Journal of Privacy and Confidentiality (2014) **6**, Number 1, 35–75

An Evaluation Framework for Privacy-Preserving Record Linkage

Dinusha Vatsalan*, Peter Christen†, Christine O'Keefe‡, and Vassilios S. Verykios§

1 Introduction

Linking data from multiple sources enables more sophisticated analysis and data mining by improving the quality of data through the identification and resolution of conflicting data values, the enrichment of data, and the imputation of missing values [30]. The analysis of integrated data can, for example, facilitate the detection of adverse drug reactions in particular patient groups, or enable the identification of terrorism suspects [4, 21].

The process of matching and integrating records that relate to the same entity from one or more datasets is known as 'record linkage', 'data matching', or 'entity resolution' [21, 30]. In computer science and statistics, a long line of research has been conducted in record linkage, based on the theoretical foundation provided by Fellegi and Sunter in 1969 [22]. Today, record linkage not only faces computational and operational challenges due to the increasing size of datasets, but also privacy and confidentiality challenges due to growing privacy concerns. Generally, record linkage is a challenging task because unique entity identifiers are not available in all the databases that are linked. Therefore, the common attributes available which are sufficiently well correlated with entities, known as quasi-identifiers (QIDs) [13], need to be used for the linkage. For databases that contain personal information about people, these common QID attributes generally include names, addresses, dates of birth, and other personal details. Using such information often leads to privacy and confidentiality concerns. The three key challenges that are associated with the record linkage problem are:

1. Scalability: The first challenge of record linkage is the scalability to large databases which is generally dependent on the complexity of the process. Assume two databases that are to be linked, $\mathbf{D^A}$ and $\mathbf{D^B}$, contain $n^A = |\mathbf{D^A}|$ and $n^B = |\mathbf{D^B}|$ records, respectively. In order to classify the record pairs (a, b) from these two databases ($a \in \mathbf{D}^A$ and $b \in \mathbf{D}^B$) into matches (i.e., pairs of records that refer to the same entity) and non-matches (i.e., pairs of records that refer to different entities), in a naïve approach the number of comparisons required is the product of the size of the two databases ($n^A \times n^B$) which is the bottleneck of the whole linkage process [6, 8]. This quadratic complexity makes naïve linkage not scalable to large databases. Blocking or indexing techniques

*Research School of Computer Science, Australian National University, Canberra, Australia, mailto:dinusha.vatsalan@anu.edu.au.

†Research School of Computer Science, Australian National University, Canberra, Australia, mailto:peter.christen@anu.edu.au.

‡Commonwealth Scientific and Industrial Research Organization, Canberra, Australia, mailto:Christine.O'Keefe@csiro.au.

§School of Science and Technology, Hellenic Open University, mailto:verykios@eap.gr.

can be used to overcome this problem [5] and will be discussed further below.

2. Linkage quality: Record linkage aims to classify the records compared across different databases into matches and non-matches based on the matching/comparison results [8]. It is commonly accepted that real-world data are 'dirty' [28], which means they contain errors, variations, values can be missing, or values can be out-of-date. Therefore, even when records that correspond to the same real-world entity are being compared using the values of their personal identifying details (QIDs), the variations and errors in these values will lead to ambiguous matches [5]. The exact comparison of QID values is therefore not sufficient to achieve accurate linkage results. Approximate matching as well as accurate classification techniques are needed to achieve high linkage quality [5].

3. Privacy: When personal information about people (contained in QIDs) is used for the linking of databases across organizations, then the privacy of this information needs to be carefully protected. Individual databases can contain information that is already highly sensitive, such as medical or financial details of individuals, or confidential business data. When linked, detailed information about individuals that is even more revealing might become available, such as for people who have certain chronic diseases and who also have financial problems; or confidential business information like the amount a company owes to all its suppliers. It is therefore paramount that the privacy of data used for record linkage across organizations, as well as the sensitive details of the matching results of such a linkage, are preserved throughout the linkage process [55].

Data privacy in data mining tasks (including record linkage) has gained significant attention in the research community [55]. The privacy challenge in linking different databases has led to an evolving research line in privacy-preserving record linkage (PPRL) [55]. PPRL attempts to identify records that refer to the same real-world entities from different databases without compromising the privacy of the entities represented by these records.

An example real-world PPRL application would be where a research team aims to study the correlations between different types of car accidents and resulting injuries. Such an analysis requires the linkage of databases from hospitals, health insurance companies, and the police [5]. Another example from the health domain is a health surveillance system that continuously links data from human health data, animal health data, and drug data, to monitor outbreaks of contagious diseases that could lead to epidemics or even pandemics [12]. Another application of current interest is where a national security agency needs to collect and link records from a diverse set of databases (such as communication providers, banks, airlines, immigration, and social security) to identify potential terrorism threats [4, 5, 12]. These example scenarios illustrate that common data from different organizations need to be linked, but privacy and confidentiality issues often arise which might prevent such record linkage applications.

In a PPRL project, the database owners (or data custodians) agree to reveal only selected information about matched records among them, or to an external party, such as a researcher. However, to identify the matched records, generally the (masked) QIDs need to be revealed between the parties involved in the PPRL process. Personal information

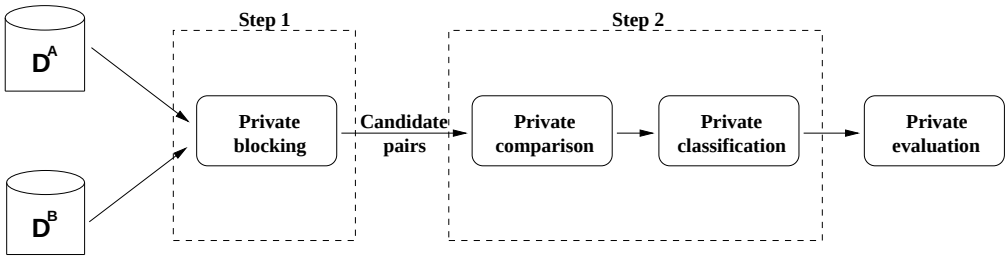

Figure 1: A general privacy-preserving record linkage pipeline.

contained in the QIDs is often not allowed to be shared or exchanged between different organizations due to privacy concerns or legal requirements. Therefore, the linkage has to be conducted on an encoded and/or perturbed version of the QIDs to preserve the privacy of entities. Encoding and/or perturbation is also known as 'masking', i.e., the original data is transformed in such a way that there exists a specific functional relationship between the original data and the masked data [24]. At the end of the linkage process, the database owners agree to reveal some of the selected attributes (only) of the record pairs that were classified as matches.

We consider the general pipeline of the PPRL process of two data sources $\mathbf{D}^A$ and $\mathbf{D}^B$, as outlined in Figure 1. The steps of this process and their challenges in a privacy-preserving setting are detailed in a recent survey [55]. The scalability challenge of PPRL can be addressed by using two-step algorithms, where in the first step (Step 1 in Figure 1) a private blocking or indexing technique is applied to reduce the number of candidate record pairs that need to be compared. For example, in a standard blocking approach [22] records are grouped into b blocks according to some criteria (known as blocking key), and candidate record pairs are generated from records in the same block (resulting in $(n^A \times n^B)/b$ candidate pairs). Private blocking for PPRL requires the identification of candidate record pairs in two databases without revealing the actual record values. Private blocking techniques that have been proposed for PPRL are surveyed in [55].

These candidate record pairs are then compared and classified into matches and non-matches in the second step (Step 2 in Figure 1) using private approximate comparison and effective classification techniques, addressing the linkage quality challenge [6]. A variety of private comparison and classification techniques has been used for PPRL, as surveyed in [55]. The complexity of PPRL also depends on the techniques employed in the linkage. Complex techniques for linkage, such as secure multi-party computation techniques [27, 39] or advanced classification techniques including machine learning or graph-based approaches [2, 29], generally have higher computational complexity (while providing high linkage quality and/or privacy), and therefore might not be scalable to large databases.

While various solutions have been developed to achieve PPRL, an evaluation scheme

to compare and assess the viability of these solutions (the final step in the pipeline shown in Figure 1) with respect to the three main challenges (or properties) of PPRL, which are scalability, quality of linkage, and privacy, has so far not been studied in the literature [55]. In this paper, we present an extensive evaluation framework for PPRL that models the scalability, linkage quality, and privacy based on an attack using an external global dataset, of PPRL solutions to provide an overall numerical score that can be used to evaluate and compare different solutions.

The remainder of this paper is structured as follows. We next provide an overview of PPRL and review privacy attacks and vulnerabilities of PPRL. In Section 3 we describe the evaluation model adopted to evaluate privacy in our framework. Section 4 presents the evaluation measures defined for each of the three properties of PPRL. Section 5 summarizes the PPRL techniques we will empirically evaluate and compare on real-world datasets in Section 6 by using the proposed framework. Finally, in Section 7 we summarize our findings and discuss future research directions.

2 Background

Over the years, various solutions for PPRL have been proposed as reviewed in [56, 55]. Privacy is addressed in these solutions using two different types of general approaches: (1) secure multi-party computation (SMC) techniques [27, 39] and (2) data perturbation (or masking) techniques [34, 57]. The former approach is generally more expensive with regard to the computation and communication complexity though it provides strong privacy guarantees, while the latter uses efficient techniques and, as opposed to SMC techniques, in many cases reveals a certain amount of information without compromising the privacy of sensitive data. However, due to the presence of partially revealed information, such perturbation techniques can be vulnerable to various types of attack.

The objective of PPRL is different from that of privacy-preserving data publishing or of statistical data disclosure [16]. Privacy-preserving data publishing masks a dataset in such a way that no identifying information about individuals can be inferred from the published dataset, while PPRL aims to identify matching records in two or more datasets without disclosing any sensitive information that can be used to identify individual records (and thus the entities they refer to) in the datasets. Therefore in data publishing, sensitive attributes which may contain some (masked) sensitive values (e.g., medical details) that are possibly disclosed with the (masked) QIDs that contain personal identifying information such as names and addresses. In PPRL on the other hand, only the (masked) QIDs are disclosed (only to the parties involved in the process) to allow the identification of matching records. We formally define PPRL as follows [55]:

Assume $\mathbf{O}_1, \ldots, \mathbf{O}_m$ are the m owners of the databases $\mathbf{D}^1, \ldots, \mathbf{D}^m$, respectively. They wish to determine which of their records $R_i^1 \in \mathbf{D}^1$, $R_j^2 \in \mathbf{D}^2$, ..., $R_k^m \in \mathbf{D}^m$ match based on their (masked) QIDs according to a decision model $C(R_i^1, R_j^2, \ldots, R_k^m)$ that classifies record pairs into one of the two classes $\mathbf{M}$ of matches, and $\mathbf{U}$ of non-matches. $\mathbf{O}_1, \ldots, \mathbf{O}_m$ do not wish to reveal their actual records $R_i^1, \ldots, R_k^m$ with any other party. They, however, are prepared to disclose to each other, or to an external party, the actual

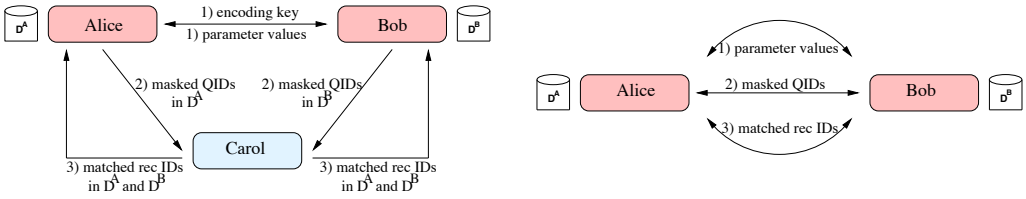

Figure 2: General three-party (left) and two-party (right) PPRL settings with linkage databases $\mathbf{D^A}$ and $\mathbf{D^B}$ and the data flow between parties. The numbers correspond to the order of the data flow in the protocols.

values of some selected attributes of the record pairs that are in class $\mathbf{M}$ for purposes such as statistical analysis.

We only address the problem of PPRL for two data sources in this paper. Assume *Alice* and *Bob* are two database owners with their respective databases $\mathbf{D^A}$ and $\mathbf{D^B}$ (generally referred as $\mathbf{D}$), who participate in a PPRL protocol to identify matching records in their databases that correspond to the same real-world entities under the privacy-preserving setting. Existing PPRL techniques can be categorized based on their need (or not) of a third party for performing record linkage [4, 55]. General settings of three-party and two-party protocols are illustrated in Figure 2. In three-party protocols, a third party, *Carol*, is involved in conducting the linkage, while in two-party protocols only the two database owners participate in the PPRL process. Three-party protocols are often not sufficient in real-world applications due to the absence of a trusted third party, since there is a risk of collusion between one of the database owners and the third party with the aim to learn the other database owner's sensitive data. Two-party protocols do not rely on a third party but they generally require more complex techniques to ensure that the two database owners cannot infer any sensitive information from each other during the linkage process [55].

The internal adversaries in a PPRL protocol are the parties involved in the process (*Alice*, *Bob*, and/or *Carol*). We assume that the parties involved follow the honest but curious behavior (HBC) [27, 39], in that they try to find out as much as possible about the data of the other parties while following the protocol. So far most developed PPRL techniques adopt the HBC model, as surveyed in [55]. It is important to note that the HBC model does not prevent collusion between parties [39]. There have been few PPRL techniques proposed for the malicious threat model [39] as well, where adversaries may behave arbitrarily. Proving privacy under the malicious model is more difficult because there exist several and potentially unpredictable ways for malicious parties to deviate from the protocol [55].

Two different general philosophies are adopted to preserve privacy and confidentiality of person-level data, which are restricted access and restricted data [16, 24]. To obtain effective results of privacy-preserving tasks, it is often preferred to have uncontrolled access to restricted data rather than restricted access to data [24]. Generally, restricted data is achieved in PPRL by first decoupling personal QID attributes from sensitive

attributes [36] and then by transforming the database ($\mathbf{D}$) into a masked version ($\mathbf{D^M}$), in order to protect the actual sensitive values in the database while preserving certain information to perform effective linkage.

Various privacy models have been used for data publishing, and different attacks have been studied in privacy preserving data publishing, including minimality attacks [58], deFinetti's theorem [35], and composition attacks [25]. However, most of these attacks are not applicable to PPRL since they use information from the (masked) sensitive attributes as well. Without sensitive attribute values disclosure, such attacks would not be possible. Several attack methods have been developed to investigate the privacy guarantees of perturbation-based PPRL solutions. The main attacks and vulnerabilities of PPRL defined in the literature include:

1. Dictionary attack: In dictionary attacks, it is assumed that the adversary knows the masking function (e.g., one-way hash function such as SHA and MD5 [55]) and potentially also the values of parameters used in a PPRL protocol, so that the adversary can mask a large list of common (global) values using the same masking function and parameter values as used in the PPRL protocol until a matching masked value is found. A keyed masking approach (such as HMAC) can overcome this problem by using a secret key for masking [55].

2. Frequency attack: Frequency attacks are still possible on the keyed masking approach (without knowing the secret key), where the frequency distribution of a set of masked values matches the distribution of known global values [40].

3. Cryptanalysis attack: Generally, Bloom filter-based PPRL techniques [18, 47, 51] are also susceptible to cryptanalysis attacks [37], where the bit distribution in a Bloom filter allows an adversary to learn the characteristics of hash functions that are used to map record values (e.g., q-grams) into a Bloom filter. This is similar to a frequency attack on bits and on the values or q-grams that are mapped to those bit positions.

4. Composition attack: Given auxiliary information (also called background knowledge [25]) about the individual datasets that are linked and/or certain records in the datasets, a composition attack can be successful by combining knowledge from more than one independent masked dataset to learn sensitive values of certain records [25].

5. Collusion: Another vulnerability associated with three-party and multi-party solutions is the collusion between some of the parties involved in the protocol (a sub-set of database owners and the third party) with the aim to learn the other database owner's data. Different types of scenarios might occur with regard to collusion, as will be discussed in Section 3.

Linkage attacks defined in the statistical disclosure community [16] are general terms for attack methods, that link a masked dataset to an external global dataset with known values using any subset of the previously discussed attacks in order to re-identify records and/or attribute values in the masked dataset. Based on such re-identification attacks, PPRL solutions can be evaluated for privacy guarantees. However, most of the PPRL solutions developed so far have not been properly evaluated in terms of the privacy aspect [55]. Some PPRL solutions provide theoretical proofs of the privacy techniques

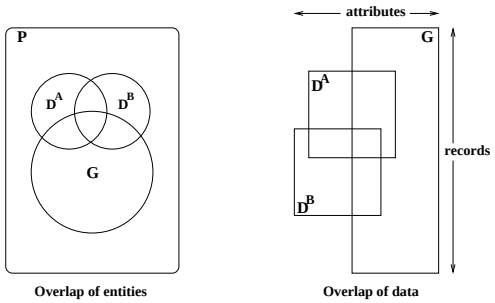

Figure 3: Overlaps of entities (left) and data (right) in the databases $\mathbf{D^A}$ and $\mathbf{D^B}$ and the global dataset $\mathbf{G}$. $\mathbf{P}$ is the total assumed population.

used in the solutions which makes the comparative practical evaluation of solutions difficult.

A general framework with a set of standard and numerical measures is therefore required to conduct such practical evaluation and comparison of PPRL solutions with respect to the three main properties of PPRL: scalability, quality, and privacy. we therefore propose a comprehensive evaluation framework that includes a wide range of measures for empirical evaluation of all three properties, and that enables quantifying and interpreting the performances of different PPRL solutions on the same scale.

3 Evaluation Model

Privacy evaluation requires assessing the risk of disclosure by calculating the probability that an adversary can correctly identify a value in a released dataset [16]. Such re-identification studies can be done through a linkage attack, as described in Section 2, using an available dataset, for example a publicly available global dataset such as a telephone book or an electoral roll. In this paper we assume the adversary is using a linkage attack for evaluating the privacy of PPRL solutions.

We assume that the adversary has access to a global dataset $\mathbf{G}$ that contains $N = |\mathbf{G}|$ unique values or combinations of values (for example, combinations of surname and first name values) of the population $\mathbf{P}$ from which the databases $\mathbf{D^A}$ and $\mathbf{D^B}$ are also drawn. This is reasonable because generally personal identifying attributes, such as names and addresses, are used for linkage and in many countries this background information is partially available in public resources (e.g., North Carolina (NC) voter registration data [7]). The individual databases that are used for the linkage ($\mathbf{D^A}$ and $\mathbf{D^B}$) can be considered as horizontal partitions of $\mathbf{G}$ (i.e., records overlap), while $\mathbf{G}$ can be a vertical partition of the linkage databases (attributes overlap). An overview of the overlaps of records and attributes in the datasets $\mathbf{G}$, $\mathbf{D^A}$, and $\mathbf{D^B}$ is illustrated in Figure 3.

In this paper we only consider insider attacks (which involve the internal adversaries

who are the database owners and/or the third party, as was discussed in Section 2) for privacy evaluation. We deem insider attacks to be the worst case because an insider adversary can be assumed to have more information than any external adversary, including knowledge about the PPRL protocol used, masking methods, and parameter values of the linkage techniques and algorithms used. It is important to note that a frequency attack might still be possible by an external adversary without this information. The possible scenarios for insider attacks in three-party and two-party protocols are:

- **Three-party protocols**

  In the first scenario, we assume that *Alice*, *Bob*, and *Carol* do not collude with each other. This case is much harder to attack because *Carol* does not know the encoding key and/or the parameter values used in the protocol, and *Alice* and *Bob* do not have access to the actual or masked values in each other's database. In this case, only a frequency attack might be possible by *Carol* depending on the PPRL protocol used.

  In the second scenario, one of the database owners (*Alice* or *Bob*) gets the other database owner's data (*Bob*'s or *Alice*'s, respectively) by colluding with the third party *Carol*. This is a worst case assumption because if two parties collude in such a way, then the privacy of the party that is not involved in the collusion cannot be assured. However, many three-party protocols assume a trusted third party (as reviewed in [55]) to reduce this risk of collusion. An alternative is to re-design a three-party protocol into a two-party protocol [51, 53, 59].

  Similar to the above scenario, *Carol* colludes with *Alice* or *Bob* in order to get the (secret) encoding key in the third scenario. Thereby it can conduct a dictionary attack using the key, and so can decode both *Alice*'s and *Bob*'s data. Instead of one of the database owners, the third party gets both database owners' data in this type of collusion. However, the colluding database owner in many cases would not like to reveal the (secret) encoding key because that would compromise the privacy of its own data as well.

  The first scenario, where no collusion between parties happens, is the best possible assumption. However, collusion can still happen in a HBC protocol [39]. The second and third scenarios are the worst case assumptions and they may be too unrealistic. Therefore, in this fourth scenario we assume that *Carol* knows only the masking function(s) and the parameter values used (and not the encoding key), either by colluding with *Alice* or *Bob*, or assuming or estimating parameter values with some background knowledge. *Carol* can perform an attack depending on the protocol, for example a cryptanalysis attack [37], with this knowledge to infer *Bob*'s or *Alice*'s values.

- **Two-party protocols**

  No collusion is obviously possible in two-party protocols. However, similar to the fourth scenario in three-party protocols described above, *Alice* and *Bob* know the masking function(s) and the parameter values used in the protocol, and as a result

they can perform attacks on the exchanged (masked) data between them to infer actual values from each other's data.

In the remainder of this paper, we assume that *Carol* knows the masking function(s) used in a PPRL protocol and knows or predicts the parameter values used in the protocol (fourth scenario for three-party protocols) to evaluate privacy of three-party protocols, similar to any two-party protocols. This assumption of an adversary's background knowledge (or partial knowledge) has been used in many attack methods that have been proposed in the literature [25, 35, 37, 58].

4 Evaluation Measures

The evaluation of a PPRL technique needs to be conducted in terms of the three properties of privacy, quality, and scalability. Quality and scalability correspond to the effectiveness and efficiency of a linkage process and can be assessed based on available standard measures (such as precision, recall, reduction ratio, pairs completeness, etc.) that will be discussed in Sections 4.2 and 4.3, respectively. However, the privacy protection provided by a PPRL technique is comparatively more difficult to assess. In the following Section 4.1, we present evaluation measures that can be used to evaluate the privacy aspect of PPRL. While the privacy measures based on information gain (see Section 4.1) have previously been used in PPRL [18, 33], the statistical disclosure risk measures based on probability of suspicion are novel. All the discussed measures will be experimentally evaluated in Section 6.

4.1 Privacy Measures

Privacy is normally measured as the risk of disclosure of information to the parties involved in a PPRL protocol (as will be described below in Section 4.1 in detail). As defined in statistical disclosure control [16], if an entity's confidential information can be identified in the disclosed (masked) data with an unacceptably narrow estimation, or if it can be exactly identified with a high level of confidence, then this raises a privacy risk of disclosure. A practical way of assessing disclosure risk is to conduct re-identification studies by linking values from a masked dataset to an external global dataset $\mathbf{G}$ [16].

We categorize the types of disclosure into record-level or identity disclosure, and attribute-level disclosure [17, 24]. Identity disclosure occurs when a record with multiple attribute values from the masked dataset $\mathbf{D^M}$ can be linked to an entity with the same attribute values in $\mathbf{G}$, which allows re-identification of the entity. It is important to note that a rare value (that only occurs in one or a small number of entities) for a single attribute could also lead to re-identification of the entity represented by that value by spontaneous recognition [17]. On the other hand, attribute-level disclosure allows an attribute value (characteristics) of an entity from $\mathbf{D^M}$ to be accurately re-identified.

Our method to evaluate privacy is to simulate attacks (as described in Section 2) on protected data in the masked dataset ($\mathbf{D^M}$) by linking them to the masked version ($\mathbf{G^M}$)

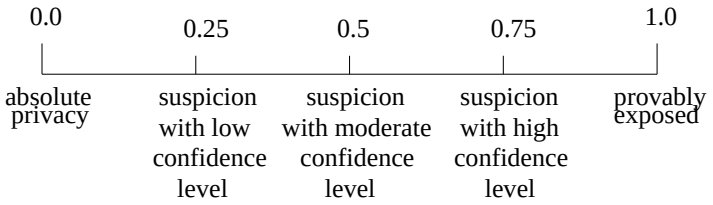

Figure 4: Degrees of privacy (adopted from [46]), ranging from absolute privacy, where the adversary cannot re-identify the actual value from the masked data, to provably exposed, where the adversary can provably re-identify the actual value.

of the known unprotected data in $\mathbf{G}$ [41]. A disclosure risk (DR) measurement that boils down to a numerical value to quantify the privacy protection of a PPRL technique based on such a simulation attack allows us to compare the privacy guarantees of several PPRL techniques.

The resulting DR measures are numerical values that are normalized between 0.0 and 1.0, where $DR = 0.0$ means no disclosure at all and $DR = 1.0$ means a provable disclosure (i.e., unique correct re-identification). These normalized values can also be specified as degrees of privacy as illustrated in Figure 4, following the work on degrees of connectivity or routing anonymity proposed by Reiter et al. [46]. In the following, we first consider DR of linkage using a single attribute in defining the privacy evaluation model, and then extend the model to include multiple attributes.

Disclosure Risk of Linkage Using a Single Attribute

If an attribute value a^M of a record R^M in a masked dataset ($R^M \in \mathbf{D}^M$) matches with exactly one value for the same attribute in $\mathbf{G}^M$, then there is a provably exposed risk of disclosure of a^M, because the masked value a^M can be identified with this one-to-one match. A value a^M that matches with a small number of values in $\mathbf{G}^M$ has a risk of suspicion with a high probability, while a value a^M that matches with possibly many values in $\mathbf{G}^M$ has a disclosure risk with a low probability. Absolute privacy is attained with values a^M that match with either no values in $\mathbf{G}^M$ (i.e., no background information is available), or with all the values in $\mathbf{G}^M$, or with a user-specified acceptable number of values k (as discussed below).

Given n_g is the number of global values in $\mathbf{G}^M$ that are matched with an attribute value a^M in the masked dataset $\mathbf{D}^M$, the probability of suspicion of a^M is calculated as $1/n_g$. We then normalize this probability into the 0.0 to 1.0 interval, where 1.0 indicates provably exposed risk and 0.0 represents absolute privacy, as defined in Equation 1 (with $N = |\mathbf{G}^M|$).

$$P_s(a^M) = \frac{1/n_g - 1/N}{1 - 1/N} \tag{1}$$

Table 1: Probability of suspicion (P_s) of values a^M in an attribute in a small (made-up) example masked dataset $\mathbf{D}^M$. The total number of a^M values is $n = 50$, and the total number of global values for the same attribute in $\mathbf{G}^M$ is $N = 1,000$. Values are sorted according to their $P_s(a_i^M), 1 \leq i \leq n$.

1.0	1.0	1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	0.33
0.33	0.25	0.25	0.2	0.2	0.2	0.2	0.2	0.2	0.1
0.1	0.1	0.1	0.1	0.1	0.01	0.01	0.01	0.01	0.01
0.002	0.002	0.002	0.002	0.0	0.0	0.0	0.0	0.0	0.0

Statistical disclosure risk measures: Using the probability of suspicion (P_s) values calculated for each of the values a^M in an attribute in $\mathbf{D}^M$, we present five different statistical disclosure risk (DR) measures to calculate the overall disclosure risk of the entire masked dataset $\mathbf{D}^M$.

As a running example, Table 1 shows the P_s values for a small made-up dataset of $n = 50$ values. This dataset contains, for example, five values of an attribute with $P_s = 1.0$, which means that these five attribute values match with only one attribute value out of $1,000$ in $\mathbf{G}^M$ (we assume $\mathbf{G}^M$ contains $1,000$ values of the same attribute), ten attribute values that match with two global values ($P_s = 0.5$), and six attribute values that match with either no values or all the $1,000$ values in $\mathbf{G}^M$ ($P_s = 0.0$).

1. Maximum risk (DR_{Max}): This measure allows us to define the maximum risk of disclosure of the masked dataset. It corresponds to the maximum value for the probability of suspicion P_s of attribute values a^M in the masked dataset, as explained in Equation 2.

$$DR_{Max} = \max_{a^M \in \mathbf{D}^M} (P_s(a^M)) \tag{2}$$

   In the example given in Table 1, the DR_{Max} is calculated as $DR_{Max} = 1.0$. This explains that the masked dataset has a maximum risk of 1.0 of any sensitive value being disclosed, i.e., there exists at least one attribute value in $\mathbf{D}^M$ that matches to a single value in $\mathbf{G}^M$.

2. Marketer risk (DR_{Mark}): It is important to know how many values in a masked dataset can be exactly re-identified. This risk is known as marketer risk and it evaluates the risk of disclosure from the perspective of a marketer adversary who wishes to re-identify as many values as possible in the disclosed dataset [14]. Marketer risk is measured as the proportion of values in $\mathbf{D}^M$ that have provably exposed risk of disclosure ($P_s = 1.0$) with one-to-one mapping in $\mathbf{G}^M$. DR_{Mark} for the running example in Table 1 is $5/50 = 0.1$, calculated using Equation 3 (as there are five of the fifty values having $P_s = 1.0$).

$$DR_{Mark} = |\{a^M \in \mathbf{D}^M : P_s(a^M) = 1.0\}|/n, \tag{3}$$

   where $P_s(a^M)$ is the probability of suspicion of a value a^M in $\mathbf{D}^M$ and $n = |\mathbf{D}^M|$.

3. Mean risk (DR_{Mean}): The mean risk calculates the average of probability of suspicion values to evaluate the average disclosure risk. DR_{Mean} is calculated using Equation 4. A value in the example masked dataset illustrated in Table 1 has an average probability of 0.28 of being re-identified, i.e., in average a value in $\mathbf{D}^M$ can be matched to around four values in $\mathbf{G}^M$.

$$DR_{Mean} = \frac{1}{n} \sum_{a^M \in \mathbf{D}^M} P_s(a^M). \tag{4}$$

4. Median risk (DR_{Med}): The median risk takes into account the distribution of probabilities of suspicions in the masked dataset and it gives the center of the distribution of disclosure risk values. DR_{Med} is calculated as shown in Equation 5, assuming $P_s(a^M)$ values are sorted in ascending order. DR_{Med} for the running example (with $n = 50$) results in $1/2 \times [P_s(a^M_{25}) + P_s(a^M_{26})] = (0.2 + 0.2)/2 = 0.2$.

$$DR_{Med} = \begin{cases} 1/2 \times [P_s(a^M_{n/2}) + P_s(a^M_{n/2+1})] & n \textit{ is even} \\ \\ P_s(a^M_{(n+1)/2}) & n \textit{ is odd.} \end{cases} \tag{5}$$

5. User acceptance (UA) mean risk (DR_{UAM}): If the users/data respondents of the linkage accept that the data will not be at a disclosure risk if a value a^M in their masked dataset matches with more than a certain number of values (k unique values) in the global dataset, then we can eliminate the risk of disclosing those masked values that are below the respective probability of suspicion, as the probabilities of suspicion of those values would be in the low confidence level, as shown in Figure 4. This approach is based on the concept of $(k, 1)$-anonymization mapping [26], where any value in a masked dataset is consistent with at least k original values and thus provides $(k, 1)$-anonymization privacy constraints. Ramachandran et al. [45] and Ferro et al. [23] proposed similar approaches to identify vulnerable records in a dataset that match with at most k global records in public data.

   The mean disclosure risk calculation can then be applied using Equation 4 after removing or setting to 0.0 the probabilities of suspicions that are acceptable by the users. For our running example, if the acceptable minimum number of global values that match with a single value in the masked dataset is set to $k = 4$ ($P_s = 0.25$), then in Table 1 we can set the probability of suspicion for the last 27 values (those with $P_s < 0.25$) to $P_s = 0.0$, and DR_{UAM} would then be calculated as $DR_{UAM} = 0.24$ using Equation 4.

We illustrate the distribution of P_s values in the example dataset shown in Table 1 and the calculated statistical disclosure risk measures in Figure 5. In Figure 6, we also present the distribution of P_s values in a real North Carolina (NC) voter dataset [7] and the disclosure risk measures calculated for a simple dictionary attack on hash encoded first name values using the same original NC voter dataset as the global dataset. As can be seen from these two figures, this set of statistical disclosure risk measures provide numerical and statistical information (maximum, mean, median, marketer, and UA mean) on the risk of disclosing a masked dataset.

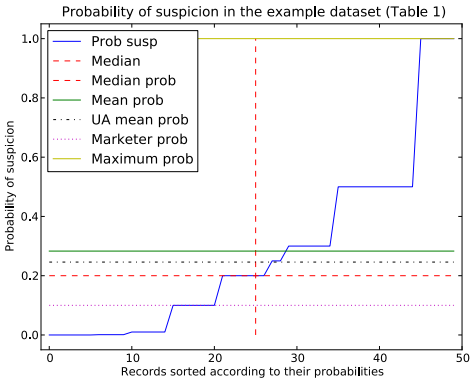

Figure 5: Distribution of probability of suspicion (P_s) values in the example dataset shown in Table 1 and the calculated statistical disclosure risk measures from Section 4.1. The acceptable minimum number of global values that match with a single value is set to $k = 4$ ($P_s = 0.25$) for DR_{UAM}.

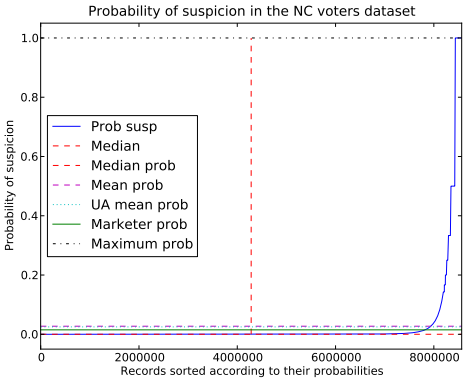

Figure 6: Distribution of probability of suspicion (P_s) of first name attribute values in the hash encoded NC voter dataset [7] and the calculated disclosure risk measures for a simple dictionary attack on hash encoded values using the same dataset as the global dataset. We set $k = 50$ for DR_{UAM}.

Information theory measures: The standard information theory measures, such as information gain (IG) and relative information gain (RIG) [18, 33], can also be used as DR measures based on a simulation attack on the masked dataset using the original dataset as the global dataset. IG assesses the possibility of inferring values in the original dataset $\mathbf{D}$, given its masked version $\mathbf{D}^M$. These information theory measures have been used for privacy evaluation in PPRL before [18, 33]. However, there are some limitations of these measures.

The first limitation is that the global dataset can only be assumed to be the same as the original linkage dataset ($\mathbf{G} \equiv \mathbf{D}$), while our statistical DR measures are independent of the choice of the global datasets. The second is that the IG measures provide only the overall total information gain from the masked dataset while our DR measures provide statistical summary information of the disclosure risk. We use a small example dataset shown in Table 2 to illustrate the calculation of IG and RIG.

Following the notation used by Durham [18] and Karaksidis et al. [33], the entropy $H(D)$ of a dataset $\mathbf{D}$ is defined as:

$$H(\mathbf{D}) = -\sum_{a \in \mathbf{D}} (n_g/N) log_2(n_g/N), \tag{6}$$

where n_g denotes the number of global values in $\mathbf{G}$ that match with a value a in $\mathbf{D}$, and N is the total number of values in $\mathbf{G}$. $H(D)$ is calculated for the example dataset with three made-up values (shown in Table 2) to 1.48, as explained in the left three columns in the table.

Table 2: Disclosure risk calculation of a small example dataset using IG and RIG. The global dataset is the same as the original dataset ($\mathbf{G} \equiv \mathbf{D}$) and the total number of global values in $\mathbf{G}$ is $N = n = 100$.

Original values in $\mathbf{D}$	Prob of values in $\mathbf{G}$ (n_g/N)	$log_2(n_g/N)$
peter	$30/100 = 0.3$	-0.522
pete	$20/100 = 0.2$	-0.464
smith	$50/100 = 0.5$	-0.5
$H(D) = -\sum(n_g/N)log_2(n_g/N)$ = 1.48		

Masked values in $\mathbf{D^M}$	Prob of values in $\mathbf{G}^M$ (n_g^M/N)	$H(\mathbf{D}\|\mathbf{D}^M = a^M)$
p360	$50/100 = 0.5$	$0.6 \times log_2 0.6 +$ $0.4 \times log_2 0.4 = 0.48$
s530	$50/100 = 0.5$	$1.0 \times log_2 1.0 = 0.0$
$H(D\|D^M) = -\sum(n_g^M/N) \times H(D\|D^M = a^M)$ = 0.48		

The conditional entropy of a dataset $\mathbf{D}$ given $\mathbf{D}^M$, $H(\mathbf{D}|\mathbf{D}^M)$, is [18, 33]:

$$H(\mathbf{D}|\mathbf{D}^M) = -\sum_{a^M \in \mathbf{D}^M} (n_g^M/N) H(\mathbf{D}|\mathbf{D}^M = a^M), \tag{7}$$

where n_g^M is the number of masked global values in $\mathbf{G}^M$ that match with a masked value a^M in $\mathbf{D}^M$, and N is the total number of values in $\mathbf{G}^M$. $H(\mathbf{D}|\mathbf{D}^M)$ for the running example is 0.48, as shown in the right three columns in Table 2. The entropy and conditional entropy form the basis for the information gain (IG) metric. IG between $\mathbf{D}$ and $\mathbf{D}^M$ is [18, 33]:

$$IG(\mathbf{D}|\mathbf{D}^M) = H(\mathbf{D}) - H(\mathbf{D}|\mathbf{D}^M). \tag{8}$$

The running example results in $IG = 1.48 - 0.48 = 1.0$. The lower the value for IG is, the more difficult it is for an adversary to infer the original dataset from a masked dataset. The relative IG (RIG) measure normalizes the scale of IG ($0.0 \leq RIG(\mathbf{D}|\mathbf{D}^M) \leq 1.0$) with regard to the entropy of the original dataset $\mathbf{D}$ [33], and is defined as $RIG(\mathbf{D}|\mathbf{D}^M) = IG(\mathbf{D}|\mathbf{D}^M)/H(\mathbf{D})$. This is calculated as $RIG = 1.0/1.48 = 0.67$ for the running example dataset. Since RIG values are normalized between 0.0 and 1.0, they provide a marginal scale for comparison and evaluation.

Disclosure Risk of Linkage Using Multiple Attributes

Record-level (or identity) disclosure is possible when multiple attributes are used for linking, as it is generally the case. Disclosure risk calculation for linking on multiple attributes can be done in three ways depending on the information available in the global dataset $\mathbf{G}$.

The first case is if the global dataset contains combinations of individual values for all attributes (m attributes) used in the linkage and/or blocking, and each combination refers to one single entity, then the disclosure risk calculation is similar to the single attribute disclosure calculation. For each record R^M in the masked dataset $\mathbf{D}^M$, the number of global records n_g that have the matching values in the same attributes of R^M is calculated and the probability of suspicion of R^M then is $P_s(R^M) = 1/n_g$. An example

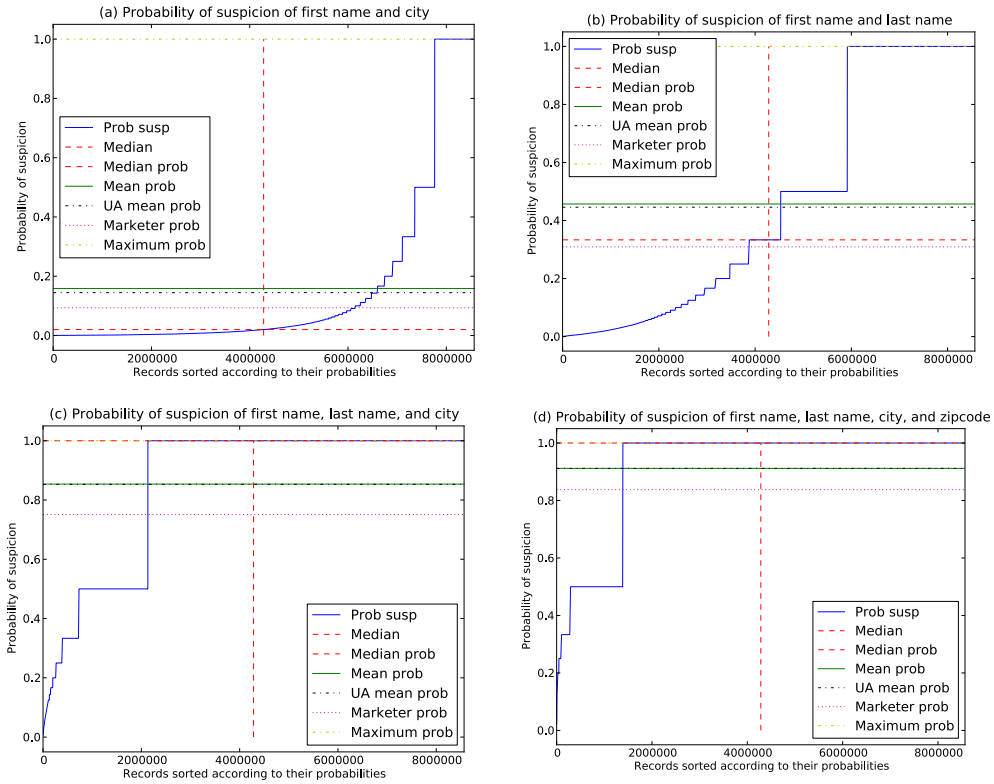

Figure 7: Distribution of probability of suspicion (P_s) of (a) first name and city, (b) first name and last name, (c) first name, last name, and city, and (d) first name, last name, city, and zipcode attribute values in the hash encoded NC voter dataset [7] and the calculated disclosure risk measures for a simple dictionary attack on hash encoded values using the same dataset as the global dataset. We set $k = 50$ for DR_{UAM} calculation.

would be if a combination of masked values of 'amilia' for the first name attribute and 'smith' for the last name attribute of a record R^M in $\mathbf{D}^M$ matches with $n_g = 2$ combinations/records in $\mathbf{G}^M$ that have the same masked values in the corresponding two attributes; then the probability of suspicion of R^M is calculated as $P_s(R^M) = 1/2 = 0.5$. This disclosure risk is higher than when only a single attribute is used in linkage, since multiple attributes (more information) of a record are compared with the entities in $\mathbf{G}$ that also have the same combination of attribute values (which could likely allow for an identity disclosure).

The distributions of probability of suspicion values in a real NC voter dataset [7] and the calculated disclosure risk measures for a dictionary attack on hash encoding of multiple attributes are shown in Figure 7. As the figure illustrates, when multiple attributes are used in linkage the disclosure risk becomes higher compared to the risk when only a

single attribute is used, as was shown in Figure 6. The probability of suspicion and the disclosure risk values become higher with more attributes used. The number of unique combinations of attribute values of first name and city is smaller than the number of unique combinations of first name and last name which results in lower disclosure risk values for the former, as can be seen in Figures 7(a) and 7(b), respectively. The probability of suspicion of the four attributes first name, last name, city, and zipcode provide a marketer risk of $DR_{Mark} = 0.84$, as shown in Figure 7(d). This is similar to the results by Sweeney [49] who showed that around 90% of the population of the USA have a unique combined value zipcode, gender, and date of birth.

The second case is where the global dataset $\mathbf{G}$ contains combinations of attribute values as in case 1, but a certain subset of attribute values of a record R^M in $\mathbf{D}^M$ do not match with any values in the corresponding attributes in $\mathbf{G}^M$. For example, a masked first name value of 'amilia' in $\mathbf{D}^M$ matches with $n_{g_1} = 2$ masked first name values in $\mathbf{G}^M$, but the corresponding masked last name value 'dickson' in $\mathbf{D}^M$ does not match with any global values ($n_{g_2} = 0$). In such a case, we calculate the probability of suspicion as $P_s(R^M) = 1/(n_{g_1} \times N) = 1/(2 \times 1,000) = 0.0005$, by considering all the global values in $\mathbf{G}^M$ as possible matches ($N = 1,000$ in this example) for masked values that match with zero global values.

In the third case, the combinations of attribute values are not available in $\mathbf{G}^M$ (i.e., $\mathbf{G}^M$ consists of individual lists of global values for each attribute, but not the combinations of different attribute values). In this case, we multiply the number of global values that match with each attribute of a record R^M in $\mathbf{D}^M$ individually, in order to calculate the total number of global values that match with the record R^M. The probability of suspicion for R^M in this case would be $P_s(R^M) = 1/(n_{g_1} \times n_{g_2} \times \cdots \times n_{g_m})$, where m is the number of attributes used for the linkage. For example, if a record R^M in $\mathbf{D}^M$ with masked values of 'amilia' and 'smith' for the first name and last name attributes matches with $n_{g_1} = 2$ global records in $\mathbf{G}^M$ that have the same (masked) first name value, and $n_{g_2} = 10$ global records that have the same (masked) last name value, then $P_s(R^M) = 1/(2 \times 10) = 1/20 = 0.05$.

4.2 Linkage Quality Measures

PPRL has to deal with the trade-off between privacy protection and the quality of linkage. Achieving more privacy generally means losing more data quality due to information lost in the protected/masked data as compared to the original data, and thus losing more quality of the linkage results. In practice, measuring the linkage quality is often difficult, because no truth data with known match status are available in many real-world applications [8]. However, the linkage quality can be assessed in a pilot study using synthetic data (representing real data characteristics) with known match status [10], or using the manual classification results obtained by clerical review in a record linkage process [5].

The quality of linkage in PPRL depends on both the quality of blocking as well as the quality of comparison and classification steps. The measures that are commonly used

in information retrieval and data mining, such as precision, recall, and f-measure [44], can be used to assess the quality of private comparison and classification results. The quality of blocking can be measured using the pairs completeness and pairs quality measures [5]. Based on the classification of the number of true matches (TM), false matches (FM), false non-matches (FN), true non-matches (TN), true matches included in the candidate record pairs generated by blocking (BM), and true non-matches included in the candidate record pairs (BN), the linkage quality measures are defined as given below.

1. Precision: the fraction of record pairs classified as matches by a decision model that are true matches: $Precision = TM/(TM + FM)$.
2. Recall: the fraction of true matches that are correctly classified as matches by a decision model: $Recall = TM/(TM + FN)$.
3. F-measure: the harmonic mean of Precision and Recall, calculated as $F\text{-}measure = 2 \times (Precision \times Recall)/(Precision + Recall)$.
4. Pairs completeness (PC): measures the effectiveness of a blocking technique (similar to Recall). $PC = BM/(TM + FN)$.
5. Pairs quality (PQ): measures the efficiency of a blocking technique and is similar to Precision: $PQ = BM/(BM + BN)$.

4.3 Scalability Measures

The third aspect of PPRL that makes the linkage process scalable to large real-world databases is dependent on the complexity of the protocol. The number of record pairs that are compared and classified using a PPRL technique determines the complexity of the protocol. A naïve pair-wise comparison of two databases is of quadratic complexity in the size of the databases [6]. Private blocking techniques [18, 32, 38, 52, 54] are used in the first step of PPRL to reduce this large number of comparisons by removing pairs that are unlikely to refer to matches without comparing them in detail in the next step.

The efficiency of a blocking technique can be measured using reduction ratio (RR) [5], which provides a value that indicates by how much a blocking technique is able to reduce the number of candidate record pairs that are being generated compared to all possible record pairs. Reduction ratio is calculated as $RR = 1.0 - (BM + BN)/(TM + FN + FM + TN)$.

The complexity of techniques (or algorithms) used in PPRL has also an impact on the scalability of the protocol. Generally the complexity of algorithms is measured using the big-O notation [43] and practically evaluated in terms of efficiency using measures that are dependent on the computing platform and the networking infrastructure used, such as the total runtime, the memory space required to perform the linkage, and the size of messages or data communicated between parties in the protocol. The challenge with these platform dependent measures is how to normalize them into the 0.0 to 1.0 interval, to allow comparison of several PPRL solutions. A possible way to evaluate runtime,

for example, is to calculate the average time required for a candidate record pair to be compared and classified using the most computationally intensive PPRL technique, and then multiply this value by the total number of candidate record pairs ($n^A \times n^B$, if no blocking is applied). This would give an upper bound for expected runtime. Then we can run all the PPRL solutions that need to be evaluated on the same computing platform, and measure their runtime. Using the upper bound calculated, the resulting runtime values can then be normalized between 0.0 and 1.0.

4.4 Overall Evaluation Score

A generic score can be calculated to evaluate PPRL techniques in terms of the three properties using the measures discussed in the above sections. For example, given the measures for disclosure risk (DR), linkage quality (LQ), and scalability (S), the overall evaluation score can be computed by using the weighted average of the three measures.

$$score = \alpha(1 - DR) + \beta(LQ) + (1 - \alpha - \beta)(S) \qquad 0 \leq \alpha + \beta \leq 1 \tag{9}$$

Different weights for the three properties can be used depending on the importance of the properties with respect to application or user preferences. This final numerical score indicates the viability of a specific PPRL solution in terms of privacy, linkage quality, and scalability. A graphical representation of the three properties of PPRL provides more insights into the analysis and comparison of different PPRL techniques. Three-dimensional plots can be used to define the three properties along the three axes of the graphs to compare PPRL solutions, as will be shown in Section 6.

5 PPRL Techniques

In this section, we summarize some of the PPRL techniques proposed in the literature which we will empirically evaluate and compare in Section 6 using our proposed evaluation framework. We also describe the methods for linkage attacks on those techniques using an external global dataset. Linkage attacks for randomized masking [24] with error bounds are out of the scope of this paper.

As explained by Duncan et al. [16], a drawback of using external datasets for risk calculation in disclosure control, is that the results are dependent on the choice of global datasets. Conducting linkage studies using a very large external dataset as the global dataset would require longer runtime and more computational resources which might not be practical for empirical evaluation. In addition, an external global dataset might not be available for privacy evaluation. In the worst case scenario, the global dataset $\mathbf{G}$ can be considered to be equivalent to the linked database $\mathbf{D}$ (i.e., $\mathbf{G} \equiv \mathbf{D}$). Conducting linkage studies of attacks such as frequency attacks, cryptanalysis attacks, and collusion using the masked dataset ($\mathbf{D^M}$) and the original dataset $\mathbf{D}$ as the global dataset would provide the highest disclosure risk in this worst case scenario. If a specific privacy technique provides sufficient privacy guarantees under such a worst case assumption, then the privacy technique would provide sufficient privacy in a real-world setting as

well, because the global dataset available to an adversary is highly likely to be different from the original dataset. If $\mathbf{G}$ is larger than $\mathbf{D}$, then there would possibly be many global values in $\mathbf{G^M}$ that match a masked value in $\mathbf{D^M}$, which therefore result in lower disclosure risk. On the other hand, if $\mathbf{G}$ is smaller than $\mathbf{D}$, there might be masked values in $\mathbf{D^M}$ that do not match with any global values in $\mathbf{G^M}$, again resulting in lower disclosure risk.

We consider the worst case assumption of $\mathbf{G} \equiv \mathbf{D}$ in this paper for privacy evaluation and comparison of several PPRL techniques in Section 6. However, the proposed framework can be used with any choice of global dataset (as long as all the techniques are compared for privacy against attacks using the same global dataset). First, in Section 5.1, we present the solutions proposed for private blocking (step 1 of the PPRL pipeline shown in Figure 1) and linkage attacks that can be applied on them, and then in Section 5.2 we present the solutions and linkage attacks for private comparison and classification (step 2 of the PPRL pipeline).

5.1 Private Blocking

Several techniques for private blocking have been proposed, and in this paper we choose the following state-of-the-art techniques to be empirically evaluated and compared using the evaluation framework.

k-NN: Karakasidis et al. [32] proposed a three-party private blocking based on k-nearest neighbor clustering and reference values. Initially, clusters are created for the set of reference values that are shared and known by both database owners using k-nearest neighbor clustering such that each cluster consists of at least k elements in the reference set to provide a k-anonymous privacy guarantee (i.e., each masked record is indistinguishable from k reference values by the adversary). Each database owner then assigns the blocking key values (BKVs) of their records to the respective clusters according to the *Dice-coefficient* similarity of q-grams [5] between the BKVs and the reference values. These clusters are sent to a third party that matches the corresponding clusters to generate candidate record pairs. A main drawback of this approach is that it requires calculation of similarities between each record and all the reference values.

HLSH: Durham [18] investigated how locality sensitive hashing (LSH) can be applied in Bloom filter-based PPRL to reduce the number of record pair comparisons. LSH allows hashing of values in such a way that the likelihood that two similar values are hashed into the same block can be specified through the use of certain hash functions. A Bloom filter is a bit array data structure where hash functions are used to map a set of elements (q-grams extracted from attribute values) into the bit array. For private blocking, an iterative approach was employed, where random bits are sampled in each iteration from the Bloom filters and sent to a third party. The third party then uses Hamming-based LSH functions to compute the Hamming distance (since the Jaccard distance-based LSH functions require longer runtime than the Hamming distance [18]), which allows efficient generation of candidate record pairs. It is difficult to tune this approach as it requires several highly sensitive and data dependent parameters that

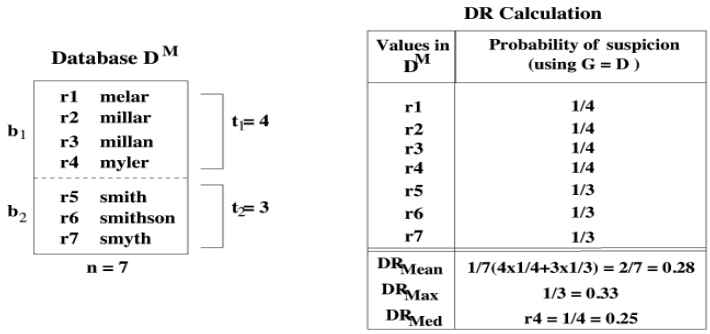

Figure 8: An attack method for three-party private blocking solutions [18, 32, 38, 52] using $\mathbf{G} \equiv \mathbf{D}$ and the statistical disclosure risk measures calculated for the linkage attack. Records $r_1 \dots r_4$ are consistent or similar with 4 records in the same block b_1 of size $t_1 = 4$ resulting in $P_s = 1/4$, while records $r_5 \dots r_7$ are consistent with 3 records in the same block b_2 of $t_2 = 3$ resulting in $P_s = 1/3$. The total number of records in $\mathbf{D}$ is $n = 7$.

have to be set with an acceptable trade-off among them [18].

SNC-3PSim and SNC-3PSize: The sorted neighborhood-based blocking [15, 28] used in traditional record linkage sorts database tables according to a 'sorting key' (an attribute or combination of attributes used to sort the records) over which a sliding window of size w is moved and candidate record pairs are generated from the records that are within the current window. This approach is very efficient compared to other blocking techniques in that its resulting number of candidate record pairs is $O((n^A + n^B)w)$, while with many other blocking techniques the number is $O((n^A \times n^B)/b)$ [6]. Due to its efficiency, the sorted neighborhood blocking has recently been considered for private blocking. Vatsalan et al. [52] proposed a three-party private blocking approach based on sorted neighborhood clustering (SNC) [15] and using a combination of the privacy technique k-mapping [26] and reference values. In this approach, the private database records are first inserted into the sorted list of public reference values according to their SKVs. This results in blocks containing one reference value and several SKVs that are sorted near the reference value. These blocks are then merged to generate k-anonymous blocks that contain at least k masked SKVs and one or more reference values. Two versions of k-mapping are proposed to generate k-anonymous blocks. The first is based on similarity between reference values (which we call **SNC-3PSim**) and the second on the size of blocks (**SNC-3PSize**). These k-anonymous blocks are sent to a third party that merges the corresponding blocks from the two database owners depending on the common reference values in the blocks to generate candidate record pairs.

Generally, in three-party private blocking techniques (the above described kNN, HLSH, SNC-3PSim, and SNC-3PSize), only the number of blocks (n_B) and the size of each block ($t_i = |b_i|, 1 \leq i \leq n_B$) are revealed to the third party that participates in the protocol. In the masked (blocked) dataset $\mathbf{D^M}$, a record r is consistent or similar with

$t_i - 1$ other records in the same block b_i where r resides. If r is consistent with t_i records (including r) in the local database then there would be at least t_i global matching values ($n_g \geq t_i$) in $\mathbf{G}$. Therefore the probability of suspicion of a record r in private blocking is $P_s = 1/t_i$ ($\geq 1/n_g$) under the worst case assumption ($\mathbf{G} \equiv \mathbf{D}$). The general attack method and DR calculation for three-party private blocking solutions are illustrated in Figure 8 with two small example blocks b_1 of size $t_1 = 4$ and b_2 of $t_2 = 3$.

HCLUST: Another privacy technique used in private blocking is differential privacy [20]. Recently Kuzu et al. [38] used differential privacy to add noise into the blocks generated using hierarchical clustering. A third party is not needed for blocking (two-party private blocking). Initially global clusters are generated for a set of reference values using hierarchical clustering. Then each database owner assigns their records into these global clusters based on their similarity. Differential privacy is used by adding noise drawn from a Laplace distribution to ensure privacy against inference due to clusters being revealed to the third party. Noise is added in the form of random new BKVs. A three-party SMC-based approach is then used in step 2 of the PPRL pipeline to compare and classify the candidate pairs generated in the blocking step [38]. This approach is computationally expensive in terms of the number of similarity calculations.

Random noise increases privacy by reducing the probability of suspicion. If r random values are added into a block b_i of size t_i, these t_i records will have the probability of suspicion $P_s = 1/(t_i + r)$, where originally it was $P_s = 1/t_i$. However, when adding extra records there is generally a trade-off between linkage quality (due to false matches of randomly added values), scalability, and privacy [33]. False matches of random values can also affect the privacy of the matched real values, since the privacy of a matched real value might be compromised due to a false match with a random value.

SNC-2P: Vatsalan et al. [54] recently converted the three-party SNC-based blocking [52] (described above) into a two-party private blocking approach. The two database owners use different sets of reference values for k-mapping to generate the k-anonymous blocks. Then they exchange certain reference values from each block over which a sorted neighborhood approach using a sliding window is conducted to determine candidate blocks from both databases.

A private blocking protocol that reveals more than the size (t_i) and number of blocks (n_B) during the protocol will provide more information on the distribution of blocks and their values. For example, the SNC-2P [54] protocol reveals reference values from each block, and the more reference values are exchanged from a block between the database owners the more information is disclosed about that block. Assume a block b_i (of size t_i) is represented by reference values $v = v_1, \ldots, v_e$ and their frequency distribution (individual block sizes) in $\mathbf{G}$ is learned as $\mathbf{f} = f_1, \ldots, f_e$. Revealing only one reference value (v_i) discloses that there are t_i records sorted near reference value v_i. But revealing several reference values discloses more information, namely that there are $f_i \times t_i / \sum f_i$ records sorted near reference value v_i, $i = 1 \ldots e$. This reduces the minimum number of global values n_g from t_i to $min(f) \times t_i / \sum f_i$. For example, if three reference values, 'melar', 'millar', and 'myler', in a block b_1 that contains nine records ($t_i = 9$) are exchanged and their frequency distribution in $\mathbf{G}$ is 'melar'= 2, 'millar'= 3,

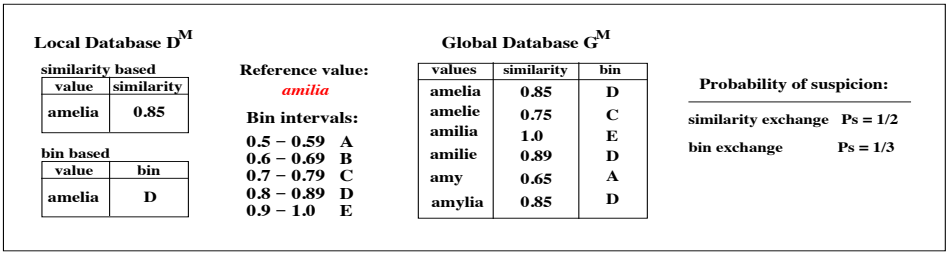

Figure 9: An attack method for reference values-based private comparison and classification solutions [42, 53]. The similarity of value 'amelia' (0.85) matches with two global values in $\mathbf{G^M}$ while the bin of similarity (D) matches with three global values and thus the P_s is reduced to 1/3 from 1/2.

and 'myler'= 4, then this reveals that there are $2t_1/9 = 2$ records sorted near 'melar', $3t_1/9 = 3$ near 'millar', and $4t_1/9 = 4$ near 'myler'. The minimum n_g now becomes 2 with the two records sorted near 'melar' and the maximum probability of suspicion P_s therefore increases to 1/2.

5.2 Private Comparison and Classification

A variety of privacy techniques has been used for private comparison and classification in PPRL [55]. We evaluate the following private comparison and classification solutions proposed in the literature which are based on two types of privacy techniques: reference values and Bloom filters.

2P-Bin: Pang et al. [42] introduced a three-party solution based on a set of reference values that are shared by both database owners. The database owners compute the distance based on the similarity between the reference values and their private attribute values, and they send the similarity values to a third party that classifies the pairs of values based on the triangular property of distance metrics. This approach was recently converted into a two-party setting by Vatsalan et al. [53]. In their approach, the similarity values calculated with the reference values are binned into intervals and instead of exchanging the actual similarity values the bins of similarity are exchanged between the database owners. Classification is conducted from the exchanged bin values-based on the reverse triangular property of distance metrics.

A frequency linkage attack method for reference values-based private comparison and classification solutions is explained in Figure 9. An adversary having access to a global dataset $\mathbf{G}$ can compute the number of matching values n_g in $\mathbf{G^M}$ that have the same similarity or bin of similarity with the same set of reference values to calculate the probability of suspicion P_s. DR measures can then be calculated using the P_s values for each value in $\mathbf{D^M}$. As illustrated in Figure 9, the exchange of bins of similarity reduces the probability of suspicion and thus increases the privacy guarantees compared to revealing the actual similarity values to a third party, as proposed in the three-party

solution [42] (assuming the third party might collude and/or it might have information about the reference values used). In addition, the number of bins used in the two-party solution determines the privacy of this approach. If the number of bins is large then the similarity range of each bin becomes smaller, and this results in a smaller number of global values n_g in $\mathbf{G^M}$ that match with a specific bin value. Therefore, the larger the number of bins the lower the privacy of the solution but the higher the quality of linkage.

2P-BF: Bloom filters are another promising privacy technique that has recently been used in several privacy-preserving solutions. Schnell et al. [47, 48] proposed a three-party Bloom filter-based private comparison and classification solution. The database owners map q-grams of attribute values of a record into a Bloom filter of length l using k hash functions by setting the corresponding bit positions to 1, and then send these Bloom filters to a third party that can calculate the similarity between the Bloom filters using a set based similarity function such as the Dice-coefficient in order to classify the pairs. This method of encoding is known as cryptographic longterm key (**CLK**) [48].

Durham [18] studied this approach in detail by using record-level Bloom filter encoding (**RBF**) to overcome the problem of cryptanalysis attack [37] associated with attribute (or field)-level encoding. The attribute values (q-grams) are hash-mapped into different Bloom filters, and then bits are selected from each of the attribute's Bloom filters according to their weights calculated based on Fellegi and Sunter's agreement and disagreement weights [22] (more bits are selected from attributes with higher weights) and frequencies (bits with certain frequencies are not included to improve privacy) in order to compose the RBF.

In a hybrid encoding we can combine both CLK and RBF (which we call **CLKRBF**) to select different numbers of hash functions k for different attributes according to their weights and map them into the same Bloom filter of length l. Having different numbers of hash functions for different attributes based on weights provides more accuracy as with RBF [18], and mapping them into the same Bloom filter improves privacy due to collision between bits as with CLK [48].

Bloom filter encoding is studied in a two-party context by Vatsalan et al. [51] by using an iterative classification method. In this approach, the database owners generate Bloom filter encodings and then iteratively exchange a certain number of bit positions from each of their Bloom filters. The minimum similarity is calculated from the bits revealed in order to classify the pairs of Bloom filters into matches, non-matches, and possible matches. The pairs that are classified as possible matches are taken to the next iteration where another set of bit positions is revealed to classify the remaining pairs. The number of bits to be revealed in each iteration is calculated in such a way that the non-matches are removed before revealing more bits [51].

We evaluate this **2P-BF** approach in Section 6 using the **CLK**, **RBF**, and **CLKRBF** encodings. A simple example of the attack method and the calculation of DR measures for Bloom filter-based private comparison and classification is presented in Figure 10. The main idea of a cryptanalysis attack [37] is that if a bit position is set to 0 in a Bloom filter, then all the possible matches (members or substrings of the string which

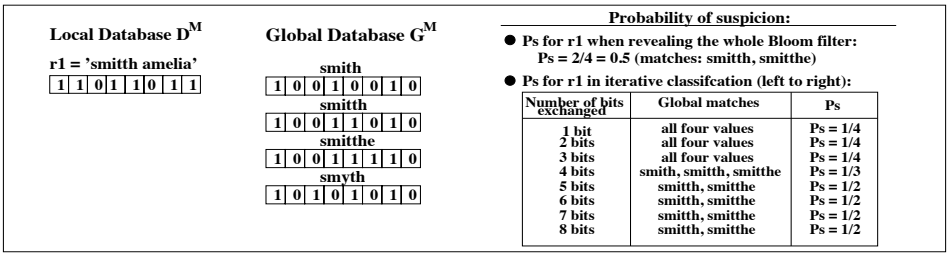

Figure 10: An attack method for Bloom filter-based private comparison and classification solutions. As the membership theory states [37], all the bit positions that are set to 0 in the Bloom filter of record r_1 must also be set to 0 in the Bloom filters in $\mathbf{G}^M$ that are possible matches to r_1. Hence, in the shown example two of four global values' Bloom filters ('smitth' and 'smitthe') in $\mathbf{G}^M$ match with the Bloom filter of r_1 and therefore $P_s = 1/2$. In the iterative classification approach, the probability of suspicion increases with more bits revealed.

is mapped to this Bloom filter) must not independently set the specific bit position to 1, as proven in [37]. In 2P-BF, the probability of suspicion increases with the number of bits revealed, as shown in Figure 10.

6 Experimental Evaluation

We used two real datasets to empirically evaluate and compare the solutions described in Section 5 by using our proposed evaluation framework.

1. OZ: The first database is an Australian telephone directory (OZ) that contains 6,917,514 records. We extracted four attributes commonly used for record linkage: given name (with 78,336 unique values), surname (404,651 unique values), suburb (town) name (13,109 unique values), and postcode (2,632 unique values). To generate datasets of different sizes, we sampled 0.1%, 1%, 10%, and 100% of records in the full database each for *Alice* and *Bob*, and stored them into pairs of files such that 50% of records appeared in both files. The record pairs that occur in both datasets are exact matches. These datasets are labeled as 'No-mod' for no modification.

To investigate the performance of PPRL solutions in the context of 'dirty data' (where attribute values contain errors and variations), we generated another series of datasets where we modified each attribute value by applying a randomly selected character edit operation (insert, delete, substitute, or transposition) [9]. These datasets are labeled as 'Mod' for modification. This leads to a much reduced number of exact matching record pairs and allows us to evaluate the quality of solutions in terms of the accuracy of approximate matching.

2. NC: The second database that we used is a large real-world voter registration database from North Carolina (NC) in the US [7], containing records of several million voters. We downloaded this dataset every two months since October 2011 to build a longitudinal dataset. As detailed in [7], we have done extensive data cleaning and

Table 3: The number of records in the datasets used for experiments, and the number of records that occur in both datasets of a pair (i.e. the number of true matches).

Dataset	*Alice*	*Bob*	Overlap
OZ-1730 No-mod / Mod	1,730	1,730	849
OZ-17,294 No-mod / Mod	17,294	17,294	8,536
OZ-172,938 No-mod / Mod	172,938	172,938	86,476
OZ-1,729,379 No-mod / Mod	1,729,379	1,729,379	864,231
NC	481,315	480,701	333,403

pre-processing to ensure that each actual voter is given a unique voter ID. We assigned voters who we believe had several IDs a new unique ID, and in cases where an ID was shared between more than one voter we also assigned each voter a new unique ID. We extracted four attributes (first name, surname, city, and zipcode) of 629,362 voters, such that 314,644 were represented by a single record and 314,718 by two or more records (up-to 6), where all duplicate records contain errors and variations. We split this dataset into two containing 481,315 and 480,701 records for *Alice* and *Bob*, respectively. Because voter registration numbers (voter IDs) identify unique voters we can calculate the linkage quality.

Table 3 provides an overview of the datasets we used. We implemented all solutions presented in Section 5 using Python (version 2.7.3). All tests were run on a computer server with two 64-bit Intel Xeon (2.4 GHz) CPUs, 128 GBytes of main memory and running Ubuntu 12.04. The programs and (the small) test datasets are available from the authors.

6.1 Private Blocking Techniques

We compared and evaluated the scalability, quality, and privacy of the six private blocking approaches described in Section 5, which are labeled as **SNC-2P** [54], **SNC-3PSim** [52], **SNC-3PSize** [52], **HCLUST** [38], **k-NN** [32], and **HLSH** [18]. We used parameter settings for these techniques in a similar range as used by the authors of these techniques.

For **k-NN**, k is set to 3 and the minimum similarity threshold is set as $s_t = 0.6$. In the **HLSH** method, the number of iterations is set to $\mu = 40$, the number of hash functions is $k = 30$, the length of Bloom filters is $l = 1,000$ bits, and the number of bits to be sampled from the Bloom filters at each iteration is $\phi = 45$. In the **HCLUST** method, the number of clusters is set to 1/10 of the number of records in the dataset, the differential privacy parameter $\epsilon = 0.3$, and the fake records tolerance parameter w_n is set as the number of records in the datasets to be linked. The parameters for the **SNC** approaches are set as minimum block size $k = 100$, minimum similarity threshold $s_t = 0.8$, and window size $w = 2$.

Figure 11 shows the scalability of blocking approaches to different sizes of the OZ datasets measured by total blocking time (averaged over the results of all parties over all variations of each dataset). As can be seen from the figure, the SNC-based approaches

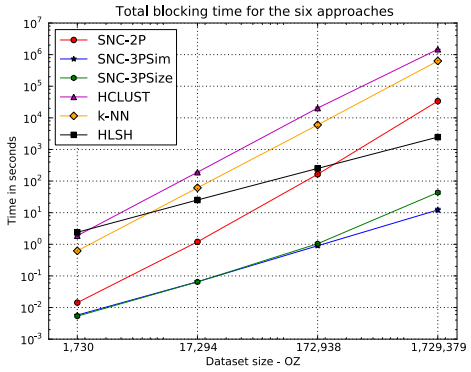

Figure 11: A comparison of scalability (evaluated by total blocking time) of the six private blocking approaches on the OZ datasets.

(**SNC-2P**, **SNC-3PSim** and **SNC-3PSize**) require less time than other approaches and are scalable to large databases. **k-NN** and **HCLUST** take significantly higher blocking time than **HLSH** and the **SNC**-based approaches.

The efficiency of blocking (scalability) measured by RR and the effectiveness of blocking (quality) measured by PC of the six private blocking approaches are compared on the OZ-172,938 Mod and NC datasets in Figure 12. **SNC-2P** achieves the highest PC at the cost of some reduction in RR, while the other approaches comparatively have lower PC with RR being almost 1.0. **HLSH** performs better by achieving high values for both RR and PC. The scalability and quality values for the blocking approaches on the NC dataset are mapped into a RR and PC plot, shown in Figure 13, to compare the trade-off of scalability and quality of blocking.

Finally, the privacy protection of the solutions are evaluated using the disclosure risk measures presented in Section 4.1. Due to time and memory constraints, we use the original dataset as the global dataset ($\mathbf{G} \equiv \mathbf{D}$) for privacy evaluation under the worst case assumption. The sizes of blocks generated by the six private blocking approaches are compared on the OZ-172,938 Mod and NC datasets in a box-and-whisker plot in Figure 14. The **SNC**-based approaches and **HCLUST** have lower variances between the block sizes which make a frequency attack using block sizes more difficult. The **HLSH** approach generates overlapping blocks of smaller sizes and the variance between block sizes is comparatively very high. It is important to note that if the third party (in three-party solutions) does not have any information regarding the parameters used and/or if it does not collude with any of the database owners, then trying to mount a frequency attack even with variant block sizes is non-trivial.

Figure 15 shows the distributions of probability of suspicion (P_s) values (similar to the examples illustrated in Figures 5, 6, and 7) in the NC dataset blocked by the six private blocking approaches. The median of the distribution (which is used to calculate

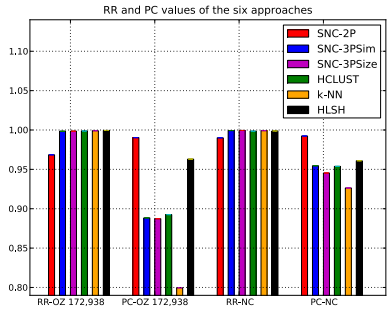

Figure 12: A comparison of reduction ratio (RR) and pairs completeness (PC) of the six private blocking approaches on the OZ-172,938 Mod and NC datasets.

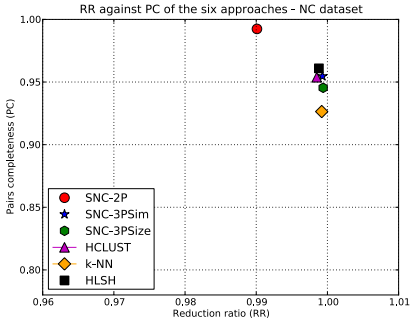

Figure 13: Reduction ration (RR) against pairs completeness (PC) of the six private blocking approaches on the NC dataset. Better solutions are closer to the upper right corner.

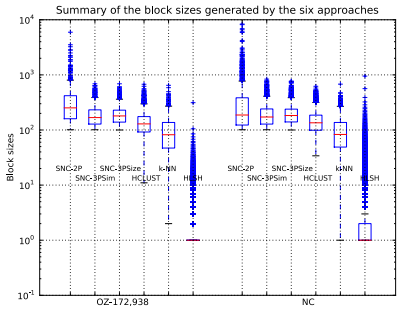

Figure 14: A comparison of block sizes generated by the six private blocking approaches on the OZ-172,938 Mod and NC datasets.

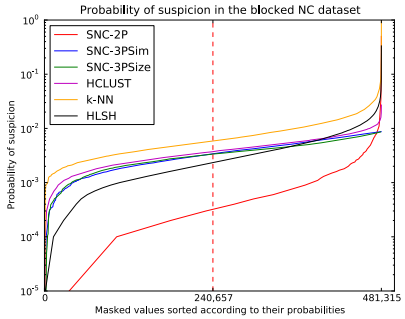

Figure 15: Distributions of probability of suspicion values of the blocked datasets generated by the six blocking approaches on the NC dataset.

DR_{Med}) is marked by a vertical dotted line in the figures. **SNC-2P** generates the lowest probability of suspicion curve on both datasets. However, its maximum P_s goes higher compared to **SNC-3PSim**, **SNC-3PSize**, and **HCLUST** approaches.

The trade-off between privacy (measured by DR_{Max}, DR_{Mean}, DR_{Med}, and RIG) and quality (measured by PC) of private blocking solutions is illustrated in Figure 16 for all six approaches on the OZ-172,938 Mod and NC datasets. **SNC-2P** provides the highest PC with reasonably lower DR. Next follow the **SNC-3PSim**, **SNC-3PSize**, and **HCLUST** approaches, which perform better compared to the **k-NN** and **HLSH** ones by achieving higher PC with lower values for DR measures.

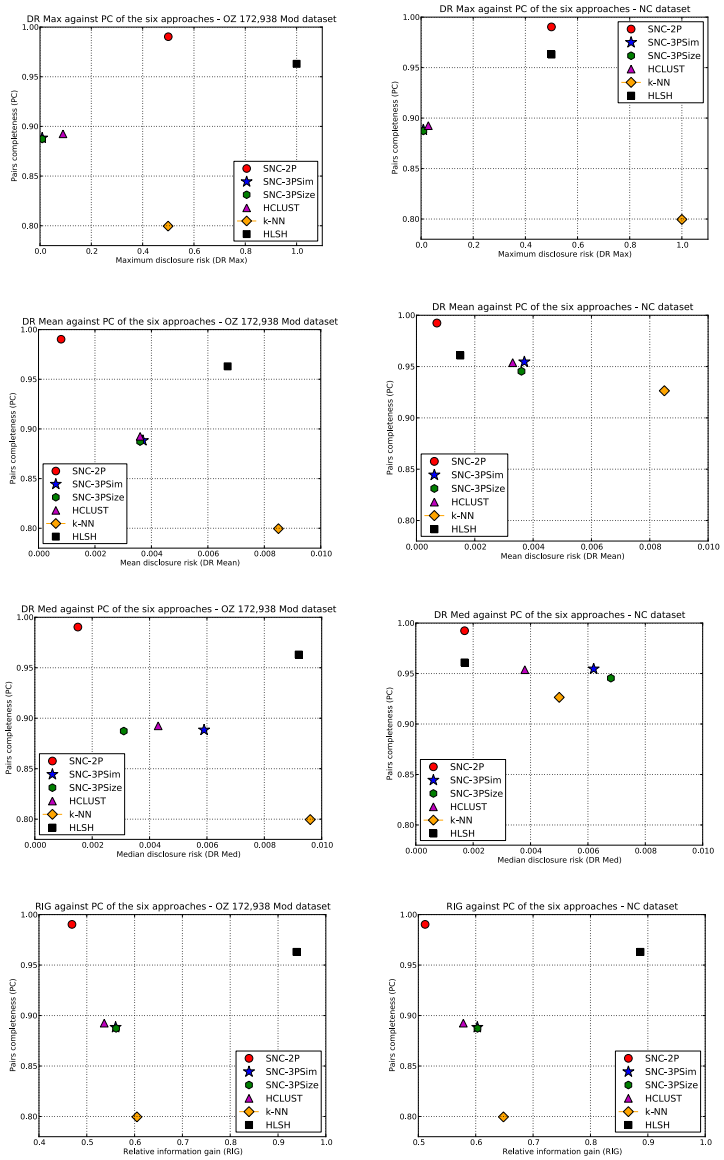

Figure 16: A comparison of disclosure risk measures (DR_{Max}, DR_{Mean}, DR_{Med}, RIG) against pairs completeness (PC) of the six private blocking approaches on the OZ-172,938 Mod (left column) and NC (right column) datasets. The best solutions are the ones closest to the upper left corner.

Table 4: Bloom filter parametrization for **CLK**, **RBF**, and **CLKRBF**.

	First name	Last name	City	Postcode
CLK hash functions (k)	30	30	30	30
CLK length (l)	1,000	1,000	1,000	1,000
RBF hash functions (k)	30	30	30	30
Agreement weight	2.5834	2.8908	1.2415	2.0852
Disagreement weight	−1.3757	−1.1752	−0.7708	−0.3543
Range (weight)	3.9591 (32%)	4.0660 (33%)	2.0123 (16%)	2.4395 (19%)
Average q-grams (g)	5.0762	5.3255	7.7592	3.9861
Dynamic BF length [19]	223	233	334	173
RBF length [19] (l)	668	689	334	397
Weight	32%	33%	16%	19%
CLKRBF hash functions (k)	29	30	15	17
CLKRBF length (l)	1,000	1,000	1,000	1,000

6.2 Private Comparison and Classification Techniques

In this section, we empirically evaluate the private comparison and classification solutions presented in Section 5.2, which are labeled as **2P-Bin** [53], **2P-BF CLK** [48, 51], **2P-BF RBF** [18, 51], and **2P-BF CLKRBF** [51]. For the **2P-Bin** [53] solution, the number of bins used is in the range of $k = [4, 6, 8, 10, 12]$ and the minimum similarity threshold is set to $s_t = 0.8$. As in previous work [47, 48, 51], the default parameters for the **2P-BF** [51]-based solutions are set to the number of hash functions $k = 30$, length of Bloom filters $l = 1,000$, $q = 2$, and the minimum similarity threshold $s_t = 0.8$. Weights, l of each attribute for the **RBF** method, and k of each attribute for the **CLKRBF** method on the NC dataset are given in Table 4.

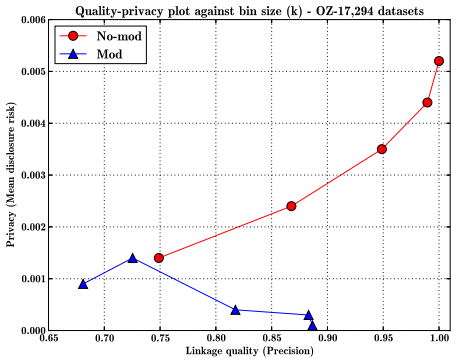

Figure 17: Disclosure risk and linkage quality plot of **2P-Bin** for different number of bins ($k = [4, 6, 8, 10, 12]$) used on the OZ-17,294 No-mod and Mod datasets.

Figure 17 shows how linkage quality increases with the number of bins (k) while disclosure risk increases (i.e., privacy reduces) in the **2P-Bin** solution. Disclosure risk values in the modified datasets are lower than the values in the non-modified datasets,

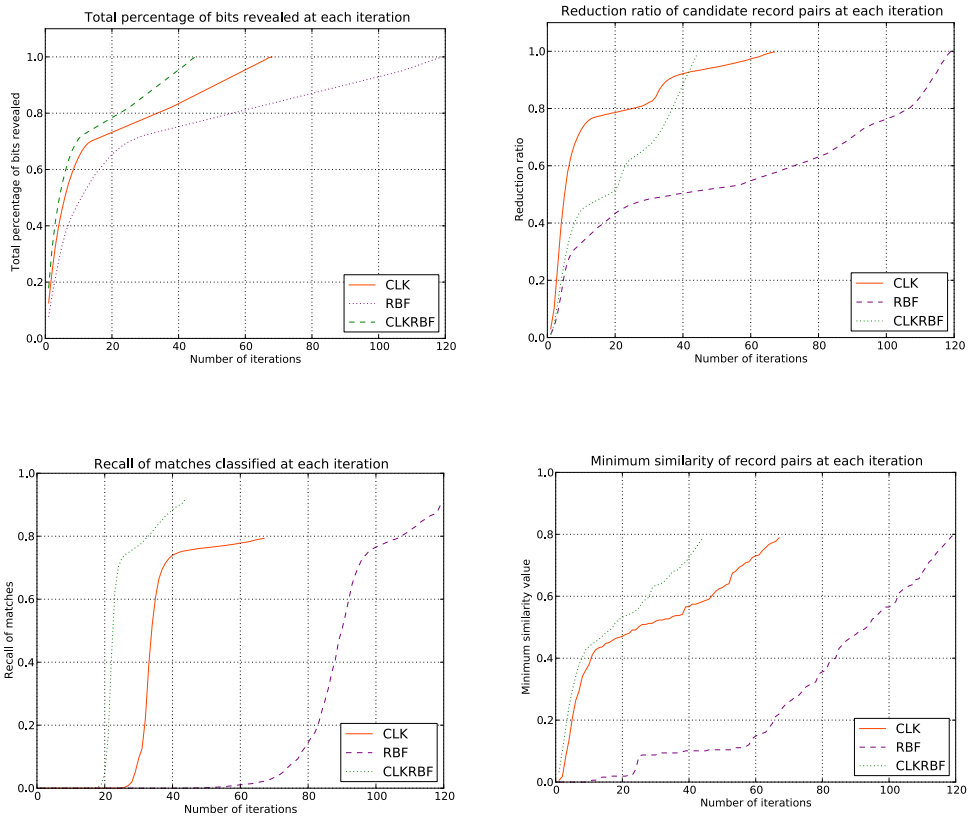

Figure 18: The percentage of bits revealed, reduction ratio of compared record pairs, recall ratio of matches, and minimum similarity value of unclassified record pairs at each iteration for **CLK**, **RBF**, and **CLKRBF** encodings in the **2P-BF** solution on the NC dataset.

because the number of global matches becomes smaller with modified (by data errors and variations) values. Interestingly, in the modified dataset the mean disclosure risk decreases with k. This is because with modified datasets, the number of global matches n_g in $\mathbf{G}^M$ with the same bin values as the bin values in $\mathbf{D}^M$ for the linkage attributes becomes zero with more bins, and thus all the N global values in $\mathbf{G}^M$ can be considered as possible matches, which decreases the disclosure risk. Small variations in the linkage attribute values would make a frequency linkage attack more difficult.

We then compared the Bloom filter-based approaches with the three encodings. As Figure 18 illustrates, the **RBF** encoding requires more iterations to converge but achieves a higher recall of matches compared to the **CLK** method that completes the task in a smaller number of iterations. The hybrid **CLKRBF** method achieves a higher recall in a smaller number of iterations. The minimum similarity value of record pairs that

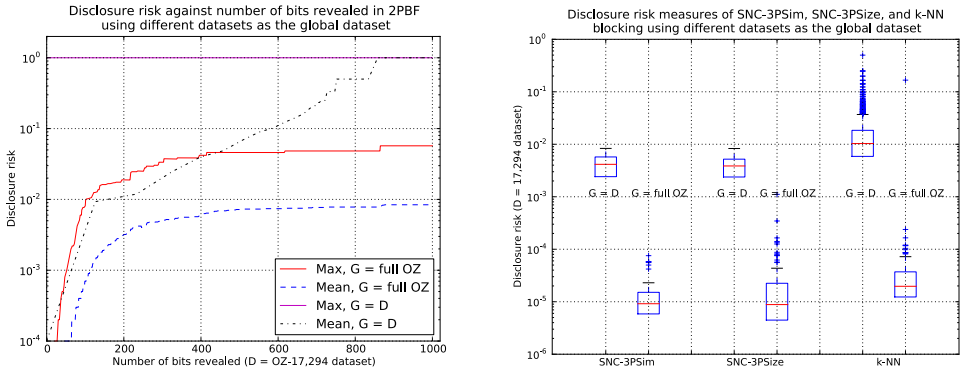

Figure 19: A comparison of disclosure risk values of the **2P-BF** solution against number of bits revealed and disclosure risk values of private blocking solutions on the $\mathbf{D}$ = OZ-17,294 dataset using $\mathbf{G} \equiv \mathbf{D}$ and $\mathbf{G}$ = full OZ database.

remain unclassified shows that the **CLK** and **CLKRBF** encoding methods have a minimum similarity of 0.5 (i.e., non-matches with less than 0.5 similarity are removed) when half of the iterations are completed, while the **RBF** encoding requires three quarter of iterations to classify pairs so that the remaining pairs have a minimum 0.5 similarity value. Hence, the **CLKRBF** encoding method outperforms the other two encodings in the **2P-BF** solution by achieving higher linkage quality, and better privacy in terms of bit distribution and pruning of non-matches.

The experiments described above assumed the worst case setting of global dataset. Since we used the original dataset as the global dataset in this worst case ($\mathbf{G} \equiv \mathbf{D}$), the number of global values n_g in $\mathbf{G}^M$ that match a certain masked value in $\mathbf{D}^M$ is very small, which results in high disclosure risk values. Ideally, a global dataset would not necessarily be equivalent to the original dataset and would have many combinations of different attribute values resulting in lower disclosure risk values, as was discussed in Section 5. Testing the privacy of the **2P-BF** technique and several private blocking techniques such as **SNC-3PSim**, **SNC-3PSize**, and **k-NN** on the OZ-17,294 dataset using a global dataset that is the full Australian telephone database (containing around 6.9 million records) provides much lower (2.5 magnitudes) disclosure risk results compared to the results in the worst case setting of $\mathbf{G} \equiv \mathbf{D}$, as shown in Figure 19.

Figure 20 shows the scalability to different sizes of datasets (calculated by total linkage time) of the four private comparison and classification techniques (**2P-Bin**, **2P-BF CLK**, **2P-BF RBF**, **2P-BF CLKRBF**) on the OZ datasets. Bin size is used as $k = 6$ for the binning-based approach (**2P-Bin**) and all four attributes in the OZ datasets are used as linkage attributes for all four techniques. The **2P-Bin** approach requires less linkage time and is more efficient than the **2P-BF**-based approaches (**2P-BF CLK**, **2P-BF RBF**, **2P-BF CLKRBF**). However, the disclosure risk is higher and the linkage quality is lower for the **2P-Bin** approach compared to the **2P-BF**-based approaches,

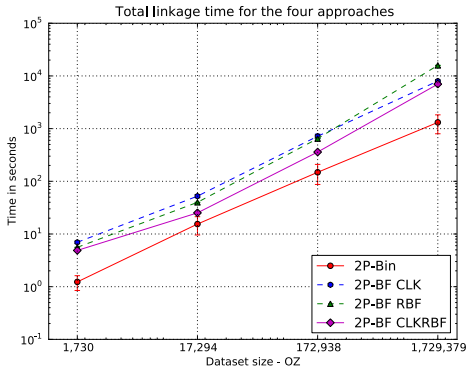

Figure 20: A comparison of scalability (measured by linkage time) of the four private comparison and classification techniques on the OZ datasets.

as will be compared in Figure 21. All three variations of the **2P-BF**-based approaches require similar linkage time. The **CLKRBF** encoding method is faster than the **CLK** and **RBF** encoding methods as it requires a smaller number of iterations to converge compared to the other two encoding methods (which we discussed in Figure 18).

A comparison of disclosure risk measures (DR_{Max}, DR_{Mark}, DR_{Mean}, DR_{Med}) against linkage quality (calculated by F-measure) of the four private comparison and classification techniques on the OZ-17,294 Mod dataset is given in Figure 21. The **2P-Bin** solution leads to higher disclosure risk values (i.e., lower privacy) and lower linkage quality than the **2P-BF**-based approaches. In **2P-BF** approaches, as can be seen from Figures 20 and 21, the **CLKRBF** encoding method performs better by achieving high F-measure, providing low disclosure risk, and requiring less linkage time than the other two encoding methods.

6.3 Discussion

The presented evaluation of several private blocking and private comparison and classification solutions using the proposed evaluation framework provides a comprehensive view of the performances of these solutions with regard to the three main properties of PPRL: scalability, linkage quality, and privacy.

The empirical results of private blocking solutions on the NC dataset and private comparison and classification solutions on the OZ-17,294 Mod dataset are summarized in Tables 5 and 6, respectively, in terms of the three properties: scalability, linkage quality, and privacy. We calculated overall scores (using Equation 9 with $\alpha = 0.33$ and $\beta = 0.33$, i.e., equal weights) to compare the viability of PPRL solutions with respect to all three properties. Scores with different weights would provide a ranking of solutions in the preferred context depending upon application and/ or user requirements.

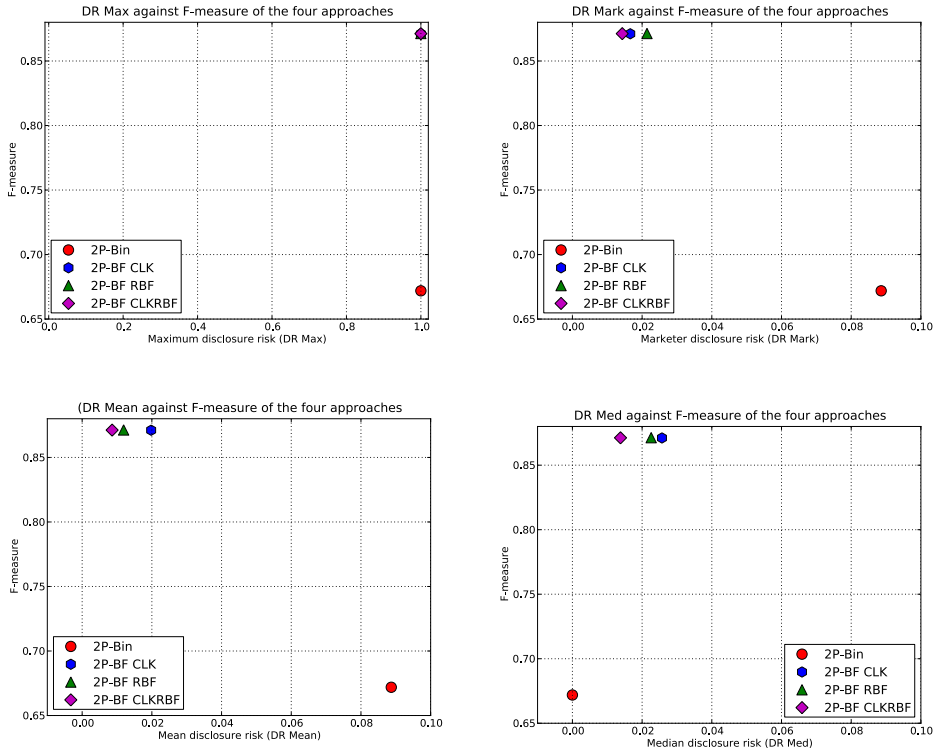

Figure 21: A comparison of disclosure risk measures (DR_{Max}, DR_{Mark}, DR_{Mean}, DR_{Med}) against linkage quality (measured by F-measure) of the four private comparison and classification approaches on the OZ-17,294 Mod dataset. The best solutions are the ones closest to the upper left corner.

However, determining appropriate weights for each aspect is a cumbersome task that requires domain and application knowledge.

Different scores are calculated with different combinations of measures for the three properties, as presented in Tables 5 and 6. For private blocking solutions we calculated the following four scores: score 1 is an average of RR, PC, and DR_{Max}, score 2 is an average of RR, PC, and DR_{Mean}, score 3 is an average of RR, PC, and RIG, and score 4 is an average of time, PC, and DR_{Mean}. The scores calculated for the private comparison and classification solutions are: score 1 is an average of time, F-measure, and DR_{Max}, score 2 is an average of time, F-measure, and DR_{Mark}, and score 3 is an average of time, F-measure, and DR_{Mean}.

The comparison results of the six private blocking solutions presented in Table 5 show that **SNC-2P** outperforms the other solutions in terms of all the measures except DR_{Max} (and thus except score 1). **HLSH** is faster and achieves higher RR and PC

Table 5: Comparison of the six private blocking approaches on the NC dataset. Best values in each row are shown in bold font. Four different scores are calculated as averages of the measures for the three properties. Measures marked with (+) have a positive impact on the overall score and measures with (-) have a negative impact.

	SNC-2P	SNC-3PSim	SNC-3PSize	HCLUST	k-NN	HLSH
Time (-)	1044.02	**2.6439**	4.5502	95225.82	47075.76	1098.73
normalized	0.0109	**0.0000**	0.0001	1.0000	0.4943	0.0115
RR (+)	0.9901	0.9993	**0.9994**	0.9985	0.9992	0.9988
PC (+)	**0.9924**	0.9546	0.9454	0.9538	0.9264	0.9609
DR_{Max} (-)	0.4999	**0.0087**	**0.0087**	0.0278	1.0000	0.4999
DR_{Mean} (-)	**0.0007**	0.0037	0.0036	0.0033	0.0085	0.0015
RIG (-)	**0.5118**	0.6028	0.6031	0.5784	0.6483	0.8870
Score 1: RR, PC, DR_{Max}	0.8275	**0.9817**	0.9787	0.9748	0.6419	0.8199
Score 2: RR, PC, DR_{Mean}	**0.9939**	0.9834	0.9804	0.9830	0.9724	0.9861
Score 3: RR, PC, RIG	**0.8236**	0.7837	0.7806	0.7913	0.7591	0.6909
Score 4: Time, PC, DR_{Mean}	**0.9936**	0.9836	0.9806	0.6502	0.8079	0.9826

Table 6: Comparison of the four private comparison and classification approaches on the OZ-17,294 Mod dataset. Best values in each row are shown in bold font. Three different scores are calculated as averages of the measures for the three properties. Measures marked with (+) have a positive impact on the overall score and measures with (-) have a negative impact.

	2P-Bin	2P-BF CLK	2P-BF RBF	2P-BF CLKRBF
Time (-)	11.2641	48.6865	39.8932	25.1866
normalized	**0.0000**	1.0000	0.7650	0.3720
Precision (+)	**1.0000**	0.9995	0.9997	0.9997
Recall (+)	0.5059	0.7719	**0.7721**	0.7720
F-measure / F (+)	0.6719	0.8711	**0.8713**	0.8712
DR Max (-)	1.0000	1.0000	1.0000	1.0000
DR Mark (-)	0.2886	0.0166	0.0214	**0.0143**
DR Mean (-)	0.2887	0.0198	0.0119	**0.0086**
Score 1: Time, F, DR_{Max}	**0.5573**	0.2904	0.3687	0.4997
Score 2: Time, F, DR_{Mark}	0.7944	0.6181	0.6950	**0.8283**
Score 3: Time, F, DR_{Mean}	0.7944	0.6171	0.6981	**0.8302**

compared to the other four approaches, however the DR and RIG measures are higher (i.e., lower privacy). **SNC-3PSim** and **SNC-3PSize** are faster as well with lower values for DR and RIG and achieve moderately higher RR and PC values. The **k-NN** and **HCLUST** approaches are slower though the other aspects provide moderate results.

Among the four private comparison and classification solutions compared in Table 6, the **2P-BF**-based approaches provide higher linkage quality results than the binning-based approach (**2P-Bin**), while the DR measures are also lower (which means privacy is higher compared to the **2P-Bin** approach). However, the **2P-Bin** solution is efficient and requires much shorter runtime compared to others. The **2P-BF** with the **CLKRBF** encoding method outperforms in terms of overall scores.

Table 7: Blocking combined with the **2P-BF** [51] private comparison and classification solution on the OZ-1,730 Mod dataset. Best values in each row are shown in bold font.

	No blocking + 2P-BF	Phonetic + 2P-BF	SNC-2P + 2P-BF
Time (seconds)	173.92	**6.6233**	15.1179
Precision	0.9208	**1.0000**	0.9972
Recall	**1.0000**	0.7680	0.9504
F-measure	0.9588	0.8688	**0.9732**
DR_{Mean}	**0.0010**	0.9909	0.0217
DR_{Mark}	**0.0000**	0.9908	0.0046

Finally, we have studied how a private blocking solution (we chose the **SNC-2P** as it provides higher scores compared to others—Table 5) combined with the **2P-BF** **CLKRBF** private comparison and classification solution determines the three properties of scalability, quality, and privacy. We combined the **2P-BF** solution with no blocking, Soundex [3]-based phonetic blocking (a standard blocking approach that has been used in non-PPRL, where records with the same phonetic encodings for the BKVs are grouped into the same block), and the SNC-based two-party private blocking (SNC-2P) solution. Table 7 presents the total time required for blocking and linkage, linkage quality results, and the DR measures in the worst case setting ($\mathbf{G} \equiv \mathbf{D}$) of the **2P-BF** solution with these three blocking scenarios. As the results show, when no blocking is applied the DR values are very low. However, no blocking requires significantly higher linkage time compared to when a blocking technique is applied. Phonetic-based blocking is faster than the **SNC-2P** private blocking, though privacy and linkage quality results are comparatively better with the **SNC-2P** approach.

Figure 22 maps score 2 of the six private blocking solutions on the NC dataset, and score 3 of the four private comparison and classification solutions on the OZ-17,294 Mod dataset into three-dimensional (3D) plots. Such a graphical representation of evaluation results allows us to analyze where a solution is placed in terms of the three properties of privacy, linkage quality, and scalability, and to compare different solutions. These 3D plots are better suited for interactive exploration or visualization than static visualization in a printed form.

7 Conclusion and Future Work

In this paper, we have presented a comprehensive evaluation framework for privacy-preserving record linkage (PPRL) solutions that enables assessment and comparison of different solutions in terms of the three main properties of PPRL, which are scalability, linkage quality, and privacy. Scalability and quality of PPRL solutions can be assessed using the standard measures that have been used in the literature. However, numerical measures to quantify the privacy guarantees provided by a solution need to be defined.

We have defined five different disclosure risk measures that can be used to measure the privacy of PPRL solutions by simulating linkage attacks on those solutions using

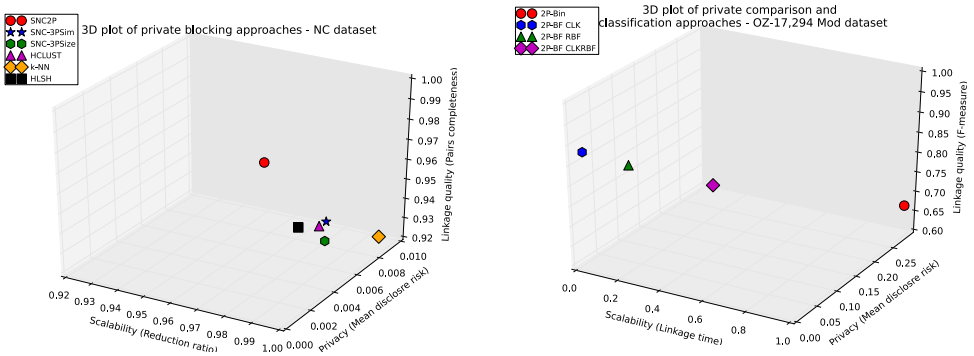

Figure 22: 3D plots showing the comparison of score 2 (RR, PC, and DR_{Mean}) of the six private blocking approaches (left), and score 3 (time, F-measure, and DR_{Mean}) of the four private comparison and classification approaches (right). The best solutions are the ones closest to the front upper right corner.

an external global dataset. We used the framework to experimentally evaluate six private blocking, and four private comparison and classification solutions using real-world databases. The results validate that our framework allows extensive evaluation, analysis, and comparison of different PPRL solutions with respect to all three properties of PPRL.

Future work includes extending the framework to address the problem of PPRL of multiple sources and to consider different adversary models such as the covert model [1] or accountable computing [31]. Scoring the solutions with appropriate weights for different measures is an important problem to be solved. Additional work is required on large scale empirical evaluation [11] on other real datasets or realistic synthetic datasets generated using our GeCo (personal data Generator and Corruptor) tool [10, 50]. Investigating efficient and interactive linkage attacks, and approximation with frequency error bounds would be another direction for future research.

References

[1] Aumann, Y. and Lindell, Y. (2010). Security against covert adversaries: Efficient protocols for realistic adversaries. *Journal of Cryptology*, 23(2): 281–343.

[2] Bhattacharya, I. and Getoor, L. (2007). Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data, 2007*, 1(1).

[3] Christen, P. (2006). A comparison of personal name matching: Techniques and practical issues. In *IEEE ICDM Workshop on Mining Complex Data*. Hong Kong. 290–294.

[4] — (2006). Privacy-preserving data linkage and geocoding: Current approaches and research directions. In *IEEE ICDM Workshop on Privacy Aspects of Data Mining*. Hong Kong. 497–501.

[5] — (2012). *Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-Centric Systems and Applications. Berlin: Springer.

[6] — (2012). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Transactions on Knowledge and Data Engineering*, 24(9): 1537–1555.

[7] — (2014). Preparation of a real voter data set for record linkage and duplicate detection research. Technical report, Research School of Computer Science, Australian National University, Canberra.

[8] Christen, P. and Goiser, K. (2007). Quality and complexity measures for data linkage and deduplication. In *Quality Measures in Data Mining*, vol. 43 of *Studies in Computational Intelligence*. Springer. 127–151.

[9] Christen, P. and Pudjijono, A. (2009). Accurate synthetic generation of realistic personal information. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2009)*, vol. 5476 of LNAI. Springer. 507–514.

[10] Christen, P. and Vatsalan, D. (2013). Flexible and extensible generation and corruption of personal data. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM '13)*. ACM Press. 1165–1168.

[11] Christen, P., Vatsalan, D., and Verykios, V. S. (2014). Challenges for privacy preservation in data integration. *ACM Journal of Data and Information Quality.* Accepted.

[12] Clifton, C., Kantarcioglu, M., Doan, A., Schadow, G., Vaidya, J., Elmagarmid, A., and Suciu, D. (2004). Privacy-preserving data integration and sharing. In *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. ACM Press. 19–26.

[13] Dalenius, T. (1986). Finding a needle in a haystack-or identifying anonymous census record. *Journal of Official Statistics*, 2(3): 329–336.

[14] Dankar, F. and El Emam, K. (2010). A method for evaluating marketer re-identification risk. In *Proceedings of the 2010 EDBT/ICDT Workshops*, 28. ACM Press.

[15] Draisbach, U., Naumann, F., Szott, S., and Wonneberg, O. (2012). Adaptive windows for duplicate detection. In *Proceedings of the 28th IEEE International Conference on Data Engineering (ICDE 2012)*. IEEE. 1073–1083.

[16] Duncan, G. T., Elliot, M. J., and Salazar-González, J.-J. (2011). *Statistical Confidentiality: Principles and Practice*. New York: Springer.

[17] Duncan, G. T., Keller-McNulty, S. A., and Stokes, S. L. (2001). Disclosure risk vs. data utility: The RU confidentiality map. Technical Report LA-UR-6428, Los Alamos National Laboratory.

[18] Durham, E. (2012). A Framework for Accurate, Efficient Private Record Linkage. Ph.D. thesis, Faculty of the Graduate School of Vanderbilt University, Nashville, Tennessee, USA.

[19] Durham, E. A., Toth, C., Kuzu, M., Kantarcioglu, M., Xue, Y., and Malin, B. (2013). Composite Bloom filters for secure record linkage. *IEEE Transactions on Knowledge and Data Engineering*, 1(99): 1.

[20] Dwork, C. (2008). Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*. Springer. 1–19.

[21] Elmagarmid, A., Ipeirotis, P., and Verykios, V. S. (2007). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1): 1–16.

[22] Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Society*, 64(328): 1183–1210.

[23] Ferro, J., Singh, L., and Sherr, M. (2013). Identifying individual vulnerability based on public data. In *Proceedings of the 11th Annual International Conference on Privacy, Security and Trust (PST 2013)*. IEEE. 119–126.

[24] Fienberg, S. E. (2005). Confidentiality and disclosure limitation. *Encyclopedia of Social Measurement*, 1: 463–69.

[25] Ganta, S. R., Kasiviswanathan, S. P., and Smith, A. (2008). Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM Press. 265–273.

[26] Gionis, A., Mazza, A., and Tassa, T. (2008). k-anonymization revisited. In *Proceedings of the 24th International Conference on Data Engineering (ICDE 2008)*. IEEE. 744–753.

[27] Hall, R. and Fienberg, S. (2010). Privacy-preserving record linkage. In *Proceedings of the 2010 International Conference on Privacy in Statistical Databases (PSD '10)*, vol. 6344 of *LNCS*. Springer. 269–283.

[28] Hernandez, M. A. and Stolfo, S. J. (1998). Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1): 9–37.

[29] Herschel, M., Naumann, F., Szott, S., and Taubert, M. (2012). Scalable iterative graph duplicate detection. *IEEE Transactions on Knowledge and Data Engineering*, 24(11): 2094–2108.

[30] Herzog, T., Scheuren, F., and Winkler, W. (2007). *Data Quality and Record Linkage Techniques*. New York: Springer.

[31] Jiang, W., Clifton, C., and Kantarcıoğlu, M. (2008). Transforming semi-honest protocols to ensure accountability. *Data and Knowledge Engineering*, 65(1): 57–74.

[32] Karakasidis, A. and Verykios, V. S. (2012). Reference table based k-anonymous private blocking. In *Proceedings of the ACM Symposium on Applied Computing (SAC 2012)*. ACM Press. 859–864.

[33] Karakasidis, A., Verykios, V. S., and Christen, P. (2012). Fake injection strategies for private phonetic matching. In *Proceedings of the 6^{th} International Conference and 4^{th} International Conference on Data Privacy Management and Autonomous Spotaneous Security (DPM '11)*, vol. 7122 of *LNCS*. Springer. 9–24.

[34] Kargupta, H., Datta, S., Wang, Q., and Sivakumar, K. (2005). Random-data perturbation techniques and privacy-preserving data mining. *Knowledge and Information Systems*, 7(4): 387–414.

[35] Kifer, D. (2009). Attacks on privacy and deFinetti's theorem. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD '09)*. ACM. 127–138.

[36] Kum, H.-C., Krishnamurthy, A., Machanavajjhala, A., Reiter, M. K., and Ahalt, S. (2014). Privacy preserving interactive record linkage (PPIRL). *Journal of the American Medical Informatics Association*, 21(2): 212–220.

[37] Kuzu, M., Kantarcioglu, M., Durham, E., and Malin, B. (2011). A constraint satisfaction cryptanalysis of Bloom filters in private record linkage. In *Proceedings of the 11^{th} International Symposium on Privacy Enhancing Technologies (PETS 2011)*, vol. 6794 of *LNCS*. Springer. 226–245.

[38] Kuzu, M., Kantarcioglu, M., Inan, A., Bertino, E., Durham, E., and Malin, B. (2013). Efficient privacy-aware record integration. In *Proceedings of the 16^{th} International Conference on Extending Database Technology (EDBT '13)*. ACM Press. 167–178.

[39] Lindell, Y. and Pinkas, B. (2009). Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1): 5.

[40] Liu, H., Wang, H., and Chen, Y. (2010). Ensuring data storage security against frequency-based attacks in wireless networks. In *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '10)*, vol. 6131 of *LNCS*. Springer. 201–215.

[41] Navarro-Arribas, G. and Torra, V. (2012). Information fusion in data privacy: A survey. *Information Fusion*, 13(4): 235–244.

[42] Pang, C., Gu, L., Hansen, D., and Maeder, A. (2009). Privacy-preserving fuzzy matching using a public reference table. In *Intelligent Patient Management*, vol. 189 of *Studies in Computational Intelligence*. Springer. 71–89.

[43] Papadimitriou, C. (2003). *Computational Complexity*. John Wiley and Sons Ltd.

[44] Raghavan, V., Bollmann, P., and Jung, G. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems*, 7(3): 205–229.

[45] Ramachandran, A., Singh, L., Porter, E., and Nagle, F. (2012). Exploring re-identification risks in public domains. In *Proceedings of the 10th Annual International Conference on Privacy, Security and Trust (PST 2012)*. IEEE. 35–42.

[46] Reiter, M. and Rubin, A. (1998). Crowds: Anonymity for web transactions. *ACM Transactions on Information System Security*, 1(1): 66–92.

[47] Schnell, R., Bachteler, T., and Reiher, J. (2009). Privacy-preserving record linkage using Bloom filters. *BMC Medical Informatics and Decision Making*, 9(1).

[48] — (2011). A novel error-tolerant anonymous linking code. *German Record Linkage Center, Working Paper Series No. WP-GRLC-2011-02*.

[49] Sweeney, L. (2001). Computational Disclosure Control: A Primer on Data Privacy Protection. Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, Massachusetts, USA.

[50] Tran, K.-N., Vatsalan, D., and Christen, P. (2013). GeCo: An online personal data generator and corruptor. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)*. ACM Press. 2473–2476.

[51] Vatsalan, D. and Christen, P. (2012). An iterative two-party protocol for scalable privacy-preserving record linkage. In *Proceedings of the 10th Australasian Data Mining Conference (AusDM '12)*, vol. 134. Darlinghurst, Australia: Australian Computer Society. 127–138.

[52] — (2013). Sorted nearest neighborhood clustering for efficient private blocking. In *Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2013)*, vol. 7819 of *LNAI*. Springer. 341–352.

[53] Vatsalan, D., Christen, P., and Verykios, V. S. (2011). An efficient two-party protocol for approximate matching in private record linkage. In *Proceedings of the 9th Australasian Data Mining Conference (AusDM '11)*, vol. 121. Darlinghurst, Australia: Australian Computer Society. 125–136.

[54] — (2013). Efficient two-party private blocking based on sorted nearest neighborhood clustering. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)*. ACM Press. 1949–1958.

[55] — (2013). A taxonomy of privacy-preserving record linkage techniques. *Journal of Information Systems (JIS)*, 38(6): 946–969.

[56] Verykios, V. S., Karakasidis, A., and Mitrogiannis, V. (2009). Privacy preserving record linkage approaches. *International Journal of Data Mining, Modelling and Management*, 1(2): 206–221.

[57] Wilson, R. L. and Rosen, P. A. (2003). Protecting data through perturbation techniques: The impact on knowledge discovery in databases. *Journal of Database Management*, 14(2): 14–26.

[58] Wong, R., Fu, A., Wang, K., and Pei, J. (2007). Minimality attack in privacy preserving data publishing. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB '07)*. VLDB Endowment. 543–554.

[59] Yakout, M., Atallah, M., and Elmagarmid, A. (2009). Efficient private record linkage. In *Proceedings of the 25th International Conference on Data Engineering (ICDE 2009)*. IEEE. 1283–1286.