Corresponding author: Javier Parra-Arnau (javier.parra@upc.edu)

This work was supported in part by Spanish Government under the Project "Enhancing Communication Protocols With Machine Learning While Protecting Sensitive Data (COMPROMISE)" funded by MCIN/AEI/10.13039/501100011033 under Grant PID2020-113795RB-C31, in part by the Project "MOBILYTICS" funded by MCIN/AEI/10.13039/501100011033 under Grant TED2021-129782B-I00, in part by European Union (EU) "NextGenerationEU"/PRTR (Plan de Recuperación, Transformación y Resiliencia), and in part by the Generalitat de Catalunya, under AGAUR Grant 2021 SGR 01413. The work of Javier Parra-Arnau was supported by "Ramón y Cajal" Fellowship funded by Spanish Ministry of Science and Innovation and EU—"NextGenerationEU"/PRTR under Grant RYC2021-034256-I.

**ABSTRACT** Continuous data publishing aims to anonymise the next publication of changing microdata while preserving privacy. The microdata can change between publications via additions, deletions, insertions, and updates. There are numerous proposals for different database types, adversaries, attacks, and notions. However, many anonymization algorithms include notions of privacy and adversarial models that are specific to the context, with their own terminology and notation. Unfortunately, these proposals are difficult to generalize or translate them to other contexts complicating their understanding and comparison. To address these issues, we propose a taxonomy of anonymization technologies, compare existing solutions, and develop a unifying framework that not only harmonizes concepts and terminology but also notation and nomenclature. We analyze the current state of the art and recent advances in the literature. The analysis enables us to understand the significance and appropriateness of the various proposals in achieving privacy.

**INDEX TERMS** Data privacy, dynamic data, syntactic privacy.

## I. INTRODUCTION

Characterized by its massive volume, high velocity and dynamicity, *big data* offers revolutionary advancements to a wide variety of fields such as health and well-being, business competitiveness, marketing, transportation and education. However, the success of big data and the realization of these promises depend, to a large extent, on whether the privacy of the individuals on whom data is collected and analyzed can be guaranteed.

To protect individuals' privacy, current legal frameworks in Europe and most democratic countries limit the collection, processing and sharing of personally identifiable information (PII). Effectively, the data controllers of PII have numerous obligations towards subjects to whom the PII corresponds (seeking their consent, guaranteeing them rights to access, rectification, erasure, etc.).

The surge of big data and the development of data science are harnessing PII-based big data for a great deal of secondary purposes (other than the purpose at collection time). Nonetheless, satisfying the previously mentioned legal obligations towards subjects is extremely challenging in a scenario with a crowd of controllers who exchange and merge data for secondary use.

Anonymization arises as the tool that allows legitimate circumvention of those restrictions and therefore can ease the tensions between the economic and societal good that comes from big-data research on the one hand and the perceived risks to individuals' privacy on the other. However, just removing the direct identifiers of a microdata[1] (names, passport numbers, etc.) is not enough to prevent any disclosure risk, including re-identification. According to a well-known study [1], 63% of the population in the US can be

The associate editor coordinating the review of this manuscript and approving it for publication was Peter Langendoerfer.

[1]A microdata is a data set whose records contain information at the individual level.

uniquely identified based only on the attributes gender, ZIP code, and date of birth.

To effectively anonymize microdata, the common approach is to modify the values of those and similar attributes, called *quasi-identifiers*. Some examples of data-modification techniques are adding Laplacian and Gaussian noise to numerical attributes, reducing the granularity of categorical data, or eliminating the whole attribute [2].

In the field of data anonymization, there are two distinct approaches to protecting microdata: ensuring a privacy property either on the anonymized data or on the mechanism for anonymizing them. In the former case, properties — typically denoted as *notions* or *models*— are referred to as *syntactic*, whereas in the latter they are called *semantic*. While semantic notions such as differential privacy [3] (DP) can provide stronger privacy guarantees (since they make no assumptions on the intruder's side knowledge), syntactic notions are preferred when the aim is to maximize the utility of the dataset studied or model performance and at the same time support a defensible level of privacy [4], [5], [6].

The first and probably best-known syntactic notion was $k$-anonymity [7], which guarantees an upper bound on the risk of reidentification. Although $k$-anonymity and its extensions (e.g., $l$-diversity [8] and $t$-closeness [9]) are well-known for generating anonymized microdata of high utility [6], unfortunately they were not conceived to protect multiple releases or publications of a changing microdata.

Essentially, there are two scenarios for publishing dynamic data with syntactic privacy guarantees (see Sec. II-C for further details): *sequential release publishing* [10] and *continuous data publishing* [11]. Both cases aim to anonymize the next release so that the combination of information from all already anonymized data (which obviously cannot be modified) does not compromise the privacy of data subjects [12]. In *sequential release publishing*, the set of records is kept fixed, whereas the set of attributes changes from one release to another. In *continuous data publishing* it is the other way around: the set of attributes is fixed and, in between releases, records can be inserted, deleted, reinserted, and updated.

Sequential data publishing captures the behaviour of a developing dataset, where the set of attributes is not yet completed or the full publication of the dataset as a static publication is not required. This may be the case, for example, when a set of individuals is studied over time. The participants do not change, but new information is added to the dataset over time. The strength of sequential data publishing lies in its ability to publish only what is relevant, while preserving the ability to publish unreleased information at a later time if necessary. This "publish only what is necessary" gives the flexibility to preserve the utility of the first release with respect to static data release. There are mechanisms to allow sequential data publishing to include new tuples [12], [13], but they come at a cost to utility and are not designed to allow a constant flow of changes to the dataset. On the other hand, continuous data publishing captures the essence of evolving data, i.e., adding users, removing them or updating their information. Examples include healthcare data, such as the people who stay in a hospital, who visit the intensive care unit, and the evolution of their illnesses. The strength of continuous data publishing lies in its ability to republish data while preserving utility, with the ability to provide the latest version of a changing dataset over time. This is important when it is necessary to maintain a complete dataset, for example due to limited data volume or to facilitate data analysis. This work focuses on this latter scenario, which is by far the most studied problem in the literature.

Since the publication of [11], the first model coping with continuous data publishing, numerous proposals have been developed that tackle a variety of aspects, including database types, adversaries, attacks, and notions. However, two important limitations are having an impact on the fragmentation of the literature and on the development and maturity of the research field itself. First and foremost, the vast majority of anonymization algorithms are designed for their own ad hoc privacy notion and threat model, often without an evaluation of previous work. Obviously, this hinders any attempt at generalizing or translating those proposals to other contexts, which makes it difficult for new practitioners to get an overview of the field. And secondly, from the standpoint of privacy experts, the fact that almost each proposal introduces its own nomenclature and notation greatly complicates the understanding and comparison of the claimed privacy guarantees.

### A. CONTRIBUTION AND PLAN OF THIS PAPER

This work aims to conduct a systematization of knowledge that addresses those limitations. Our main contribution arises in response to the need for a unifying framework that enables practitioners and non-experts to compare notions, adversaries, and privacy guarantees (Sec. IV). Our theoretical framework, however, harmonizes not only concepts and terminology but also notation and nomenclature (Sec. III). Under the perspective of this framework, we analyze the state of the art and recent advances in the literature (Sec. IV). Through this analysis, our systematic taxonomy of possible datasets, adversaries, attacks, metrics of utility, notions of privacy, and their guarantees, allows us to comprehend, through numerous examples, the meaningfulness and suitability of different combinations of all those aspects. This way, we address a question of great practical relevance: *given a dataset type and adversary model, which privacy models and guarantees could be achievable?* Last but not least, we identify and discuss research gaps and future directions on anonymization technology (Sec. V). Next, we summarize the major contributions of this work:

- We present a thorough, comprehensive taxonomy of dataset types, adversaries, attacks, and privacy notions in the field of continuous data publishing.
- We develop a theoretical framework that unifies terminology, notation, and nomenclature. To this end, we provide, whenever possible, existing definitions of

all aspects above and novel ones to complete missing information.
- We classify the existing literature to facilitate the choice of an appropriate algorithm based on the dataset, the attacker and the expected level of protection.
- We summarize the main advancements in syntactic anonymization and classify algorithms based on the developed taxonomy.
- We address the practical question of which privacy guarantees can be achieved for a given dataset and adversary.
- We discuss open research directions and critical aspects for the future development of the field.

The methods used to ensure the confidentiality, integrity and security of sensitive data during its processing in a practical scenario are beyond the scope of this paper. We focus on the algorithms used to create data releases. We refer to [2], [14], and [15] as references.

The remainder of this paper is organized as follows: Sec. II recalls general aspects of microdata anonymization and describes the methodology used to survey the literature. Sec. III defines key notation and concepts of the more specific area of continuous data publishing, laying the foundation for the rest of our contributions. Sec. IV presents the proposed taxonomy, develops further the theoretical framework formulated in the previous section, and surveys the state of the art. Sec. V identifies research gaps and future research directions. Finally, conclusions are drawn in Sec. VI.

## II. BACKGROUND

This section aims to provide the reader with the necessary depth to understand the technical contributions of this work. First, Sec. II-A briefly examines the broader field of *statistical disclosure control* (SDC), to which continuous data publishing belongs. Sec. II-B elaborates on the differences between the two main families of privacy notions, which were briefly mentioned in the introduction. Afterwards, Sec. II-C describes the three main data-publication scenarios for dynamic databases under syntactic security. Finally, Sec. II-D presents the methodology we used to perform our work.

### A. STATISTICAL DISCLOSURE CONTROL

The area of continuous data publishing belongs to the broader field of *statistical disclosure control* (SDC) [16]. SDC is the research field that deals with the inherent compromise between protecting the privacy of the individuals in a microdata set and ensuring that those data are still useful for researchers.

In SDC, a microdata set is a database whose records contain information at the level of an individual. In those databases, each row corresponds to an individual and each column to an attribute. According to the nature of attributes, we may classify them into *identifiers*, *quasi-identifiers* or *confidential attributes*. On the one hand, identifiers allow us to unequivocally identify individuals. For example, it would be the case of social security numbers or full names, which would be removed before the publication of the microdata set. On the other hand, key attributes are those attributes that, in combination, may be linked with external information to reidentify the individual to whom the records in the microdata set refers. Last but not least, confidential attributes contain sensitive information about the individual, such as their health condition, political affiliation, religion, or salary.

Before SDC was well-established, a common bad practice by data controllers was to eliminate all identifiers appearing in microdata before its release. However, this is totally insufficient to prevent re-identification or attribute-disclosure attacks, as numerous privacy breaches (e.g., Netflix prize [17], Sweeney's attack in 1997 [18]) have unfortunately shown us.

The bulk of the work done in SDC investigates the case of single data release, i.e., the publication of a single anonymized dataset. Several notions have been proposed for this case. By far, the most popular is *k-anonymity* [7], which is the requirement that each tuple of quasi-identifier values be shared by at least $k$ records in the database. To satisfy this requirement, quasi-identifier values are altered via generalization and suppression, two perturbation mechanisms by which those values are respectively coarsened and eliminated. As a result of applying these methods, all quasi-identifier values within each group are replaced by a common tuple, and thus a record cannot be unambiguously linked to any public database containing identifiers. Consequently, $k$-anonymity is said to protect microdata against *linking attacks*.

### B. SYNTACTIC AND SEMANTIC PRIVACY

With the appearance of new attacks, several alternative privacy notions were developed following an entirely different approach. They can be broadly classified as syntactic and semantic notions (or noise methods) [6].

Syntactic security derives from the imposition of a certain structure on the published, protected dataset. Examples include $k$-anonymity [7], $l$-diversity [8], [19] and $t$-closeness [9]. Semantic notions, on the other hand, enforce some property on the anonymization algorithm. The best-known semantic notion is $\varepsilon$-DP [3], [14], [20], which guarantees the presence or absence of any single record within a dataset will not be noticeable up to an exponential factor of $\varepsilon$. Although DP can take advantage of a composition property to preserve (to a limited extent) the privacy guarantee after repeated data releases, it is at the cost of a significant degradation in data utility [21], [22], an effect that is even exacerbated if one wishes to publish changing microdata multiple times (also known as the *non-interactive* setting of DP). This important limitation is typically addressed with unreasonably large values of $\varepsilon$, which unfortunately vanishes any expectation of privacy for data subjects [5], [23].

To the best of our knowledge, the DP-based proposals [24], [25] that consider the publication of the dataset in a dynamic

scenario, are limited to data streams. Even in this case, DP has difficulties in preventing information leakage [26]. No clear DP-based method exists for the scenario of continuous data publishing. Since the use of DP has not been developed for continuous data publishing, we focus our attention on the existing literature on syntactic privacy with the aim of facilitating in a unified way the work that has already been done.

On the other hand, although it is not explicitly stated, syntactic approaches assume quasi-identifiers are the main target to be attacked and consequently aim to modify the values of those attributes while keeping the sensitive ones unmodified. Furthermore, in contrast to $\epsilon$-DP (which does not make any assumption on the attacker's background knowledge), syntactic models are regarded as rigid in their assumptions of knowledge available to intruders. While this is what allows clear proofs and guarantees in terms of the privacy offered, it is not always clear if such assumptions are reasonable in real practice [4]. For a detailed, complete explanation on the subject, the reader is referred to [2].

### C. DATA-PUBLISHING SCENARIOS

Although the focus of this work is continuous data publishing, it is important to recall which other scenarios concerning dynamic-database publishing have been studied with syntactic privacy protection [12]:

- Multiple data release. Several views of the same underlying dataset are published simultaneously, i.e., publications with all tuples but only a subset of their attributes. This can be useful when several institutions demand different information from the same dataset and the publication of the whole dataset is not necessary.
- Sequential data release. Views of a dataset are sequentially published. This case can be considered when the underlying dataset is being completed as publications are made. In most publications, the number of tuples is assumed to be fixed, with two notable exceptions [12], [13] that consider an increasing number of tuples.
- Continuous data publishing. Publications of a dataset that is updated in between releases. The dataset changes via insertions, deletions, reinsertions, and updates of tuples. The set of attributes is fixed.

### D. METHODOLOGY

Our taxonomy of the literature was conducted based on the methodology of [27] to find relevant papers in continuous data publishing. Our objective was to find the bulk of publications done in continuous data publishing with a focus on syntactic privacy guarantees and algorithms. In one question, *"What is the state of the art in syntactic data privacy for dynamic datasets with continuous data releases?"*.

To this end, a search was carried out on Google Scholar, IEEE xplorer, ACM Library and the Digital Bibliography & Library Project (DBLP) which yield hundreds of publications from 2006 to 2021. Publications were excluded if: the publication did not reach peer-reviewer quality; the dataset studied was not a table (graph, queries, counts, etc); the dataset was not dynamic; the paper was a partial work of a more complete publication; the privacy was not attained using syntactic security (differential privacy, etc); not enough detail/high level explanations/ not significant results; and the publication was not in English. Once the corpus was filtered out, a final set of 37 publications was considered for this work.

## III. NOTATION AND KEY CONCEPTS

Numerous definitions of key concepts in the field have been proposed independently multiple times, often with identical meanings but different names or identical names but different meanings. One of the aims of this work is to harmonize not only nomenclature but also notation. To this end, this section begins by defining key notation and concepts, which will be complemented later on in Sec. IV with additional definitions to complete missing information and unify the work done in the literature. This section, therefore, lays the foundation for the rest of our contributions.

### A. NOTATION

The following is a complete notation set for the problem of continuous data publishing. Henceforth, the phrase "the dataset" will refer to the underlying dataset that is constantly updated and needs to be protected.

- $t$: tuple, i.e., a finite list of quasi identifiers and sensitive attributes.
- $p$: user. A source that generates tuples. We identify a user with its tuple when it does not leave space for error, but a user can have different tuples associated with it.
- $t^*$: anonymized tuple. If the anonymization is achieved through generalization, we shall refer to the anonymized tuple as the *generalized* tuple.
- TS: Timestamp of a tuple. The moment in time when it was released.
- ID: identifier. Each user has a unique identifier.
- QI: quasi identifiers.
- sd: sensitive attribute(s).
- Lifespan $[x, y]$ of $t$: (maximal) interval of time where the tuple is in the underlying dataset. Note that a tuple can have several lifespans. For example, if tuple $t$ is in the dataset in time instants $1, 2, 4, 5$ and 6, it has lifespans $[1, 2]$ and $[4, 6]$.
- L: life of a tuple, i.e., set of lifespans.
- $T_i$: dataset at time $i$.
- $Q_i$: Class. A class is a subset of tuples of a dataset satisfying some relation.
- $T = \{Q_1, \ldots, Q_m\}$: disjoint classes of $T$. Two elements are in the same class if they have the same QI (generalization) or share SD class (anatomization).
- $\mathbf{T} = \{T_1, \ldots, T_n\}$: historic values of the dataset, i.e,. the different values it has taken over time.

- $g_1, \ldots, g_l$: repetition subsets of a class $Q$, i.e., partition of the class into sets, each one containing tuples with a common sensitive attribute.
- $\mathbf{T}T_R$: dataset of all tuples at each time (with possible repetitions).
- $T_i^*$: anonymized release of $T_i$.
- $\mathbf{T}^* = \{T_1^*, \ldots, T_n^*\}$: historic values of the anonymized dataset.
- $\mathbf{T}T_R^*$: dataset of all published tuples (with possible repetitions).
- $SD(T) = \{sd_1, \ldots, sd_l\}$: set of sensitive attributes of tuples in $T$ (without repetition).
- $SD_R(X)$: multiset of sensitive attributes of tuples in X (with repetition).
- $t_1^* \sim t_2^*$: relation of agreement, i.e., they could represent the same user.
- $p \approx t^*$: $t^*$ is a tuple that represents $p$ in the database.
- $t_1^* \approx t_2^*$: $t_1^*$ and $t_2^*$ represents the same user.
- $C(t, T_i)$: set of tuples in $T_i$ that agree with t.
- $Q(t, T^*)$: class that contains $t$ in $T^*$.
- $t[A]$: attribute $A$ of tuple $t$.
- $AK$: attacker knowledge.

### B. KEY CONCEPTS

Next, we define key concepts commonly used in the literature. In general, we shall assume that privacy notions come with some guarantee of privacy that will be of interest to justify their application.

*Definition 1: We call $T^*$ anonymized dataset of some dataset $T$ to a dataset derived from $T$ which satisfies some form of privacy while keeping some of its utility. We call historic values $\mathbf{T} = \{T_1, \ldots, T_n\}$ to the set formed of the different versions of a dataset and the historic anonymizations $\mathbf{T}^* = \{T_1^*, \ldots, T_n^*\}$ to the set of anonymized datasets of the elements in $\mathbf{T}$.*

*Definition 2 (Privacy Breach): the sensitive information of at least one tuple in the dataset has been revealed.*

#### 1) RANDOM WORLD ASSUMPTION

The random world assumption [28] is stated or implicitly assumed in several papers analyzed in this work. For the sake of completeness, we provide a brief explanation of what this assumption implies.

*Definition 3: The (Naive) Random World Assumption states the following: the probability that an answer to a question is the correct solution is equal to $1/m$ where m is the number of reasonable answers. "Reasonable" signifies that a world where such an answer was correct would not cause any contradiction with our knowledge.*

*Definition 4: The (Smart) Random World Assumption states the following: the probability that an answer* `a` *to a question is the correct solution* `cs` *is equal to $P(\texttt{cs} = \texttt{a}|AK)$, where AK is our knowledge. In other words, the likeliness of an answer is conditioned by our knowledge.*

The following example illustrates the two concepts.

*Example 1: Suppose that a coin is tossed. The Naive Random World assumption states that two new worlds are possible, one where the coin shows tails and another where the coin shows heads, and that each one of them is equally possible. Additionally, it states that we do not consider worlds where the coin stays floating in the air since it contradicts our logic and knowledge of the world. The Smart Random World assumption agrees with the naive one if we do not have knowledge about the coin. However, if we know that 7 out of 10 times the coin shows heads, then it suggests that the probability of being in the world where the coin shows heads is 0.7 and not 0.5 as in the previous case.*

In most works, the Random World assumption, regardless of whether naive or smart, is employed to justify the definition of privacy risk being used. In the sequel, we elaborate on the privacy risks typically considered in the literature.

#### 2) RISK

Not entirely surprising, the concept of privacy risk can change depending on the type of dataset and adversary. Next, we provide definitions of risk that will be central to our taxonomy of privacy notions.

*Definition 5 (Generic Risk): Let $\mathbf{T}$ be the historic values of a dynamic dataset, $\mathbf{T}^*$ the historic anonymized releases, and AK the adversary knowledge. Let p be a user that participates in $\mathbf{T}$. We define the risk of p at timestamp ts, $risk(p, ts)$ as the probability to link correctly the user with the sensitive attribute of their associated tuple at timestamp ts knowing AK.*

Notice from the previous definition that the value of $risk(p, ts)$ will depend on the probabilistic framework considered, i.e., how the adversary knowledge affects the probability of linkage.

*Definition 6 (Re-publishing Risk): Let $\mathbf{T}$ be the historic values of a dynamic dataset and $\mathbf{T}^*$ be the historic anonymized releases with $|\mathbf{T}^*| = n$. We define the re-publishing risk of a user p as*

$$risk(p) = \max_{1 \le j \le n} risk(p, j).$$

A definition of privacy risk based on Random World assumption follows.

*Definition 7: Let $\mathbf{T}$ be the historic values of a dynamic dataset and $\mathbf{T}^*$ the historic anonymized releases. We define the "risk 0" of a user p as*

$$risk_0(p) = \frac{n_c}{n_{total}},$$

*where $n_c$ is the number of possible $\mathbf{T}$ that could derive $\mathbf{T}^*$ and assign correctly p to its sensitive attribute; and $n_{total}$ is the total number of possible $\mathbf{T}$ that could have generated $\mathbf{T}^*$.*

The previous definition states that the probability of some linkage is equal to reconstructing $\mathbf{T}$ in such a way that we "casually" link correctly user with sensitive attribute. Observe that the concept of "risk 0" is coherent only if we assume the Naive Random World assumption.

Now we give a definition of risk for fully dynamic datasets. A dataset, or more specifically, a microdata, is fully dynamic if, in between releases, records can be inserted, deleted, reinserted, and updated. A more detailed discussion of fully dynamic datasets can be found in Sec. IV.

*Definition 8: Let* $\mathbf{T}$ *be the historic values of a fully dynamic dataset and* $\mathbf{T}^*$ *be the historic anonymized releases. We define the "risk 1" of a user p at timestamp* $\mathtt{ts}$ *as*

$$risk_1(p, \mathtt{sd}, \mathtt{ts}) = \max_{1 \le j \le \mathtt{ts}} \frac{n_c(p,j)}{n_{total}(p,j)},$$

*where* $n_c(p,j)$ *is the number of* $\mathbf{T}'$ *that could have generated* $\mathbf{T}^*$ *that assign to tuple p at timestamp ts the sensitive attribute sd and* $n_{total}$ *the total number of* $\mathbf{T}'$ *that could have generated* $\mathbf{T}^*$.

Like in the previous case, this definition assumes that all possible tables that can derive $\mathbf{T}^*$ are equally likely, and consequently, the adversary will take one such table randomly, which reduces the risk to a problem of luck.

## IV. A TAXONOMY OF ANONYMIZATION TECHNOLOGY

This section presents the main contribution of this work, a taxonomy of the most important ingredients in continuous data publishing, including datasets, attackers and attacks, metrics of utility, notions of privacy, and their guarantees. In our systematic analysis of the most relevant contributions in this area, we shall also survey the state of the art in anonymization algorithms and introduce novel definitions of those ingredients to unify the work done in the literature. This section is organized as follows: Sec. IV-A focuses on datasets and adversary models; Sec. IV-B on attacks; Sec. IV-C on privacy notions; Sec. IV-D on anonymization algorithms; and Sec. IV-E on utility metrics.

### A. DATASETS AND ADVERSARY MODEL

Relying upon the classification of [29], we begin defining which dataset types are considered in the literature. Afterwards, we define the attackers that have been used more frequently in the field, which, as far as we know, has not been done before.

#### 1) DATASETS

Most progress made in SDC has only considered static datasets, but nowadays the capacity to handle dynamic datasets is crucial for several purposes. To create a well-suited algorithm for a concrete dataset, a clear understanding of what can happen and what cannot in a dataset must be known. Next, we provide a classification of datasets based on their update capacity. This classification was originally presented in [29].

*Static:* the static datasets are the ones considered in the single release model where only one anonymized dataset is published; incremental: only new records are added (see Fig. 2); external dynamic: additions and deletions are considered.[2] (see Fig. 1); fully dynamic: the most general model where all editions can be done, from updating sensitive data to deleting and reinserting a tuple. This model does not allow horizontal updates, i.e., increasing the number of attributes of the tuples (see Fig. 4).

**TABLE 1. Table of possible databases.**

| Dataset | Additions | Deletion | Updates | Reinsertions |
|---|---|---|---|---|
| Static (S) | No | No | No | No |
| Incremental (I) | Yes | No | No | No |
| (External) Dynamic (D) | Yes | Yes | No | No |
| Fully Dynamic (DU) | Yes | Yes | Yes | Yes |

#### 2) ADVERSARIES

Due to the lack of common notation for attackers, it is difficult to know against which knowledge an algorithm can protect. Next, we present a novel classification of attackers.

Table 2 shows the different attackers and their knowledge. *Singular* means of only one user; *bounded*, less than a constant number; and *P. bounded*, upper bounded by a constant probability. Each field (i.e., column) of the table corresponds with: *participants*: users in the dataset; *Temporal K.*: knowledge of the insertions, deletions, and reinsertions. The knowledge can be total if all changes are known, or bounded if it is limited to a certain number of tuples. *S.D.K (Sensitive data Knowledge)*: knowledge of the sensitive values of some users; *S.B.K. (Sensitive Background Knowledge)*: knowledge of the correlation between QI and SDs; C.B.K (Correlation Background Knowledge): knowledge of the correlation between a sensitive attribute and its possible updates.

In this table and throughout this work, we assume any adversary knows:

- The algorithm used to anonymize the dataset.[3]
- The dimensions of the dataset, i.e., the number of columns and the attributes that appear.
- The internal hierarchy of each attribute, i.e., the structure that underlies each attribute.[4]

In some circumstances, adversaries can have knowledge of the updates of sensitive attributes, which may cause some corner cases that generate vulnerabilities. We say that an adversary is aware if it has such knowledge.

Among all cases identified in Table 2, minimal and trivial adversaries represent a powerless type of attacker that endeavors to acquire information with almost no previous knowledge. In most cases, the static privacy notions are good enough to stop their attack.

Incomplete, target, limited, and complete adversaries represent a more organized menace. Since they can track the

[2]It is important to notice that this model is not suitable to update tuples via deletion and the addition of the updated version since this can cause information leakages derived from the model assumption that each addition corresponds to a new user.

[3]We do not consider security through obscurity.

[4]The hierarchy normally indicates specificity. For example, illness is less specific than stomach illness, which is less specific than gastritis. This relation of specificity defines a rooted graph or hierarchy.

**TABLE 2.** Table of adversaries and their knowledge.

| ADVERSARY | PARTICIPANTS | QI | TEMPORAL K. | S.D.K. | S.B.K. | C.B.K. |
|---|---|---|---|---|---|---|
| TRIVIAL | No | No | No | No | No | No |
| MINIMAL | SINGULAR | SINGULAR | No | No | No | No |
| INCOMPLETE | Yes | BOUNDED | No | No | No | No |
| TARGET | Yes | SINGULAR | Yes | No | No | No |
| LIMITED | Yes | Yes | BOUNDED | No | No | No |
| COMPLETE | Yes | Yes | Yes | No | No | No |
| INTERIOR | Yes | Yes | Yes | BOUNDED | No | No |
| PROBABILISTIC | Yes | Yes | Yes | P. BOUNDED | Yes | Yes |

participants in the dataset and their QI, they can associate to each participant a subset of tuples and, accordingly, can make reasonable guesses on the SD of the participants. Static privacy notions cannot stop their possible attacks, nor can stronger ones.

Unlike the previous attackers, an interior adversary has partial knowledge of the sensitive information in the dataset. They can cause a cascade effect by acquiring several chunks of information. In general, a defense against them is unlikely unless some bound is assumed in the amount of sensitive information they have.

Finally, the probabilistic attacker utilizes statistical knowledge to train a classifier and thus deduce with high probability the sensitive information of the participant tuples. Their capacity includes all the possible attacks of complete or weaker attackers. Furthermore, they can mimic an interior attack using probabilistic assumptions, i.e., assuming the values of some attributes with high probability.

#### 3) SUBTLETIES OF DATASETS

It is important to know the particular behaviour of the dataset, not only from its updateability but also from how the tuples may change.

*Definition 9 ([30], [31]): A dataset has arbitrary updates if there is no correlation between previous values and new values of the attributes. In other words, the update of some attribute of a tuple is an event of a random distribution which does not depend on the previous attribute value.*

In general, a fully dynamic dataset with arbitrary updates can be protected with mechanisms from a dynamic dataset since an updated tuple cannot be distinguished from a new one; see [31] for further discussions.

*Definition 10 [32]: An attribute value is permanent if a tuple with that particular value in that attribute cannot be updated to any other value for that attribute. If an attribute is not permanent, it is transient.*

Examples of permanent values are ''deceased'' for medical records or the date of birth.

Another factor to keep in mind is the size of the update. If only one tuple is added to a dataset, it is very likely that a weakness emerges from the similarity of the two releases. In general, we assume that the updates are big enough to avoid such problems unless the method used already prevents such weaknesses. Consider, for example, a complete attacker. If an update of the dataset only consists of the addition of a tuple, he can easily deduce which tuple is new and, from it, deduce their sensitive attribute.

#### 4) SUBTLETIES OF ATTACKERS

Other weaker forms of knowledge from the dataset are possible; since most works in the literature do not tackle this, we give a brief commentary on some of them.

*Definition 11 ([30], [31]): An adversary A is aware if it knows when tuples attributes change (not necessarily knowing the value).*

In general, updates in QI are an extra difficulty for attackers, and updates in SD are source of weaknesses. An aware attacker has the capacity to derive consequences from those updates.

For example, an attacker knows that a tuple is updated in some attribute. In the previous release, the QI class that contained such a tuple had sensitive attributes FLU,CANCER I and the new one has GASTRITIS,CANCER II. The attacker could consider an update from CANCER I to CANCER II.

*Definition 12: If an aware attacker knows which value has updated the attribute (if it was known in the previous release), it is an up-to-date attacker.*

If an attacker is up-to-date, even if quasi identifiers change dramatically, they can still relate tuples to users.

### B. ATTACKS

For the different combinations of datasets and attackers, several attacks are possible. In this subsection, we elaborate on them. Table 3 provides a summary of our analysis. *Attacker* indicates which knowledge is necessary to feasibly perform such an attack, and *achievement* refers to the final objective of the attack.

We start with attacks derived from the insertion of new tuples. These attacks exploit weaknesses that can appear in increasing, external dynamic and fully dynamic databases.

#### 1) INTERSECTION ATTACK

This attack derives from the knowledge of which datasets contain a particular user and the capacity to partially identify that user.

*Example 2: Let Fig. 1(a) and Fig. 1(c) be two instances of an increasing database, and 1(b) and 1(d) be their 2-diversed versions. A curator may believe that 2-diversity provides satisfactory protection. However, if an adversary is aware of the quasi identifiers of user 1 and their participation in*

**TABLE 3. Main attacks in continuous data publishing.**

| Attack | Database | Attacker | Achievement | Reference |
|---|---|---|---|---|
| Minimal | Any | Minimal or stronger | Reveal Sensitive Attributes | [11] |
| Correspondence | Any | Target or stronger | Reveal Sensitive Attributes | [33] |
| Critical Absence | Dynamic/ Fully Dynamic | Target or stronger | Reveal Sensitive Attributes | [34] |
| Critical Addition | Any | Target or stronger | Reveal Sensitive Attributes | [33] |
| Equivalence | Any | Limited or stronger | Relate sets of tuples with common multiset of SD | [35] |
| Interior | Any | Interior or Probabilistic | Reveal Sensitive Attributes | [36] |
| Probabilistic | Any | Probabilistic | Reveal Sensitive Attributes with some probability | [37] |

**TABLE 4. Combinations where *intersection* attacks may happen.**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| MINIMAL | x | | |
| INCOMPLETE | x | | |
| TARGET | x | x | x |
| LIMITED | x | x | x |
| COMPLETE | x | x | x |
| INTERIOR | x | x | x |
| PROBABILISTIC | x | x | x |

| Id | SEX | AGE | S.D. |
|---|---|---|---|
| 1 | MALE | 20 | HIV |
| 2 | FEMALE | 22 | FLU |

(a) Raw Table at time 1

| Id | SEX | AGE | S.D. |
|---|---|---|---|
| 1 | - | [20 – 22] | HIV |
| 2 | - | [20 – 22] | FLU |

(b) 2-diverse table at time 1

| Id | SEX | AGE | S.D. |
|---|---|---|---|
| 1 | MALE | 20 | HIV |
| 2 | FEMALE | 22 | FLU |
| 3 | MALE | 19 | ACNE |
| 4 | FEMALE | 22 | COUGH |

(c) Raw Table at time 2

| Id | SEX | AGE | S.D. |
|---|---|---|---|
| 1 | MALE | [19-20] | HIV |
| 3 | MALE | [19-20] | ACNE |
| 2 | FEMALE | 22 | FLU |
| 4 | FEMALE | 22 | COUGH |

(d) 2-diverse table at time 2

| Id | SEX | AGE | S.D. |
|---|---|---|---|
| 1 | MALE | 20 | HIV |
| 2 | FEMALE | 22 | FLU |
| 3 | MALE | 19 | ACNE |
| 5 | FEMALE | 23 | HIV |

(e) Raw table at time 3

| Id | SEX | AGE | S.D. |
|---|---|---|---|
| 1 | MALE | [19-20] | HIV |
| 3 | MALE | [19-20] | ACNE |
| 2 | FEMALE | [22-23] | FLU |
| 5 | FEMALE | [22-23] | HIV |

(f) 2-diverse table at time 3

**FIGURE 1. Historic values and 2-diverse publications of an external dynamic dataset at different timestamps. Example of intersection attack and critical addition/deletion.**

*the database, they can infer from Table 1(b) that the user has HIV or flu, and from Table 1(d) that they have either HIV or acne. Therefore, it can be concluded that user 1 has HIV.*

*Definition 13: Let A be an attacker, and* $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$, $\mathbf{T}^* = \{T_1^*, T_2^*, \ldots, T_n^*\}$ *be the historic values of a database and their respective anonymizations. An intersection attack proceeds as follows: Let p be a user with known QI.*

- *For each* $T_i^* \in \mathbf{T}^*$, *use the QI's of p to compute* $C_i = C(p, T_i^*)$, *that is, the set of tuples that could belong to him/her.*
- *For each maximal interval* $[x, y]$ *where p has not changed their sensitive attribute, compute the intersection*

$$\bigcap_{i=x \wedge C_i \neq \emptyset}^{y} SD(C_i),$$

*where* $SD(C)$ *denotes the subset of sensitive attributes of C. Then, the sensitive attribute of p is in that intersection.*

### 2) CORRESPONDENCE ATTACK

**TABLE 5. Combinations where correspondence attack may happen.**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| TARGET | x | x | x |
| LIMITED | x | x | x |
| COMPLETE | x | x | x |
| INTERIOR | x | x | x |
| PROBABILISTIC | x | x | x |

From temporal knowledge, three possible attacks emerge: forward-attack, cross-attack and backward-attack. They were formalized in [33] and the first example appears in [11]. The following examples and tables are from [33].

The main weakness is the fact that for each pair of releases, the records can appear or not in each release. See Table 6 for clarification.

**TABLE 6. Types of correspondence attacks.**

| | User Participation | Attacked Release | Background Release |
|---|---|---|---|
| F-attack | $T_1, T_2$ | $T_1^*$ | $T_2^*$ |
| C-attack | $T_1, T_2$ | $T_2^*$ | $T_1^*$ |
| B-attack | $T_2$ | $T_2^*$ | $T_1^*$ |

#### *a: FORWARD-ATTACK*

The attack focuses on reducing the candidate tuples for a particular user. By using the information derived from two consecutive releases, knowing that the target user tuple was in two consecutive releases.

*Example 3: Suppose that an adversary studies a user, Alice, with QI =* $[France, Lawyer]$ *and knows that she was in Table 2(b) and 2(d) (see Fig. 2) which satisfy 5-anonymity. Since there are only two tuples in Table 2(d) with [France, PROF., FLU], one of the records 1,2,3 of Table 2(b) cannot belong to Alice; otherwise, the three of them would be of the form [France, Lawyer, Flu]. With this, we have compromised the 5-anonymity of Alice in Table 2(b).*

*Definition 14 [33]: Let A be an attacker and* $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$, $\mathbf{T}^* = \{T_1^*, T_2^*, \ldots, T_n^*\}$ *be the historic values of a database and their respective anonymized versions. A forward-attack proceeds as follows:*

| Id | BIRTH | JOB | S.D. |
|---|---|---|---|
| 1 | UK | LAWYER | FLU |
| 2 | UK | LAWYER | FLU |
| 3 | UK | LAWYER | FLU |
| 4 | FRANCE | LAWYER | HIV |
| 5 | FRANCE | LAWYER | HIV |

(a) Table $T_3$

| Id | BIRTH | JOB | S.D. |
|---|---|---|---|
| 1 | EUROPE | LAWYER | FLU |
| 2 | EUROPE | LAWYER | FLU |
| 3 | EUROPE | LAWYER | FLU |
| 4 | EUROPE | LAWYER | HIV |
| 5 | EUROPE | LAWYER | HIV |

(b) 5-anonymized $T_3$

| Id | BIRTH | JOB | S.D. |
|---|---|---|---|
| 1 | UK | LAWYER | FLU |
| 2 | UK | LAWYER | FLU |
| 3 | UK | LAWYER | FLU |
| 4 | FRANCE | LAWYER | HIV |
| 5 | FRANCE | LAWYER | HIV |
| 6 | FRANCE | LAWYER | HIV |
| 7 | FRANCE | DOCTOR | FLU |
| 8 | FRANCE | DOCTOR | FLU |
| 9 | UK | DOCTOR | HIV |
| 10 | UK | LAWYER | HIV |

(c) Table $T_4$

| Id | BIRTH | JOB | S.D. |
|---|---|---|---|
| 1 | UK | PROF. | FLU |
| 2 | UK | PROF. | FLU |
| 3 | UK | PROF. | FLU |
| 9 | UK | PROF. | HIV |
| 10 | UK | PROF. | HIV |
| 4 | FRANCE | PROF. | HIV |
| 5 | FRANCE | PROF. | HIV |
| 6 | FRANCE | PROF. | HIV |
| 7 | FRANCE | PROF. | FLU |
| 8 | FRANCE | PROF. | FLU |

(d) 5-anonymized $T_4$

**FIGURE 2. Historic values and 5-anonymous publications of an incremental dataset. Example of Forward, Cross and Backward-attack.**

*Let p be a user with known QI.*

- *For each lifespan $[x, y]$ of p where it does not change their sensitive attribute, consider each pair $T_i^*, T_{i+1}^*$ with $i \in [x, y-1]$ and:*
  - *Compute the set $C_i = C(p, T_i^*)$*
  - *Compute the set of tuples in $T_{i+1}^*$ that agree with the ones in $C_i$ and with the QI of p. The tuple of p is among them.*

#### b: CROSS ATTACK

The attack focuses on reducing the candidate tuples for a particular user using the information derived from two consecutive releases, knowing that the target user tuple was in both releases.

*Example 4: Let Alice be a user with QI [France,Lawyer] that appears in Tables 2(b) and 2(d) (see Fig. 2) which satisfy 5-anonymity. From records 4,5,6 in Table 2(d) it can be deduced that at least one was not in Table 2(b) since only two cases of HIV were published. From here, it is clear that one of those registers does not belong to Alice, thereby breaking 5-anonymity.*

*Definition 15: Let A be an attacker, and $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$, $\mathbf{T}^* = \{T_1^*, T_2^*, \ldots, T_n^*\}$ be the historic values of a database and their respective anonymizations. A cross-attack proceeds as follows:*

*Let p be a user with known QI.*

- *For each lifespan $[x, y]$ of p where p does not change their sensitive attribute, consider each pair $T_i^*, T_{i+1}^*$ with $i \in [x, y-1]$ and:*
  - *Compute the set $C_{i+1} = C(p, T_{i+1}^*)$*
  - *Compute the set of tuples in $T_i^*$ that agree with the ones in $C_i$ and with the QI of p. The tuple of p is among them.*

#### c: BACKWARD ATTACK

The attack focuses on reducing the candidate tuples for a particular user using the information derived from two consecutive releases, knowing that the target user tuple was in the latter but not in the former release.

*Example 5: Let Alice be a user with QI [UK,Lawyer] which appears in Table 2(d) but not in 2(b) (see Fig. 2) which satisfy 5-anonymity. From records 1,2,3 of Table 2(d) it is clear that at least one cannot be Alice. Otherwise, at least one of the records 1,2,3 in Table 2(b) would not have an instance in Table 2(d).*

*Definition 16: Let A be an attacker, and $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$, $\mathbf{T}^* = \{T_1^*, T_2^*, \ldots, T_n^*\}$ be the historic values of a database and their respective anonymizations. A backward-attack proceeds as follows:*

*Let p be a user with known QI.*

- *For each lifespan $[x, y]$ of p where p does not change their sensitive attribute, consider the pair $T_{x-1}^*, T_x^*$:*
  - *Compute the set $C_x = C(p, T_x^*)$*
  - *Compute the set $X$ of tuples in $T_x^*$ that represent some user of $T_{x-1}^*$. The tuple of p is in $C_x \setminus X$.*

#### 3) CRITICAL ABSENCE/ADDITION

Critical absence is a phenomenon originally presented in [34] that occurs when an uncommon tuple is removed from the dataset. If the anonymization method does not take into account the rarity of such tuples, their absence can be notorious to attackers with temporal knowledge. Similarly, critical addition occurs when a new tuple with a new sensitive attribute value (i.e., not observed in the database previously) is added; this phenomenon is a particular case of a backward attack.

**TABLE 7. Combinations where critical *absence* may happen.**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| TARGET | | x | x |
| LIMITED | | x | x |
| COMPLETE | | x | x |
| INTERIOR | | x | x |
| PROBABILISTIC | | x | x |

**TABLE 8. Combinations where critical *addition* may happen.**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| TARGET | x | x | x |
| LIMITED | x | x | x |
| COMPLETE | x | x | x |
| INTERIOR | x | x | x |
| PROBABILISTIC | x | x | x |

*Example 6: Let Tables 1(a),1(c),1(e) (see Fig. 1) be the historic values of a dataset, and Tables 1(b),1(d),1(f) (see Fig. 1) be their anonymizations. If an adversary has temporal knowledge of user 4, they can learn that that user participated in Table 1(d) but not in 1(f). Since the value COUGH does not appear in Table 1(f) the adversary can link it to user 3. Similarly, user 5 can be linked to HIV (critical addition).*

*Definition 17: Let A be an attacker, and $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$, $\mathbf{T}^* = \{T_1^*, T_2^*, \ldots, T_n^*\}$ be the historic values of a database and their respective anonymizations. A critical absence/addition attack proceeds as follows:*

*For each pair $T_i, T_{i+1}$ where deletions/additions are made:*

- *Compare the quantities of each sensitive attribute in $T_i^*$ and $T_{i+1}^*$.*
  - *Each sensitive attribute that has been reduced in quantity must belong to some user deleted from the dataset. (critical absence) (if no sensitive updates)*
  - *Each sensitive attribute that has increased in quantity must belong to a new or reinserted user. (critical addition) (if no sensitive updates)*

In combination with the intersection attack, it is called the join attack [38].

#### 4) EQUIVALENCE ATTACK

The authors of [35] present value-equivalence attacks. Instead of searching for data leaks, the value-equivalence attack tries to relate the sensitive attributes of users. In other words, it tries to find two disjoint subsets of tuples with common sensitive attributes.

**TABLE 9. Combinations where equivalence attacks may happen.**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| LIMITED | x | x | x |
| COMPLETE | x | x | x |
| INTERIOR | x | x | x |
| PROBABILISTIC | x | x | x |

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | 20 | MALE | FLU |
| 2 | 21 | FEMALE | FLU |
| 3 | 18 | MALE | ACNE |
| 4 | 22 | FEMALE | HIV |

(a) Raw Table

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | [18-20] | MALE | FLU |
| 3 | [18-20] | MALE | ACNE |
| 2 | [21-22] | FEMALE | FLU |
| 4 | [21-22] | FEMALE | HIV |

(b) 2-diversed Table

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | 20 | MALE | FLU |
| 3 | 18 | MALE | ACNE |
| 4 | 22 | FEMALE | HIV |
| 5 | 19 | MALE | FLU |

(c) Updated Table

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | [20-22] | - | FLU |
| 4 | [20-22] | - | HIV |
| 3 | [18-19] | MALE | ACNE |
| 5 | [18-19] | MALE | FLU |

(d) 2-diversed Table

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | 20 | MALE | FLU |
| 4 | 22 | FEMALE | HIV |
| 5 | 19 | MALE | FLU |
| 6 | 16 | MALE | ACNE |

(e) Updated Table

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | [20-22] | - | FLU |
| 4 | [20-22] | - | HIV |
| 5 | [16-19] | MALE | FLU |
| 6 | [16-19] | MALE | ACNE |

(f) 2-diversed Table

**FIGURE 3. Historic values and 2-diverse publications of an external dynamic dataset. Example of equivalence attack.**

*Example 7: Let Tables 3(a) and 3(c) be the historic values and Tables 3(b) and 3(d) be their anonymizations (see Fig. 3). Observe that users 1 and 3 are in the same class in the first release (Table 3(b). Since users 3 and 5 are in the same class in the second release (Table 3(d), from the fact that in both releases the sensitive attributes were FLU and ACNE, we deduce that users 1 and 5 have the same sensitive attribute.*

*Definition 18 [35]: Let $\mathbf{T}T_R$ be the list of historic tuples, $\mathbf{T}T_R^*$ the list of anonymized tuples historic values of a database, and $\mathbf{T} = \{T_1^*, \ldots, T_n^*\}$ the anonymized releases. We say that there is an e-value equivalence attack if an adversary can find $P_1$ and $P_2$ two sets of tuples such that:*

- $P_1 \cap P_2 = \emptyset$ *and* $P_1, P_2 \subset \mathbf{T}T_R^*$.
- $\forall t_1, t_2 \in P_1 \cup P_2$, $t_1 \not\approx t_2$.
- $|P_1| = |P_2| = e$
- $SD_R(P_1) = SD_R(P_2)$.

*Example 8: From Tables 3(b) and 3(f) (see Fig. 3) with quasi identifiers and temporal knowledge, it can be seen that $P_1 = \{1, 3\}$ and $P_2 = \{5, 6\}$ have the same sensitive attributes, namely $\{FLU, ACNE\}$. This implies that there exists a 2-value equivalence attack.*

*Definition 19 [35]: Let $\mathbf{T}T_R^*$ be the list of all published tuples, and let $\{P_1, \ldots, P_N\}$ be a partition of $\mathbf{T}T_R^*$, $U(\mathbf{T}^*) = \{p_1, \ldots, p_n\}$ be the users in $\mathbf{T}^*$ and $SD(\mathbf{T}T_R^*) = \{sd_1, \ldots sd_l\}$ the sensitive attributes in $\mathbf{T}T_R^*$. We define:*

- $S_k = [s_1, \ldots, s_n]$ *as the vector with $s_i = 1$ if $p_i$ is in $P_k$ and zero otherwise.*
- *The person matrix* $S = \begin{bmatrix} S_1 \\ \vdots \\ S_n \end{bmatrix}$.
- $U_k = [u_1, \ldots, u_m]$ *as the vector with value $u_i$ the number of occurrences of $sd_i$ in $P_k$.*
- *The value matrix* $U = \begin{bmatrix} U_1 \\ \vdots \\ U_n \end{bmatrix}$.

*Definition 20 [35]: Let $\mathbf{T}T_R^*$ be a database of historic releases, $S$ be its person matrix, and $U$ its value matrix. We say that $W = [w_1, \ldots, w_n]$ is an instance of an e-equivalence attack if*

- $W \neq \vec{0}$
- $e = \frac{||W \cdot S||_1}{2} \neq 0$
- $W \cdot U = \vec{0}$
- $W \cdot S \in \mathbb{Z}^n$

*Furthermore, if $\frac{||W \cdot S||_1}{2}$ is minimal among all instances of an equivalence attack, we say that it is a minimum equivalence attack.*

*Example 9: Consider Tables 3(b) and 3(d) (see Fig. 3) it has person matrix*

$$S = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix},$$

*with partitions $P_1 = \{1, 3\}, P_2 = \{2, 4\}$ in Table 3(b) and $P_3 = \{1, 4\}, P_4 = \{3, 5\}$ in Table 3(b) which are the classes of each release. Each $s_{i,k}$ has value 1 if tuple/user $k$ appears in partition $P_i$ and 0 otherwise. The user matrix has value*

$$U = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix},$$

*where $u_i, 1$ has the number of instances of FLU in $P_i$, $u_{i,2}$ the number of instances of ACNE in $P_i$ and $u_{i,3}$ the*

*number of instances of HIV in $P_i$. Examples of instances of an e-equivalence attack are $W_1 = (1, 0, 0, -1)$ and $W_2 = (0, 1, -1, 0)$. Checking each condition, we have*:

- $W_1, W_2 \neq 0$.
- $e = \frac{||W_1 \cdot S||_1}{2} = \frac{||W_2 \cdot S||_1}{2} = 1 \neq 0$.
- $W_1 \cdot S = (1, 0, 0, 0, -1) \in \mathbb{Z}^5$, $W_2 \cdot S = (-1, 1, 0, 0, 0) \in \mathbb{Z}^5$.

*The pair of multisets related to the equivalence attack can be extracted from the value $W \cdot S = (ws_1, \ldots, ws_n)$. One multiset is the i's tuples with $ws_i = 1$ and the other is the i's tuples with $ws_i = -1$. In the cases of our examples, we deduce that tuples 1 and 5 have common sensitive attribute and that tuples* 1 *and* 2 *also have common sensitive attribute.*

#### 5) INTERIOR ATTACK

Daniele et. al. [36] present interior attacks. An attacker knows the sensitive attributes of some users and wants to extract more using them. This attack can be done using the equivalence attack in combination with known sensitive attributes. It can also be done probabilistically, i.e., assuming with high probability the sensitive attributes of some tuples.

**TABLE 10. Combinations where interior attacks may happen (p probabilistic).**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| INTERIOR | x | x | x |
| PROBABILISTIC | p | p | p |

In general, knowledge of sensitive values can be used to attack a dataset by studying its historical correlations.

*Definition 21 ( [36] (Historical correlation)): Let $T_i^*$, $T_j^* \in \mathbf{T}^*$ with $i \neq j$ be two releases, $Q_1 \in T_i^*, Q_2 \in T_j^*$ be two classes with $SD_R(Q_1) = SD_R(Q_2)$ and $U_1, U_2 \subset U(\mathbf{T}T_R)$ are two sets of users. We say that $U_1$ and $U_2$ are in historical correlation if*

- $U_1 \subset U(Q_1)$.
- $U_2 \subset U(Q_2)$.
- $U(Q_1) \setminus U_1 = U(Q_2) \setminus U_2$.
- *The attributes of the users $U(Q_1 \cup Q_2)$ have not changed in the interval $[i, j]$.*

*Theorem 1 [36]: If two sets of users $U_1, U_2$ are in historical correlation, then $SD_R(U_1) = SD_R(U_2)$.*

*Definition 22: An interior attack exploits the knowledge of the sensitive attributes of some compromised tuples via equivalence attacks, a study of historic correlation or some other method to obtain sensitive information from users.*

#### 6) PROBABILISTIC ATTACKS

**TABLE 11. Combinations where probabilistic background knowledge attacks may happen (p probabilistic).**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| PROBABILISTIC | p | p | p |

The Probabilistic Background Knowledge attack (PBK) consists of using probabilistic knowledge of the data to be able to guarantee that a tuple has a certain sensitive attribute with some probability p. If an adversary can do that, we say that it has done a p-value association attack. This kind of attacks are strongly based on the Smart Random World Assumption since they use their knowledge to decide which raw dataset is more likely. The main work on this subject for continuous data release is done in [37].

### C. PRIVACY NOTIONS

Now that we have seen the principal attacks that can be presented in the continuous data release, the next step is to show in which cases a method of protection is known. Although there are several methods to protect data privacy, in most of them there are not clear results that show the guarantee of privacy that they provide [42], [43], [44]. We present here the ones with a clear guarantee. Table 12 summarizes the different notions.

Tables 14,15 or completely solved in the literature. If some case is not always achievable, it is because some property of the dataset is assumed. For the definitions of risk, see Sec. III-B2.

#### 1) BCF-ANONYMITY

A method to protect against Backward, Cross and Forward attacks [33] is presented. This method is limited to incremental datasets. All of the attacks mentioned try to discard tuples among the candidate set of a particular user. Such tuples are called cracked tuples.

*Definition 23 (Repetition Subset): We call repetition subset to the subsets $g_1, \ldots, g_n$ of the class $Q_1 \in T_1^*$ satisfying*

- $g_i \cap g_j = \emptyset$ *for all* $i \neq j$.
- $|SD(g_i)| = 1$.
- $\bigcup_{i=1}^{n} g_i = Q_1$.

First, we define cracked tuples for Forward attacks.

*Definition 24 [33]: (FA, crack size) Let $p$ be a user that appears in $T_1^*, T_2^*$. Let $Q_1 \in T_1^*$ and $Q_2 \in T_2^*$ be the classes where the tuple of $p$ appears. Let $g_1, \ldots, g_n$ be repetition subsets of $Q_1$ where $g_i$ is the set of tuples with the i-th sensitive attribute in $Q_1$. Then, $g_i$ has a crack size of $c$ with respect to $p$ if $c$ is the biggest integer such that at least $c$ users in $U(g_i)$ do not have the same quasi identifiers as $p$.*

In other words, the crack size of a class with respect to $p$ is the number of tuples that we can guarantee that do not belong to $p$ because they must belong to another tuple with different quasi identifiers.

*Example 10: Consider the Fig. 2, a user A has quasi identifiers $[FRANCE, LAWYER]$ and matches the classes $Q_1 = [EUROPE, LAWYER] = \{1, 2, 3, 4, 5\}$ with $g_1 = \{1, 2, 3\}$ and $g_2 = \{4, 5\}$, and $Q_2 = [FRANCE, PROF.] = \{4, 5, 6, 7, 8\}$ with $g_1' = \{4, 5, 6\}$ and $g_2' = \{7, 8\}$. The set $g_1$ has crack size (at least) 1 with respect to A, because if all $U(g_1)$ had the same quasi identifiers as A, then the second release would have had at least three tuples of the form $[FRANCE, PROF., FLU]$.*

**TABLE 12. Main privacy notions in continuous data publishing.**

| Name | Database | Adversary | Guarantee | Reference |
|---|---|---|---|---|
| BCF-anonymity | Increasing | BCF attacks | Protection against BCF attacks | [33] |
| m-invariance | Dynamic | See table 16 | bound on $risk_0$ | [34] |
| $\tau$-safety | Fully Dynamic (arbitrary updates) | See table 16 | bound on $risk_0$ | [30], [31], [39] |
| m-Distinct | Fully Dynamic | Complete | bound on $risk$ | [29], [40] |
| Pvr-safety | Dynamic | ct-pvr and hc-pvr attacks | bound on $p_{pb}$ | [36] |
| (Refined) HD-composition | Fully Dynamic | Limited | bound on $risk_1$ | [32] |
| "Bayesian Based Protection" | Fully Dynamic | Probabilistic | - | [37] |
| "Microaggregation Method" | Fully Dynamic | - | - | [41] |

**TABLE 13. Cases where some notion of risk is bounded for incremental datasets. $X'$ : not always achievable, $X_a$ : includes aware adversaries.**

| INCREMENTAL | $risk$ | $risk_0$ | $risk_1$ | other |
|---|---|---|---|---|
| MINIMAL | $X'$ | $X_a$ | $X$ | |
| INCOMPLETE | $X'$ | $X_a$ | $X$ | |
| TARGET | $X'$ | $X_a$ | $X$ | |
| LIMITED | $X'$ | $X_a$ | $X$ | |
| COMPLETE | $X'$ | $X_a$ | | |
| INTERIOR | | | | |
| PROBABILISTIC | | $X_a$ | | |
| OTHER | | | | $BCF$ ,$p_{pb}$-secure |

**TABLE 14. Cases where some notion of risk is bounded for dynamic datasets. $X'$ : not always achievable, $X_a$ : includes aware adversaries.**

| DYNAMIC | $risk$ | $risk_0$ | $risk_1$ | other |
|---|---|---|---|---|
| MINIMAL | $X'$ | $X_a$ | $X$ | |
| INCOMPLETE | $X'$ | $X_a$ | $X$ | |
| TARGET | $X'$ | $X_a$ | $X$ | |
| LIMITED | $X'$ | $X_a$ | $X$ | |
| COMPLETE | $X'$ | $X_a$ | | |
| INTERIOR | | | | |
| PROBABILISTIC | | $X_a$ | | |
| OTHER | | | | $p_{pb}$-secure |

**TABLE 15. Cases where some notion of risk is bounded for fully dynamic datasets. $X'$ : not always achievable, $X_{a,u}$ : includes aware adversaries and assumes arbitrary updates.**

| FULLY DYNAMIC | $risk$ | $risk_0$ | $risk_1$ | other |
|---|---|---|---|---|
| MINIMAL | $X'$ | $X_{a,u}$ | $X$ | |
| INCOMPLETE | $X'$ | $X_{a,u}$ | $X$ | |
| TARGET | $X'$ | $X_{a,u}$ | $X$ | |
| LIMITED | $X'$ | $X_{a,u}$ | $X$ | |
| COMPLETE | $X'$ | $X_{a,u}$ | | |
| INTERIOR | | | | |
| PROBABILISTIC | | $X_{a,u}$ | | |
| OTHER | | | | |

*Definition 25 ( [33] (F-anonymity)): Let $F(p, Q_1, Q_2)$ be the sum of the crack sizes of each $g_i$ of $Q_1$ with respect to p. Let $F(Q_1, Q_2)$ be the maximum among all $F(p, Q_1, Q_2)$, with p a user that appears in $Q_1$ and $Q_2$. Let $F(Q_1)$ be the maximum among all $F(Q_1, Q_2)$, with $Q_2$ a class in $T_2$. We define $FA(T_1^*, T_2^*)$ (or FA) the F-anonymity of $(T_1^*, T_2^*)$ as the minimum value of $(|Q_1| - F(Q_1))$ among all $Q_1$ in $T_1^*$.*

Now we state the theorem that computes the crack size.

*Definition 26: We define $CG(Q_1, Q_2)$ as the set of pairs $(g_i, g'_j) \subseteq Q_1 \times Q_2$ of repetitions subsets satisfying $SD(g_i) = SD(g'_j)$.*

*Theorem 2 [33]: Let p be a user that participates in $Q_1 \in T_1^*$ and $Q_2 \in T_2^*$. Let $g_1 \in Q_1$ and $g_2 \in Q_2$ be repetition subsets such that $SD(g_1) = SD(g_2)$, then:*

- *The crack size of $g_1$ with respect to p is equal to $|g_1| - \min(|g_1|, |g_2|)$.*
- *$F(Q_1, Q_2) = \sum_{(g_i, g'_j) \in CG(Q_1, Q_2)} |g_i| - \min(|g_i|, |g'_j|)$*

*Corollary 1 [33]: The (FA) crack size of a repetition subset is not dependent on the user.*

Now we present the same work for Cross attacks.

*Definition 27 [33]: (CA,crack size) Let p be a user that participates in $Q_1$ and $Q_2$. A repetition group $g_2 \in Q_2$ has crack size c with respect to p if c is maximal such that at least c users in $U(g_2)$ do not coincide in quasi identifiers with p or have different timestamps with respect to the releases $T_1^*, T_2^*$.*

*Definition 28 [33]: (C-anonymity) Let $C(p, Q_1, Q_2)$ be the sum of the crack sizes of each $g_i$ of $Q_1$ with respect to p. Let $C(Q_1, Q_2)$ be the maximum among all $C(p, Q_1, Q_2)$, with p a user that appears in $Q_1$ and $Q_2$. Let $C(Q_2)$ denote the maximum $C(Q_1, Q_2)$ among all $Q_2$ classes in $T_2^*$. The C-anonymity of $(T_1^*, T_2^*)$ denoted by $CA(T_1^*, T_2^*)$ or CA is the minimum $(|Q_2| - C(Q_2))$ among all $Q_2$ classes of $T_2^*$.*

*Theorem 3 [33]: Let p be a user that participates in $Q_1 \in T_1^*$ and $Q_2 \in T_2^*$. Let $g_1 \in Q_1$ and $g_2 \in Q_2$ be repetition subsets such that $SD(g_1) = SD(g_2)$, then:*

- *The crack size of $g_2$ with respect to p is equal to $|g_2| - \min(|g_1|, |g_2|)$.*
- *$F(Q_1, Q_2) = \sum_{(g_i, g'_j) \in CG(Q_1, Q_2)} |g'_j| - \min(|g_i|, |g'_j|)$*

*Corollary 2 [33]: The (CA) crack size of a repetition subset is not dependent on the user.*

We state a theorem that relates CA and FA.

*Theorem 4 [33]: The values of FA and CA coincide, i.e., $FA(T_1^*, T_2^*) = CA(T_1^*, T_2^*)$.*

We now state now the definitions for Backward attacks.

*Definition 29 [33]: Let p a user that participates in $Q_2$. A repetition subset $g_2$ of $Q_2$ has crack size c with respect to p if c is maximal such that at least c records in $U(g_2)$ have participated in $T_1^*$.*

*Definition 30 [33]: Let $B(p, Q_2)$ be the sum of the crack sizes of all repetition subsets in $Q_2$ with respect to P. Let $B(Q_2)$ be the maximum $B(p, Q_2)$ among all p that appear in $Q_2$. The B-anonymity of $(T_1^*, T_2^*)$ denoted by $BA(T_1^*, T_2^*)$ or BA is the minimum $(|Q_2| - B(Q_2))$ among all $Q_2 \in T_2^*$.*

*Theorem 5 [33]: Let p be a user that appears in $Q_2$ but not in $T_1^*$. Let $g_2$ be a repetition subset of $Q_2$. Let $G_1$ be the*

*set of tuples in $T_1^*$ that could belong to $g_2$, and $G_2$ be the set of tuples $t^*$ in $T_2^*$ that satisfy $t^* \sim t'^*$ for some tuple $t'^*$ in $G_1$.*

- *If $|G_2| < |g_2|$, $g_2$ has crack size 0 with respect to $p$.*
- *If $|G_2| \geq |g_2|$, $g_2$ has crack size $c = \max(0, |G_1| - (|G_2| - |g_2|))$*
- *$B(Q_2) = \sum_{g_i \in Q_2} c_{g_i}$ where $c_{g_i}$ is the crack size of $g_i$ with respect to any $p$.*

Finally, we state BCF-anonymity.

*Definition 31 [33]: Let $\mathbf{T}^*$ be the historic values of an anonymized increasing dataset. Then it is BCF-anonymous with parameter $k$ if BA, CA and FA are bigger or equal to $k$.*

This condition guarantees that BCF attacks cannot reduce the candidate tuples of any user to less than $k$.

**TABLE 16. Cases where m-invariance (○) or $\tau$-safety (×) prevents attacks.**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| MINIMAL | ⊗ | ⊗ | × |
| INCOMPLETE | ⊗ | ⊗ | × |
| TARGET | ⊗ | ⊗ | × |
| LIMITED | ⊗ | ⊗ | × |
| COMPLETE | ⊗ | ⊗ | × |
| INTERIOR | ⊗ | ⊗ | × |
| PROBABILISTIC | ⊗ | ⊗ | × |

### 2) m-invariance AND $\tau$-SAFETY

The notion of m-invariance [34] was proposed to prevent intersection attacks for incremental and dynamic datasets; $\tau$-safety [30], [31] is an improvement that extends m-invariance to fully dynamic datasets. Several implementations and variations for both notions exist [45], [46].

*Definition 32: Let $Q_i$ be a class in $T_i^*$ for any $i \in [1, n]$. The signature of $Q_i$ is the set of sensitive attributes in $Q_i$ denoted by $SD(Q_i)$.*

*Definition 33 [34]: (m-invariance) An anonymized table $T^* = \{Q_1, \ldots, Q_k\}$ is m-unique if each class in $T^*$ contains at least m tuples, and all tuples in the class have different sensitive attributes.*

*$\mathbf{T}^* = \{T_1^*, \ldots, T_n^*\}$ is m-invariant if the following conditions hold:*

- *$T_j^*$ is m-unique for all $j \in [1, n]$.*
- *For any tuple $t \in \mathbf{T}T_R^*$ with lifespan $[x, y]$, it is satisfied $SD(Q(t, T_i^*)) = SD(Q(t, T_j^*))$ for all $i, j \in [x, y]$.*

Clearly, m-uniqueness implies m-diversity. With that, it follows that all releases satisfying m-invariance have the privacy guarantees that m-diversity yields for any static attack. Now we state definitions that will be used to prove the privacy guarantee of m-invariance.

*Definition 34 [34]: Consider $\mathbf{T}$ and $\mathbf{T}^*$ of a dynamic dataset. Let $U^*$ be the set of tuples in $\mathbf{T}T_R^*$ restricted to attributes TS, QI, SD and B the set of tuples in $\mathbf{T}T^*$ restricted to L, QI, without repetitions, i.e., only one tuple per user. We define a rebuilding function $f : U^* \to B$ as an exhaustive function that, for each $f(t^*) = b$, satisfies:*

- *User $b[L]$ contains timestamp $t^*[TS]$.*
- *Tuple $t^*[QI]$ generalizes $b[QI]$.*
- *User $b$ appears in class $Q(t^*, T^*_{t^*[TS]})$.*

Observe how each rebuilding function defines a unique $f^{-1}$ that represents a possible correspondence between tuples and users, but not all rebuilding functions are feasible. That is why we now define reasonable rebuilding functions.

*Definition 35 [34]: A rebuilding function $f$ as defined in 34 is reasonable if it satisfies*

- *For all $b \in B$*
  - *$SD(f^{-1}(b))$ has only one sensitive attribute.*
  - *For all $x$ in some lifespan of $b[L]$, there exists a tuple $t^* \in f^{-1}(b)$ with $t^*[TS] = x$.*
- *The correspondence derived from $f^{-1}$ satisfies the generalization principles assumed by $\mathbf{T}^*$.*

Now observe that for each reasonable $f$ there exist a $\mathbf{T}$ and $\mathbf{T}^*$ such that $f^{-1}$ sends each user to the tuples derived from it.

*Theorem 6 [34]: If a dynamic dataset satisfies m-invariance, for any tuple $p$, not more than $\frac{1}{m}$ percent of reasonable rebuilding functions send that tuple to the correct sensitive attribute.*

*Proof:* Let $t$ be a user and $b$ be their corresponding tuple in $B$ as defined in 34. Let $f$ be any reasonable rebuilding function, and let $AQ(b, f)$ be the set of classes of each $T^* \in \mathbf{T}^*$ that contain at least a tuple in $f^{-1}(b)$.

We define a class over the set of all reasonable rebuilding functions where two functions $f, g$ are in the same class if and only if $AQ(b, f) = AQ(b, g)$. Let $cnt(F_i, sd)$ denote the number of surjections in the i-th class $F_i$ such that the sensitive value of $t$ is assigned as $sd$.

From m-invariance it is deduced that all classes in $AQ(b, f)$ have the same signature. Let $sd_1, \ldots, sd_x$ be that signature. From m-invariance, $x \geq m$.

Let $f_1$ be a function that reconstructs the sensitive attribute of $t$ as $sd_1$. We will construct a reasonable rebuilding function $f_2$ such that it reconstructs the sensitive attribute as $sd_2$. Let $QI$ be a class with two tuples $t_1^*, t_2^*$ such that $t_1^*[SD] = sd_1$ and $t_2^*[SD] = sd_2$, then $f_2(t_1^*) = f_1(t_2^*)$ and $f_2(t_2^*) = f_1(t_1^*)$. For all the other cases, $f_2(t^*) = f_1(t^*)$. It is straightforward to check that $f_2 \neq f_1$ and that $f_2$ is a reasonable rebuilding function. Notice that $AQ(b, f_1) = AQ(b, f_2)$, which implies $f_2 \in F_i$.

From the construction of $f_2$, we derive that $cnt(F_i, sd_1) \leq cnt(F_i, sd_2)$. Using the same argument with $f_2$ and $sd_1$ we have $(F_i, sd_1) = (F_i, sd_2)$. Again, using the same argument for all sensitive attributes, we conclude $(F_i, sd_1) = (F_i, sd_2) = \ldots = (F_i, sd_x) = \frac{|F_i|}{x}$ as claimed. □

*Corollary 3 [34]: For any database and attacker combination in Table 16, if $\mathbf{T}^*$ is m-invariant, then*

$$risk_0(p) \leq \frac{1}{m}$$

*for any user p. (as defined in Def 7)*

*Proof:* Since $risk(p)$ is the number of reasonable rebuilding functions that send the tuples derived from $p$ to the correct sensitive attribute over all reasonable rebuilding functions, from the proof of theorem 6 we have

$$risk_0(p) = \frac{\sum_{i=1}^{n_c} cnt(F_i, t[SD])}{n_{total}} \leq \frac{\sum_{i=1}^{n_c} |F_i|}{m \cdot n_{total}} = \frac{1}{m}$$

where $n_c$ is the number of classes, $n_{total}$ is the number of reasonable rebuilding functions, and the inequality follows from $(F_i, t[SD]) = \frac{|F_i|}{x} \leq \frac{|F_i|}{m}$. □

$\tau$-safety generalizes m-invariance to fully dynamic datasets and aware adversaries. The main particularity is that it assumes arbitrary updates, unlike m-invariance, and treats sensitive attribute updates as new tuple insertions.

*Definition 36 ($\tau$-safety [30], [31]):* $\mathbf{T}^* = \{T_1^*, \ldots, T_n^*\}$ *is $\tau$-safety if it satisfies the following conditions*:

- $T_j^*$ *is m-unique for all* $j \in [1, n]$.
- *For any tuple* $t \in \mathbf{T}T_R^*$ *with lifespan* $[x, y]$, *it is satisfied* $SD(Q(t, T_i^*)) = SD(Q(t, T_j^*))$ *for* $i, j \in [x, y]$.
- *For any tuple* $t$ *with consecutive lifespans* $[x, y], [i, j] \in L(t)$ *holds* $SD(Q(t, T_x^*)) = SD(Q(t, T_i^*))$.

*Theorem 7 ( [30], [31]): If a dataset satisfies $\tau$-safety, for any tuple p, not more than $\frac{1}{m}$ percent of reasonable rebuildings of the dataset send that tuple to the correct sensitive attribute at any timestamp.*

*Proof:* Notice that the argument of the proof of Theorem 6 can still be applied to the fully dynamic dataset if we assume that a tuple that has updated the sensitive attribute is a new tuple. This assumption can be made because we consider only arbitrary updates. □

*Corollary 4: For any* $\mathbf{T}$ *and any aware attacker combination in Table 16, if* $T^*$ *is $\tau$-safety, then*

$$risk_0(p) \leq \frac{1}{m}$$

*for any user p at any timestamp. (as defined in Def. 7 considering each tuple after a sensitive attribute update a new user)*

This definition of $\tau$-safety can be slightly generalized with the following version to not treat sensitive attribute updates as new tuple reinsertions.

*Definition 37 ($\tau$-safety v2 [47]):* $\mathbf{T}^* = \{T_1^*, \ldots, T_n^*\}$ *is $\tau$-safety if it satisfies the following conditions*:

- $T_j^*$ *is m-unique for all* $j \in [1, n]$.
- *For any tuple* $t \in \mathbf{T}T^*$ *with lifespan* $[x, y]$, *it is satisfied*

$$SD(Q(t, T_i^*)) = SD(Q(t, T_j^*))$$

*or*

$$SD(Q(t, T_i^*)) \cap SD(Q(t, T_j^*)) = \emptyset$$

*for each* $i, j \in [x, y]$.

- *For any tuple* $t$ *with consecutive lifespans* $[x, y], [i, j] \in L(t)$ *holds*

$$SD(Q(t, T_y^*)) = SD(Q(t, T_i^*))$$

*or*

$$SD(Q(t, T_y^*)) \cap SD(Q(t, T_i^*)) = \emptyset.$$

Like m-invariance, $\tau$-safety does not protect against probabilistic attacks, that is, because the guarantee of both methods relies on the quantity of possible reconstructions of the dataset and not on their reasonability, i.e., it uses the Naive Random World Assumption.

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | 20 | MALE | FLU |
| 2 | 21 | FEMALE | FLU |
| 3 | 18 | MALE | ACNE |
| 4 | 22 | FEMALE | HIV |

(a) Raw Table

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | [18-20] | MALE | FLU |
| 3 | [18-20] | MALE | ACNE |
| 2 | [21-22] | FEMALE | FLU |
| 4 | [21-22] | FEMALE | HIV |

(b) $\tau$-safe Table

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | 20 | MALE | FLU |
| 3 | 18 | MALE | ACNE |
| 4 | 22 | FEMALE | AIDS |
| 5 | 21 | FEMALE | CROUP |

(c) Updated Table

| ID | AGE | SEX | SD |
|---|---|---|---|
| 1 | [18-20] | MALE | FLU |
| 3 | [18-20] | MALE | ACNE |
| 4 | [21-22] | FEMALE | AIDS |
| 5 | [21-22] | FEMALE | CROUP |

(d) $\tau$-safe Table

**FIGURE 4. Historic values and 2-diverse publications of a dataset. Example of probabilistic attack.**

*Example 11: Observe Fig. 4, the releases are $\tau$-safe. An aware adversary knows that tuple* 4 *was updated. Before and after the update, the possible sensitive attributes of tuple* 4 *are FLU,HIV, and CROUP,AIDS respectively. It is more likely that tuple* 4 *had HIV and evolved into AIDS than any other combination. Such an attack on the sensitive attribute of tuple* 4 *was not prevented with $\tau$-safety since it assumed an arbitrary update behaviour, which is not satisfied.*

The only publication that addresses the attack shown in 11 is due to Amiri et. al. [37], and a general solution for such attacks is still pending.

### 3) m-DISTINCT

The notion of m-Distinct [29], [40] appears as an improvement in m-invariance. It studies how to provide protection when the dataset is fully dynamic. When a tuple can be updated, we assume that there is a certain pool of options to which it can be updated; that set is called the Candidate update set.

*Definition 38 ([29] (Candidate Update Set)): Let a be a particular value of an attribute A. We denote as $CUS(a)$ the set of possible values of attribute A to which a can be updated.*

*Definition 39 [29]: (Update Set Signature) Let Q be a class in T that contains n records and $SD_R(Q) = \{sd_1, \ldots, sd_n\}$. Then we define the Update Set Signature $USS(Q)$ of Q as the multiset $\{CUS(sd_1), \ldots, CUS(sd_n)\}$*

*Definition 40 [29]: (Legal Update Instance) A set of sensitive attributes $S = \{sd_1, \ldots, sd_n\}$ is a legal update instance of a $USS(Q)$ of some Q if*:

- $|S| = |USS(Q)|$.
- *For all* $sd \in S$, *there exists some* $CUS \in USS(Q)$ *such that* $sd \in CUS$.
- *For all* $CUS \in USS(Q)$ *there exists* $sd \in S$ *such that* $sd \in CUS$.

We finally state the definition of m-Distinct.

*Definition 41 [29]: (m-Distinct) Let $T^*$ be the historic releases of a fully dynamic dataset. $T^*$ is m-Distinct if*:

- *For all* $i \in [1, n]$, $T_i^*$ *is m-unique.*
- *Let* $t \in T_i \cap T_j$ *for some* $i, j \in [1, n]$ *with* $i < j$ *and* $t_i^*$ *its anonymized version in* $T_i^*$, *then for all* $i \in [1, n]$, $SD_R(Q(t^*, T_i^*))$ *is a legal update instance of* $USS(t_i)$

The capacity of m-Distinct can be slightly improved with the following version.

*Definition 42 [29]: (m-Distinct') Let $T^*$ be the historic releases of a fully dynamic dataset. $T^*$ is m-Distinct' if:*

- *It is m-Distinct.*
- *For any user that appears for the first time as a tuple $t^*$ in $T^*$, for all pairs $CUS_i, CUS_j \in USS(t^*)$, holds $CUS_i \cap CUS_j = \emptyset$.*

This notion guarantees the following notion of security.

*Theorem 8 [29]: If a historic release of a fully dynamic dataset is m-Distinct' then the republication risk satisfies $risk(p) \leq 1/m$ for all users p.*

The main limitation of this method is that the second condition of m-Distinct' cannot always be satisfied.

#### 4) Pvr-safety

The notion of ct-pvr-safety (compromised tuple-value restriction) [36] appears to give protection under the assumption that some tuples have been compromised, for example, if the attacker appears in the database or some information has been leaked.

**TABLE 17. Cases where pvr-safety gives protection against interior attacks.**

| | INCREMENTAL | DYNAMIC | FULLY DYNAMIC |
|---|---|---|---|
| INTERIOR | × | × | |

*Definition 43 [36]: Let p be a user, $S_p$ the set of possible sensitive attributes of p, $T^*$ an historic dataset, and K the compromised tuples in the dataset. We define the function $ct-pvr(p, S_p, T^*, K)$ as*

$$ct-pvr(p, S_p, T^*, K) = S_p \setminus \{a \in S_p \mid \exists i \in [1,n], \exists Q \in T_i^* \mid \forall t \in Q,\ t[SD] = a \Rightarrow t \in K\}$$

*where Q is a class of $T_i^*$. In words, the image is the set of possible sensitive attributes of p minus the set of sensitive attributes that we can dismiss with the knowledge of the compromised tuples K. To do this, we check at each class of each release if some sensitive attribute of such class was from a compromised tuple. If that is the case for all the instances of a particular sensitive attribute, then we can guarantee that it does not belong to our target user p.*

*Definition 44 [36]: (hc-pvr function) Let p be a user, $S_p$ the set of possible sensitive attributes of p, and $R(T^*, p)$ be the sets of users in historical correlation with p, we define the function*

$$hc-pvr(p, S_p, R(T^*, p)) = S_p \setminus \{a \in S_p \mid \exists R \in R(T^*, p), \forall r' \in R, a \notin S_{r'}\}$$

*In other words, the image is the set of possible sensitive attributes of p minus the set of sensitive attributes that cannot be of a historically correlated tuple. Recall that two historically correlated sets of tuples have the same sensitive attributes; if one of the sets cannot have a particular attribute, neither can the other.*

To present the guarantee of privacy, first a generalization of m-invariance is presented.

*Definition 45 [36]: (weak m-uniqueness) An anonymized table $T^* = \{Q_1, \ldots, Q_k\}$ is weak m-unique if each class $Q \in T^*$*

- *Contains at least m tuples with different sensitive attributes.*
- *All the sensitive attributes in $SD_R(Q)$ have the same number of occurrences.*

*Definition 46 [36]: (weak m-invariance) A historic dataset $T^*$ satisfies weak m-invariance if*

- *For all $i \in [1,n]$, the set $T_i^*$ satisfies weak m-uniqueness.*
- *For all anonymized tuples $t_1^*, t_2^*$ if $t_1^*, t_2^* \in Q_i$ and $t_1^* \in Q_j$ then $SD(Q_i) = SD(Q_j)$.*

*Definition 47 [36]: (hc-safety) Let $T^*$ be a historic dataset and Q a class of some $T_i$ with $i \in [1,n]$. Q is hc-safe with degree n if either*

1) *No set of users is historically correlated with the set of users in Q.*
2) *The cardinality of each set of historically correlated users with the users of Q is greater or equal to n.*

*Definition 48 [36]: ((m,n)-historically safety) Given $m, n \in \mathbb{N}$ with $n \leq m$ and $T^*$ a historic dataset. $T^*$ is historically safe (normally refers to the generalization function) if*

- *All $T_i$ satisfy weak m-invariance.*
- *Each class $Q \in T_i$ for all $i \in [1,n]$ is hc-safe with degree n with respect to $\{T_1^*, \ldots, T_i\}$.*

The second condition is the one that protects against historic correlations.

*Definition 49 ( [36](pvr-safe)): A historic dataset $T^*$ is pvr-safe with threshold $h \in (0,1]$ if, for each tuple $t \in T_i$ for any $i \in [1,n]$ holds*

$$p_{pb}(t) < h$$

*where $p_{pb}$ is the probability that a privacy breach of tuple t can occur using ct-pvr and hc-pvr functions.*

*Theorem 9 [36]: Let q be the probability that a particular tuple is compromised, L the maximum number of times that a single tuple can be republished, $h \in (0,1]$ the threshold for pvr-safety, and $m \in \mathbb{N}^+$ the required level of weak m-uniqueness. If there exists the smallest natural number $n \in \mathbb{N}^+$ such that*

$$\left(1 - (1-q)^L \cdot \left(1 - \left(q - \frac{q}{m}\right)^n\right)^{L\lfloor \frac{m}{n} \rfloor}\right)^{m-1} < h$$

*then if $T^*$ is (m,n)-historically safe, it is also pvr-safe with threshold h.*

#### 5) OTHERS

Here we state other methods that provide some interesting insight but that their presentation would be too large to simply state or that they do not give a clear guarantee of privacy.

#### a: HD-COMPOSITION

HD composition [32] was defined to give privacy against limited adversaries in the presence of permanent sensitive attributes in fully dynamic databases. To do that, a system of holder and decoys is used, i.e., whenever a tuple with a permanent sensitive attribute (a holder) appears in the dataset, a set of tuples with non-permanent sensitive attributes (decoys) are associated with it. With this system, no limited adversary can deduce who the tuple with the sensitive attribute is. The guarantee of privacy that HD composition gives is the following.

*Theorem 10 [32]: If $T^*$ a historic dataset of a fully dynamic database satisfies HD-composition (or refined HD-composition) then*

$$risk_1(p, sd, n) \leq \frac{1}{l}$$

*for any user p, any sensitive attribute sd, and any time n, where l is the parameter of HD-composition.*

#### b: BAYESIAN-BASED PROTECTION

In [37], a method to protect against probabilistic attackers is presented as an improvement of [48]. It focuses on mimicking the procedure that an attacker would do with a Bayesian approach and then edits the publications to reduce the amount of information that can be extracted with such methods. Its main limitation, however, is that it does not present a clear and close guarantee of privacy.

#### c: MICROAGGREGATION METHOD

In [41], a method intended for fully dynamic datasets is presented. It does not explicitly give any strong guarantee of privacy, but it is remarkable since it uses microaggregation instead of generalization or anatomization, unlike many other publications in the field.

### D. ANONYMIZATION ALGORITHMS

The first work on continuous data release was proposed by Byun et. al. [11]. The authors were the first to identify how independent releases of the same underlying table can cause redundant computations, vulnerabilities, and low utility. To mitigate some of these issues and particular vulnerabilities, they propose ''buffering'' records until $l$-diversity is guaranteed. Therefore, whenever an equivalence class is split, it is checked to see if an inference is possible via different attacks. However, we note that this is problematic since those checks are made with all previous releases, which increases the workload as more datasets are released. In the current context of big data, it seems unreasonable from a utility and security standpoint.

Xiao et. al. [34] continued the work of [11] but with dynamic datasets. In their proposal, the deletions can cause a critical absence, i.e., the disappearance of a sensitive attribute. To address this problem, the proposed algorithm adds counterfeits to the classes whenever necessary to ensure the m-invariance guarantee. This guarantee gives conditions on the structure of the tuple classes, imposing that each class has, at most, one sensitive attribute of each kind, at least $m$ different sensitive attributes, and that two classes where a common tuple is contained must have the same sensitive attributes. The process by which the algorithm creates the releases is via *bucketization*. The tuples are split into groups, each of which has a common signature; then each group is balanced, filled, and changed until m-invariance is satisfied. During this process, some counterfeits are added. While this is a great improvement with respect to [11], since the algorithm is less dependent on previous releases (making it much more efficient and general), the addition of counterfeits is hard to avoid, and it can be problematic for some sensitive datasets, i.e., sparse datasets with many sensitive attributes or those related to making far-reaching decisions, such as medical trials.

Another work that addresses the incremental case is [49]. The main idea is generalizing the tuples in a consistent manner to keep a monotonic rule in the information released (same information or less generalization). More specifically, the algorithm checks if a tuple can be less generalized at each release. While the proposed solution provides more utility than previous algorithms, it assumes a weaker attacker which makes it unsuitable to counter against background knowledge attacks, in particular those with information of the quasi identifiers. The proposed method is therefore not reasonable for most plausible attackers, which makes it an unlikely option.

Fung et. al. [33] proposed so-called *correspondence (BCF) attacks*. The authors showed how these attacks are hard to prevent optimally, showing that the problem is NP-hard. The algorithm searches on (a taxonomy tree) a minimal release in the sense that it is as refined as possible and still prevents BCF-attacks. Such attacks are only defined for the incremental case, and a deeper study in more general cases is still pending (or correspondence with other attacks). The proposed algorithm assumes a not powerful attacker and only provides security for BCF attacks, which makes it unreasonable for most situations.

The authors of [50] extended [34] with a critique of m-invariance. They argue that this privacy notion comes at the expense of a significant utility loss. Besides, they claim that, if some sensitive information is leaked, a data breach can occur under this guarantee. To tackle these two issues, the proposed solution uses a random noise system that groups each sensitive attribute into a set where each element is a possible sensitive attribute of the tuple. Due to the particular publication system, it may not be suitable for classical models that assume a specific value for each entry.

The fully dynamic case was first tackled by Bu et. al. [32]. It makes a differentiation between transient and permanent sensitive attributes, stating that protection for permanent sensitive attributes is, in general, unfeasible. To protect tuples with permanent sensitive attributes (holders), the proposed solution uses other tuples (decoys) with transient sensitive attributes to hide which one corresponds to a permanent

attribute. Although the system works for fully dynamic datasets, the protection is limited to weak adversaries and assumes the number of holders is reduced. The utility of the proposed algorithm was studied in range queries, which is a common practice in differential privacy methods.

Numerous aspects related to internal/external updates as well as the use of counterfeits were investigated in [29]. A thorough analysis of the assumptions on such updates resulted in the m-distinct (and m-distinct') guarantee. Although the proposed notion provides one of the strongest guarantees of security, it is not always achievable since it depends on certain properties of the dataset to be protected. A slight improvement of this work is presented in [51].

The authors of [52] presented an alternative to generalization and anatomization, where the sensitive attributes of the different classes are permuted among the tuples in it. The algorithm implements permutations in such a way that the m-invariance guarantee is satisfied. It represents an alternative to previous algorithms, but, depending on the metric of utility, it can yield an extremely lossy system.

Riboni et. al. [36] studied the interior attacker, in which an attacker knows the sensitive attributes of some tuples. The authors showed probabilistic formulas for the capacity of an attacker, assuming it has some bounded amount of sensitive information. In contrast to other methods, the protection given is a bound on the probability that an attack succeeds instead of a bound on the probability of a correct linkage between users and attributes. The algorithm structure is similar to the one in [34] but with an additional step to impose "$(m, n)$-historical safety", which prevents some interior attacks.

A related work is [54], which presents its own method to impose a guarantee of security on the probability that any tuple can be linked with their sensitive attribute. While it does not provide definitions or concepts of security, the proposed method constitutes a good standalone system to ensure privacy for the fully dynamic dataset case.

The 'equivalence attack' was developed in [35] and investigated an attacker's ability to relate sets of tuples with common sensitive attributes, without necessarily knowing their precise sensitive values. The authors proved that the decision problem of knowing if an equivalence attack of a particular size exists in a dataset is NP-hard. As argued in Sec. IV-B, these attacks represent a very dangerous threat since they allow for very powerful cascade effects once the sensitive attribute of some tuple is compromised. The proposed algorithm enforces m-invariance, since it was the state-of-the-art in security at that moment, and $e$-equivalence, which protects against equivalence attacks of $e$ or lower size.

Anjum et. al. [30], [31] proposed $\tau$-safety, a notion of security that extends m-invariance to fully dynamic datasets with arbitrary updates, i.e., the behaviour in which the sensitive attribute of the tuples change does not depend on previous values. The authors proved that $\tau$-safety implies that the capacity of any attack is bounded. This notion of security represents the first attempt to safely publish fully dynamic datasets. Sadly, the arbitrary update assumption (see Sec. IV-C) is not always reasonable since in most datasets the sensitive attribute updates are very dependent on the previous values. To date, a general notion of privacy for fully dynamic datasets is still lacking, and no significant improvements have been made since the publication of $\tau$-safety. For any reasonably strong non-probabilistic attacker, $\tau$-safety and m-invariance are the security guarantees to compare with.

The authors of [41] show an alternative approach where microaggregation is presented as an alternative to generalization and anatomization. The algorithm creates a Voronoi diagram where each region is defined by its centroid (with respect to the tuples inside). Each region contains $k$ to $2k - 1$ elements and is divided whenever the size exceeds $2k$. Although it is simple and mathematically tractable for the fully dynamic case, however, it does not prove or guarantee the security of the tuples.

A completely different approach is [55] and [56], which is an anonymization algorithm that aims to handle bigger datasets using fuzzy systems and cuckoo filters. While this approach provides a large improvement in terms of computation efficiency with respect to previous works, its main limitation is the lack of a clear guarantee of security for the information that is processed with their systems.

Lastly, [37] studied the probabilistic attacker from a Bayesian perspective. The most remarkable aspect of this work is that it is the only one that considers the capacity of an attacker to decide, among all possible raw underlying datasets, which one is more likely to be the generator of the published dataset; that is, it assumes the Smart Random World assumption. However, again, the main limitation is the lack of a definition of the privacy guarantee assumed. A deeper study of such probabilistic attackers and related ones is still a strand of future research.

Table 18 shows a summary of the anonymization algorithms examined in this section. The data fields in the table are described next:

- METHOD: indicates which procedure has been used to achieve protection; it can be generalization, the use of fake/counterfeit tuples, permutation of QI (QIT-PT), anatomization, microaggregation or segmentation.
- DYN. NOTION: core notion of privacy used in the publication.
- DATASET: which dataset is considered in terms of updateability.
- ADVERSARY: which adversary is considered.
- EMPIRICAL STUDY: which utility metric is used to show the capacity of the proposed model.

### E. METRICS OF UTILITY

As important as it is to guarantee data subjects' privacy, it is also to keep the utility of the dataset. In general, evaluating the utility of a perturbed, protected dataset is a challenging task, and a variety of metrics have been proposed to tackle it.

**TABLE 18. Main algorithms for continuous data publishing.**

| PAPER | METHOD | DYN. NOTIONS | DATASET | ADVERSARY | EMPIRICAL STUDY |
|---|---|---|---|---|---|
| [11] | GENERALIZATION | | INCREMENTAL | COMPLETE | INFORMATION LOSS |
| [34] | GENERAL.+FAKES | M-INVARIANCE | DYNAMIC | COMPLETE | AGGREGATE QUERY |
| [49] | GENERALIZATION | | INCREMENTAL | INCOMPLETE | DISCERNABILITY PENALTY SCORE |
| [33] | GENERALIZATION | BCF-ANONYMITY | INCREMENTAL | TARGET | DISCENIBILITY COST |
| [50] | QIT-PT | | DYNAMIC | COMPLETE | |
| [32] | GENERALIZATION | HD-COMPOSITION | FULLY DYNAMIC | LIMITED | RANGE QUERIES |
| [29] | GENERAL.+FAKES | M-DISTINCT | FULLY DYNAMIC | COMPLETE | QUERY ERROR |
| [53] | ANATOMIZATION | | | | |
| [52] | PERMUTATION | ORDER RANDOMIZATION | FULLY DYNAMIC | MINIMAL | |
| [36] | GENERAL.+FAKES | WEAK M-INVARIANCE (M,N)-HISTORICAL SAFETY | DYNAMIC | INTERIOR | QUERY ERROR |
| [51] | GENERAL.+FAKES | | FULLY DYNAMIC | | QUERY ERROR |
| [54] | GENERAL.+FAKES | GLOBAL GUARANTEE | FULLY DYNAMIC | COMPLETE | RELATIVE ERROR |
| [35] | ANATOMIZATION/ GENERAL+FAKES | GRAPH ANONYM. EQUIVALENCE ATTACK | DYNAMIC | | RANGE QUERIES |
| [30] | GENERAL.+FAKES | T-SAFETY | FULLY DYNAMIC | COMPLETE | QUERY ERROR |
| [31] | GENERAL.+FAKES | T-SAFETY ARBITRARY UPDATES | FULLY DYNAMIC | COMPLETE | CERTAINTY PENALTY QUERY ACCURACY |
| [47] | GENERAL.+FAKES/ SEGMENTATION | (L,K)-DIVERSITY T-SAFETY | FULLY DYNAMIC | | NORMALIZED CERTAINTY PENALTY QUERY ERROR |
| [41] | MICROAGGREGATION | | FULLY DYNAMIC | COMPLETE | INFORMATION LOSS |
| [55] | GENERAL.+FAKES | PMF/ M-SIGNATURE | FULLY DYNAMIC | | INFORMATION LOSS |
| [56] | ANATOMIZATION | CUCKOO FILTER | DYNAMIC | | ONLY OF DEL INSERT UPDATE |
| [37] | GENERALIZATION | BAYESIAN APPROACH | FULLY DYNAMIC | PROBABILISTIC | |

Next, we briefly examine some of the most relevant metrics. Before we proceed, however, we show a table with the datasets employed to evaluate the utility of the anonymization algorithms examined in Sec. IV-D.

**TABLE 19. Datasets employed by the most relevant anonymization algorithms.**

| DATASET | USED IN |
|---|---|
| UCI | [11], [12], [30], [31], [33], [41], [49], [53], [55], [57], [58] |
| IPUMS | [12], [29], [34]–[36], [59] |
| IPUMS (USA) | [47], [56] |
| BKSEQ | [37] |
| CADRMP | [32] |

**TABLE 20. Metrics of utility used in continuous data publishing.**

| UTILITY METRIC | USED IN |
|---|---|
| QUERY ERROR | [29]–[32], [34]–[36], [47], [51] |
| INFORMATION LOSS [11] | [11], [41], [55] |
| DISCERNIBILITY METRIC [60], [61] | [33], [49] |
| NCP [62] | [31], [47] |
| KL-DIVERGENCE [63] | [31] |

### 1) QUERY ERROR

In most cases, not all information is needed, but only some statistical information about it. When that is the case, query error is the usual metric used.

*Definition 50: A query is a request for information from a dataset. An aggregate/count query is a query that asks for the number of tuples in the dataset satisfying certain restrictions. The restrictions are of the form* $(A_1, \ldots, A_n) \in D_1 \times \ldots \times D_n$, *where each* $A_i$ *is an attribute of the tuple and each* $D_i$ *is a subset of the domain of the attribute* $A_i$. *If all* $D_i$ *with* $i \in [1, \ldots, n]$ *are intervals, it is a range query.*

The query error metric computes the difference between the results of the raw dataset and its anonymization.

*Definition 51: Let* $T$ *be a dataset and* $T^*$ *its anonymization. The query error metric computes the error of a count query as*

$$E = \frac{|CQ(T^*) - CQ(T)|}{CQ(T)}$$

*where* $CQ(T)$ *is the answer of the count query on dataset* $T$.

### 2) INFORMATION LOSS

Measures the intensity of the generalization process applied to each tuple. The definition assumes a finite domain for all attributes.

*Definition 52 [11]: Let* $T^*$ *be an anonymized dataset, and let* $Q$ *be a class of* $T^*$ *where each tuple has* $m$ *quasi identifiers. The information loss of* $Q$ *denoted as* $IL(Q)$ *is*

$$IL(Q) = |Q| \sum_{j=1}^{m} \frac{|G_j|}{|A_j|}$$

*where* $|A_j|$ *indicates the domain size of attribute* $j$ *and* $|G_j|$ *indicates the length of the interval of attribute* $j$.

When we refer to the length of the interval attribute, we are assuming some form of measure on the attribute domain. For example, if the attribute represents age and has a value $[18 - 30]$, normally we would establish the classical real measure of intervals, which yields $30 - 18 = 12$. If the data is qualitative but has some linear order, we could establish a discrete distance. For example, *bad* $<$ *mediocre* $<$ *good* $<$ *excellent* $<$ *superb* where we could establish the length of an interval as the number of elements inside minus one.

#### 3) DISCERNIBILITY

The discernibility is the capacity to distinguish elements. Since most protection methods try to make several tuples indistinguishable from each other, this metric computes how severe such a transformation is.

*Definition 53: [60]: Let $\{Q_1, \ldots, Q_n\}$ be the disjoint classes of a dataset $T^*$. The discernibility metric with parameter k is equal to*

$$DM(T^*, k) = \sum_{\substack{i \in \{1,\ldots,n\} \\ s.t.\ |Q_i| \geq k}} |Q_i|^2 + \sum_{\substack{j \in \{1,\ldots,n\} \\ s.t.\ |Q_j| < k}} |T^*||Q_j|$$

The objective of the parameter is to punish the existence of classes with less than $k$ tuples, i.e., penalize when the algorithm does not generate classes big enough.

#### 4) NORMALIZED CERTAINTY PENALTY

A natural approach to utility metrics is checking how much the attributes have changed. With Normalized Certainty Penalty (NCP), you can condense that information with the bonus that you can also assign weights to the different attributes depending on their importance.

*Definition 54: Let $t \in T$ be a tuple and $t^* = \{[x_1, y_1], \ldots, [x_n, y_n]\}$ be its generalized tuple with attributes $A_1, \ldots, A_n$. The NCP on attribute $A_i$ of $t^*$ is*

$$NCP_{A_i}(t^*) = \frac{y_i - x_i}{|A_i|},$$

*where $|A_i|$ is $\max\{t[A_i] \mid t \in T\} - \min\{t[A_i] \mid t \in T\}$. The weighted certainty penalty of $t^*$ with weights $(w_1, \ldots, w_m)$ is*

$$NCP(t^*) = \sum_{i=1}^{n} w_i NCP_{A_i}(t^*),$$

*where each $w_i$ is a non-negative real number.*

#### 5) KL-DIVERGENCE

In general, most previous metrics were exhaustive and computationally inefficient. A more competent method consists of using the KL-Divergence, which captures differences in the distributions of $T$ and $T^*$.

*Definition 55: Let $t_1, \ldots, t_n$ be the tuples of $T \cup T^*$ and let $p_i$ be the probability of $t_i$ in $T$ and $p_1^*$ its probability in $T^*$. We define KL-divergence as*

$$KL(T, T^*) = \sum_{i=1}^{n} p_i \log \frac{p_i}{p_i^*}.$$

The KL divergence is often referred to as *relative entropy*, as it may be regarded as a generalization of the Shannon entropy of a distribution relative to another. Although the KL divergence is not a distance function, because it is neither symmetric nor satisfies the triangle inequality, it does provide a measure of discrepancy between distributions, in the sense that $KL(T, T^*) \geqslant 0$, with equality if, and only if, $T = T^*$.

## V. DISCUSSION: RECENT ADVANCEMENTS AND RESEARCH DIRECTIONS

The work carried out in this paper achieves a classification of the different types of attackers, data publishing scenarios, types of risk, and allows for a more complete view of the state of the art on continuous data publishing. Now we devote ourselves to provide a better understanding of the most recent research directions and gaps in the literature.

This section identifies subareas of the field of continuous data publishing for which missing or inadequate contributions limit the ability of privacy designers to develop technology that provides both high privacy and high utility guarantees in practical, realistic data release scenarios. We start the discussion contexualizing existing notions and guarantees of privacy.

The anonymization algorithm proposed in [37] is intended to counter probabilistic adversaries, but, as with many other works in the literature, no clear guarantee of security is proved for the intended attack (or for other types of attacks). Similarly, although m-invariance and $\tau$-safety handle most attackers and datasets, notions defined for weaker adversaries that yield higher utility would of course be of interest. Likewise, [36] provides guarantees for the external dynamic interior attack but not for the fully dynamic case, where again a clear notion of security is still needed. Besides, whereas new privacy notions are necessary to deal with the most complex scenarios (e.g., fully dynamic datasets and/or probabilistic attackers), the existing ones are not comparable in most cases. In this sense, we believe an empirical comparison of the main notions via privacy guarantees, utility, and capacity would provide a better insight into which ones are better or worse for a given scenario.

With regard to applications, SRS, which handles multiple tuples for the same user, needs further attention. The same happens with the study of rare events/attributes, which seems to have been deprecated.

In the most complex case of fully dynamic datasets, we notice that quasi identifiers can provide information about sensitive attributes. Consider, for example, a quasi identifier that corresponds to ''status'' level. If one of the sensitive attributes is salary, it can be learned from the evolution of ''status'' (through several releases) that the salary must have been updated to a higher value. Notice, however, that this is not the same as Sensitive Background Knowledge, since the information is derived from the changes in the quasi identifiers and not from their particular values.

Another aspect that needs to be dealt with is multi-valued sensitive attributes, i.e., the case where each tuple has several sensitive attributes and the absence of particular entries in the tuples of the dataset are all possibly interesting situations. The anonymization of scarce datasets while preserving their already reduced utility is still far from solved.

Since the field is fairly new, no deep study of algorithmic complexity has been done. Finding lower bounds for the anonymization of a database, optimality/minimality, the

complexity of several algorithms, and their equivalence with other problems has been almost left apart, with small exceptions that can be derived directly from the NP-hardness of the static case.

Regarding anonymization methods or strategies, the generation of counterfeits (i.e., fake records) may imply a significant danger to the utility of the dataset. Searching for a solution without counterfeits that preserves high utility or investigating their necessity and the guarantees that can be only guaranteed with them are naturally two open strands of research of utmost importance, with initial proposals in [39] and [64].

There are currently no implementations or packages available in the public domain. This makes it difficult to develop papers dedicated to improving usability while maintaining the same levels of protection such as in [65] and [66]. A common workspace is needed to demonstrate the practical capabilities of each algorithm. It is also necessary to find realistic dynamic databases, as in most cases they are created artificially. All together this causes increased complexity to replicate and improve the results from the literature.

Some study has been done in the particular case of decremental datasets, that is, dynamic datasets which only accept tuple deletions [67], [68]. Other frameworks, such as the one presented by Hossain et. al. [69], study trajectory data as a particular intricate case of continuous data publishing and are also interesting spaces of study.

## VI. CONCLUSION

Anomymization is the tool that allows the circumvention of the legal restrictions applicable to personal data. In this work, we tackle the problem of anonymizing dynamic microdata, that is, how to protect the next publication of a changing microdata so that all previously anonymized versions of it, when combined with this next release, do not compromise individuals' privacy.

Within this rather general problem, we focus on syntactic privacy protection and continuous data publishing, where the microdata, in between releases, can experience record additions, deletions, insertions, and updates.

Since the first anonymization algorithm for continuous data publishing, a variety of contributions have been proposed aimed at addressing database types, adversaries, attacks, and notions. However, the vast majority of anonymization algorithms come with notions of privacy and adversarial models crafted specifically for the occasion, with their own nomenclature and notation, and very often without an evaluation of previous work. All this poses serious challenges to the scientific community and may compromise the development and maturity of the research field at hand.

To mitigate these issues, we have made several contributions. First, we have proposed a theoretical framework that unifies concepts, terminology, notions, and nomenclature in the literature. Our second and main contribution is a comprehensive taxonomy of the field of continuous data publishing, organized in terms of datasets, attackers and attacks, metrics of utility, notions of privacy, and their guarantees. In our extensive analysis of the literature, we have covered the most relevant contributions in anonymization techniques, provided numerous illustrative examples, and introduced novel definitions of all those aspects to harmonize the state of the art. Thirdly, we have addressed the practical problem of finding out which privacy guarantee one could obtain for several combinations of dataset and adversary model. And finally, we have elaborated on research gaps and future directions on anonymization technology.

## REFERENCES


[1] P. Golle, "Revisiting the uniqueness of simple demographics in the U.S. population," in *Proc. 5th ACM Workshop Privacy Electron. Soc.* New York, NY, USA: ACM, Oct. 2006, pp. 77–80.

[2] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. S. Nordholt, K. Spicer, and P.-P. de Wolf, *Statistical Disclosure Control*. Hoboken, NJ, USA: Wiley, 2012.

[3] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Lang., Program.* Berlin, Germany: Springer-Verlag, 2006, pp. 1–12.

[4] C. Clifton and T. Tassa, "On syntactic anonymity and differential privacy," *Trans. Data Privacy*, vol. 6, no. 2, pp. 161–183, 2013.

[5] J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "The limits of differential privacy (and its misuse in data release and machine learning)," *Commun. ACM*, vol. 64, no. 7, pp. 33–35, Jun. 2021.

[6] S. D. C. D. Vimercati, S. Foresti, G. Livraga, and P. Samarati, "Data privacy: Definitions and techniques," *Int. J. Uncertain Fuzziness Knowl. Based Syst.*, vol. 20, no. 6, pp. 793–818, 2012.

[7] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, pp. 1010–1027, Dec. 2001.

[8] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "$\ell$-diversity: Privacy beyond $k$-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, Atlanta, GA, USA, Apr. 2006, p. 24.

[9] N. Li, T. Li, and S. Venkatasubramanian, "$t$-closeness: Privacy beyond $k$-anonymity and $\ell$-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 106–115.

[10] K. Wang and B. C. M. Fung, "Anonymizing sequential releases," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, Aug. 2006, pp. 414–423.

[11] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure anonymization for incremental datasets," in *Secure Data Management*. Berlin, Germany: Springer, 2006.

[12] E. Shmueli, T. Tassa, R. Wasserstein, B. Shapira, and L. Rokach, "Limiting disclosure of sensitive data in sequential releases of databases," *Inf. Sci.*, vol. 191, pp. 98–127, May 2012.

[13] E. Shmueli and T. Tassa, "Privacy by diversity in sequential releases of databases," *Inf. Sci.*, vol. 298, pp. 344–372, Mar. 2015.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2013.

[15] M. Templ, A. Kowarik, and B. Meindl, "Statistical disclosure control for micro-data using theRPackagesdcMicro," *J. Stat. Softw.*, vol. 67, no. 4, pp. 67–85, 2015.

[16] L. Willenborg and T. DeWaal, *Elements of Statistical Disclosure Control*. New York, NY, USA: Springer, 2001.

[17] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2008, pp. 111–125.

[18] L. Sweeney, "$k$-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[19] H. Jian-Min, C. Ting-Ting, and Y. Hui-Qun, "An improved V-MDAV algorithm for $\ell$-diversity," in *Proc. Int. Symposiums Inf. Process.*, May 2008, pp. 733–739.

[20] M. Bewong, J. Liu, L. Liu, and J. Li, "Privacy preserving serial publication of transactional data," *Inf. Syst.*, vol. 82, pp. 53–70, May 2019.

[21] J. Bambauer, K. Muralidhar, and R. Sarathy, "Fool's gold: An illustrated critique of differential privacy," Arizona Legal Studies Discussion Paper nos. 13-47, James E. Rogers College of Law, Univ. Arizona, Tucson, AZ, USA, 2013.
[22] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. 23rd USENIX Conf. Secur. Symp. (SEC)*. San Diego, CA, USA: USENIX Association, 2014, pp. 17–32.
[23] S. Ruggles, C. Fitch, D. Magnuson, and J. Schroeder, "Differential privacy and census data: Implications for social and economic research," *AEA Papers Proc.*, vol. 109, pp. 403–408, May 2019.
[24] B. C. Leal, I. C. Vidal, F. T. Brito, J. S. Nobre, and J. C. Machado, "$\delta$-DOCA: Achieving privacy in data streams," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, J. Garcia-Alfaro, J. Herrera-Joancomartí, G. Livraga, and R. Rios, Eds. Cham, Switzerland: Springer, 2018, pp. 279–295.
[25] J. Parra-Arnau, T. Strufe, and J. Domingo-Ferrer, "Differentially private publication of database streams via hybrid video coding," *Knowl.-Based Syst.*, vol. 247, Jul. 2022, Art. no. 108778.
[26] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong, "Quantifying differential privacy in continuous data release under temporal correlations," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 7, pp. 1281–1295, Jul. 2019.
[27] B. Kitchenham, "Procedures for performing systematic reviews," Dept. Comput. Sci., Softw. Eng. Group, Keele Univ., Keele, U.K., Tech. Rep. TR/SE-0401, Jul. 2004.
[28] F. Bacchus, A. J. Grove, J. Y. Halpern, and D. Koller, "From statistical knowledge bases to degrees of belief," *Artif. Intell.*, vol. 87, nos. 1–2, pp. 75–143, Nov. 1996.
[29] F. Li and S. Zhou, "Challenging more updates: Towards anonymous re-publication of fully dynamic datasets," 2008, *arXiv:0806.4703*.
[30] A. Anjum and G. Raschia, "Anonymizing sequential releases under arbitrary updates," in *Proc. Joint EDBT/ICDT Workshops*. New York, NY, USA: ACM, Mar. 2013, pp. 145–154.
[31] A. Anjum, G. Raschia, M. Gelgon, A. Khan, S. U. R. Malik, N. Ahmad, M. Ahmed, S. Suhail, and M. M. Alam, "$T$-safety: A privacy model for sequential publication with arbitrary updates," *Comput. Secur.*, vol. 66, pp. 20–39, May 2017.
[32] Y. Bu, A. W. C. Fu, R. C. W. Wong, L. Chen, and J. Li, "Privacy preserving serial data publishing by role composition," *Proc. VLDB Endowment*, vol. 1, no. 1, pp. 845–856, Aug. 2008.
[33] B. C. M. Fung, K. Wang, A. W.-C. Fu, and J. Pei, "Anonymity for continuous data publishing," in *Proc. 11th Int. Conf. Extending Database Technol., Adv. Database Technol.* New York, NY, USA: ACM, Mar. 2008, pp. 264–275.
[34] X. Xiao and Y. Tao, "$m$-invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. New York, NY, USA: ACM, Jun. 2007, pp. 689–700.
[35] Y. He, S. Barman, and J. F. Naughton, "Preventing equivalence attacks in updated, anonymized data," in *Proc. IEEE 27th Int. Conf. Data Eng.*, Apr. 2011, pp. 529–540.
[36] D. Riboni and C. Bettini, "*Cor − Split*: Defending privacy in data re-publication from historical correlations and compromised tuples," in *Scientific and Statistical Database Management*, M. Winslett, Ed. Berlin, Germany: Springer, 2009, pp. 562–579.
[37] F. Amiri, N. Yazdani, A. Shakery, and S.-S. Ho, "Bayesian-based anonymization framework against background knowledge attack in continuous data publishing," *Trans. Data Privacy*, vol. 12, pp. 197–225, Dec. 2019.
[38] T. Soontornphand and J. Natwichai, "Joint attack: A new privacy attack for incremental data publishing," in *Proc. 19th Int. Conf. Network-Based Inf. Syst. (NBiS)*, Sep. 2016, pp. 364–369.
[39] R. Khan, X. Tao, A. Anjum, S. R. Malik, S. Yu, A. Khan, W. Rehman, and H. Malik, "($\tau$,m)-slicedBucket privacy model for sequential anonymization for improving privacy and utility," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 6, p. e4130, Jun. 2022.
[40] X.-Y. Ren, P. Zhang, and Y.-Q. Zhou, "Distinct model on privacy protection of dynamic data publication," *Cluster Comput.*, vol. 22, no. S6, pp. 15127–15136, Nov. 2019.
[41] J. Salas and V. Torra, "A general algorithm for $k$-anonymity on dynamic databases," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, J. Garcia-Alfaro, J. Herrera-Joancomartí, G. Livraga, and R. Rios, Eds. Cham, Switzerland: Springer-Verlag, 2018, pp. 407–414.
[42] B. Seisungsittisunti and J. Natwichai, "An efficient algorithm for incremental privacy breach on (k, e)-anonymous model," in *Proc. 16th Int. Conf. Network-Based Inf. Syst.*, Sep. 2013, pp. 97–104.
[43] T. Soontornphand, N. Harnsamut, and J. Natwichai, "Privacy preservation based on full-domain generalization for incremental data publishing," in *Information Science and Applications (ICISA)*, K. J. Kim and N. Joukov, Eds. Singapore: Springer, 2016, pp. 577–588.
[44] A. S. M. T. Hasan and Q. Jiang, "A general framework for privacy preserving sequential data publishing," in *Proc. 31st Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Mar. 2017, pp. 519–524.
[45] J. Liao, C. Jiang, and C. Guo, "Data privacy protection based on sensitive attributes dynamic update," in *Proc. 4th Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Aug. 2016, pp. 377–381.
[46] Q. Wei, Y.-S. Lu, and L. Zou, "$\epsilon$-inclusion: Privacy preserving re-publication of dynamic datasets," *J. Zhejiang Univ.-Sci. A*, vol. 9, no. 8, pp. 1124–1133, Aug. 2008.
[47] H. Zhu, H.-B. Liang, L. Zhao, D.-Y. Peng, and L. Xiong, "$\tau$-safe $(l, k)$-diversity privacy model for sequential publication with high utility," *IEEE Access*, vol. 7, pp. 687–701, 2019.
[48] D. Riboni, L. Pareschi, and C. Bettini, "Preserving privacy in sequential data release against background knowledge attacks," 2010, *arXiv:1010.0924*.
[49] J. Pei, J. Xu, Z. Wang, W. Wang, and K. Wang, "Maintaining $k$-anonymity against incremental updates," in *Proc. 19th Int. Conf. Sci. Stat. Database Manage. (SSDBM)*, Jul. 2007, p. 5.
[50] J. Lee, H.-J. Ko, E. Lee, W. Choi, and U.-M. Kim, "A data sanitization method for privacy preserving data re-publication," in *Proc. 4th Int. Conf. Networked Comput. Adv. Inf. Manage.*, Sep. 2008, pp. 28–31.
[51] X. Zhang and H. Bi, "Secure and effective anonymization against re-publication of dynamic datasets," in *Proc. 2nd Int. Conf. Comput. Eng., Technol.*, vol. 7, 2010, pp. V7-399–V7-403.
[52] W. Choi, J. Ryu, W. Kim, and U. Kim, "Simple data transformation method for privacy preserving data re-publication," in *Proc. 1st IEEE Symp. Web Soc.*, Aug. 2009, pp. 209–212.
[53] G. Wang, Z. Zhu, W. Du, and Z. Teng, "Inference analysis in privacy-preserving data re-publishing," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 1079–1084.
[54] R. C. Wong, A. W. Fu, J. Liu, K. Wang, and Y. Xu, "Global privacy guarantee in serial data publishing," in *Proc. IEEE 26th Int. Conf. Data Eng. (ICDE)*, Mar. 2010, pp. 956–959.
[55] J. Le, D. Zhang, M. Nankun, X. Liao, and F. Yang, "Anonymous privacy preservation based on $m$-signature and fuzzy processing for real-time data release," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 10, pp. 3871–3883, Oct. 2020.
[56] O. Temuujin, J. Ahn, and D.-H. Im, "Efficient $\ell$-diversity algorithm for preserving privacy of dynamically published datasets," *IEEE Access*, vol. 7, pp. 122878–122888, 2019.
[57] X. Sun, H. Wang, and J. Li, "$\ell$-diversity based dynamic update for large time-evolving microdata," in *AI 2008: Advances in Artificial Intelligence*, W. Wobcke and M. Zhang, Eds. Berlin, Germany: Springer, 2008, pp. 461–469.
[58] P. Lv, "Utility-based anonymization for continuous data publishing," in *Proc. IEEE Pacific–Asia Workshop Comput. Intell. Ind. Appl.*, Dec. 2008, pp. 290–295.
[59] A. Anjum and G. Raschia, "Privacy-preserving data publication: A review on 'updates' in continuous data publication," in *Proc. Int. Conf. Inform., Commun. Technol.*, 2011, pp. 1–5.
[60] A. Skowron and C. Rauszer, *The Discernibility Matrices and Functions in Information Systems*. Dordrecht, The Netherlands: Springer, 1992, pp. 331–362.
[61] R. J. Bayardo and R. Agrawal, "Data privacy through optimal $k$-anonymization," in *Proc. 21st Int. Conf. Data Eng. (ICDE)*, Apr. 2005, pp. 217–228.
[62] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based anonymization using local recoding," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: ACM, Aug. 2006, pp. 785–790.
[63] D. Kifer and J. Gehrke, "Injecting utility into anonymized datasets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, NY, USA: ACM, Jun. 2006, pp. 217–228.

[64] H. Attaullah, T. Kanwal, A. Anjum, G. Ahmed, S. Khan, D. B. Rawat, and R. Khan, "Fuzzy-logic-based privacy-aware dynamic release of IoT-enabled healthcare data," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4411–4420, Mar. 2022.
[65] M. Zhang, X. Zhang, Z. Chen, and D. Yu, "A privacy-preserving approach for continuous data publication," in *Algorithms and Architectures for Parallel Processing*. Cham, Switzerland: Springer, 2020, pp. 441–458.
[66] A. Tobar, J. Castro, and C. Gentile, "A new mathematical optimization-based method for the m-invariance problem," 2023, *arXiv:2306.15371*.
[67] S. Riyana, N. Harnsamut, U. Sadjapong, S. Nanthachumphu, and N. Riyana, "Privacy preservation for continuous decremental data publishing," in *Image Processing and Capsule Networks*, J. L.-Z. Chen, J. M. R. S. Tavares, S. Shakya, and A. M. Iliyasu, Eds. Cham, Switzerland: Springer, 2021, pp. 233–243.
[68] S. Riyana, N. Riyana, and S. Nanthachumphu, "An effective and efficient heuristic privacy preservation algorithm for decremental anonymization datasets," in *Image Processing and Capsule Networks*, J. L.-Z. Chen, J. M. R. S. Tavares, S. Shakya, and A. M. Iliyasu, Eds. Cham, Switzerland: Springer, 2021, pp. 244–257.
[69] Md. M. Hossain, A. H. M. S. Sattar, and F. Wahida, "Privacy preserving serial publication of trajectory data," in *Proc. Int. Conf. Inf. Commun. Technol. Sustain. Develop. (ICICT4SD)*, Feb. 2021, pp. 130–135.

**ADRIÁN TOBAR NICOLAU** received the B.Sc. degree in mathematics from Universitat de les Illes Balears, Palma de Mallorca, Spain, in 2019, and the M.Sc. degree in advanced mathematics from Universitat Politècnica de Catalunya, in 2020, where he is currently pursuing the Ph.D. degree with the Department of Networking Engineering. He is also an Assistant Professor with the Department of Mathematics and Computer Science, Universitat de les Illes Balears. He received the Extraordinary End-of-Studies Award for the B.Sc. degree.

**JAVIER PARRA-ARNAU** received the M.S. degree in telecommunications engineering and the M.S. and Ph.D. degrees in telematics engineering from Universitat Politecnica de Catalunya, Spain, in 2004, 2009, and 2013, respectively. He is currently a Ramón y Cajal Researcher with Universitat Politècnica de Catalunya.

**JORDI FORNÉ** received the M.S. and Ph.D. degrees in telecommunications engineering from Universitat Politècnica de Catalunya (UPC). Currently, he is a Full Professor with the Telecommunications Engineering School, Barcelona—ETSETB. He is also with the Smart Services for Information Systems and Communication Networks (SISCOM) Research Group, leading the research team on data privacy.