PRIMAT: A Toolbox for Fast Privacy-preserving Matching

Martin Franke
Database Group
Leipzig University &
ScaDS Dresden/Leipzig
franke@informatik.uni-leipzig.de

Ziad Sehili
Database Group
Leipzig University &
ScaDS Dresden/Leipzig
sehili@informatik.uni-leipzig.de

Erhard Rahm
Database Group
Leipzig University &
ScaDS Dresden/Leipzig
rahm@informatik.uni-leipzig.de

ABSTRACT

Privacy-preserving record linkage (PPRL) is increasingly demanded in real-world applications, e.g., in the health-care domain, to combine person-related data for data analysis while preserving the privacy of individuals. However, the adoption of PPRL is hampered by the absence of easy-to-use and powerful PPRL tools covering the *entire* PPRL process. We therefore demonstrate PRIMAT, a flexible and scalable tool that enables the definition and application of tailored PPRL workflows as well as the comparative evaluation of different PPRL methods. We introduce the main requirements for PPRL tools and discuss previous tool efforts that do not fully meet the requirements and have not been applied in practice. By contrast, PRIMAT covers the whole PPRL life-cycle and improves applicability by providing various components for data owners and the central linkage to be executed by a trusted linkage unit.

PVLDB Reference Format:
Martin Franke, Ziad Sehili, Erhard Rahm. PRIMAT: A Toolbox for Fast Privacy-preserving Matching. *PVLDB*, 12(12): 1826-1829, 2019.
DOI: https://doi.org/10.14778/3352063.3352076

1. INTRODUCTION

The integration of person-related data, e.g., for customers or patients, is needed in many applications for improved data analysis. However, stricter legal data protection requirements increasingly ask for privacy-preserving data integration that does not reveal the identity of persons for whom data is combined and analyzed. These requirements are met by privacy-preserving record linkage (PPRL) techniques that encode identifying attributes, e.g., name and birth date, and often perform the linkage of encoded records in a separated, trusted environment. A large number of such PPRL methods has been proposed in the last years as surveyed in [20, 21]. Some of these approaches have also been applied, primarily in medical research studies to combine patient-related data from different hospitals or medical offices, e.g., to analyze diseases or treatments [6, 11, 18]. Despite the large number of proposed PPRL schemes, their practical use in real applications is still limited due to the absence of convenient tools and the high complexity to properly select and configure a suitable PPRL approach. In fact, the relative strengths and weaknesses of different PPRL approaches and configurations regarding privacy, effectiveness and efficiency are largely unknown and ask for more comparative evaluations. While there are some available PPRL implementations and prototypes (see Sec. 3) they focus on research aspects and do not provide enough functionality for use in practice. Previous PPRL applications in medicine are based on tailored solutions that are not usable in different applications. There is also proprietary PPRL software in use, e.g., in an Australian record linkage centre to support medical research projects [6, 12].

There is therefore a strong need for easy-to-use, powerful and open-source tools to facilitate the adoption of PPRL in real applications. We have thus started developing an open-source PPRL toolbox, named PRIMAT[1] (**Pri**vate **Ma**tching **T**oolbox). It includes our previously developed methods for fast and scalable PPRL based on the use of blocking and parallel matching of encoded records [3, 5, 7] as well as for post-processing to select the best matches [4]. PRIMAT focuses on practical usability and provides different linkage modes, protocols and components that support creating individual linkage workflows. Moreover, PRIMAT offers an evaluation framework to uniformly compare PPRL methods regarding their efficiency and effectiveness.

In the following, we first introduce the main requirements for PPRL tools such as PRIMAT (Sec. 2) and discuss related implementations (Sec. 3). Then, we provide an overview about PRIMAT and its components (Sec. 4), describe our demonstration scenario (Sec. 5) and finally conclude.

2. REQUIREMENTS FOR PPRL TOOLS

To achieve broad acceptance and operability the following requirements should be satisfied by a PPRL toolbox:

Tackle PPRL key challenges. PPRL has three key challenges that need to be carefully addressed. In particular, a high degree of *privacy* has to be ensured by providing state-of-the-art encoding techniques that reduce the risk for data breaches. Moreover, a high *linkage quality* must be achieved, i.e., the number of false and missing matches should be minimized. Finally, PPRL needs to be scalable to large data volumes and possibly many data owners where up to millions

[1]For source code see `https://github.com/gen-too/primat`

Proceedings of the VLDB Endowment, Vol. 12, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3352063.3352076

of records need to be linked. Hence, blocking or filtering methods [1] as well as parallel and distributed processing are beneficial to reduce the complexity of linkage and to speed up similarity computations.

Support of entire PPRL process. PPRL demands a multi-step process involving data owners and trusted third parties, e.g., the linkage unit, to ensure high efficiency and effectiveness. Data owners need to prepare their data to ensure comparability and encode their records to support privacy. Data preparation includes unification to resolve schema differences as well as cleaning procedures, e.g., to resolve data entry errors. The actual linkage also requires multiple steps, in particular blocking or filtering to avoid comparing every possible pair of records, similarity calculation and a classification step to decide whether a record pair is considered as a match. Consequently, supporting the entire linkage process is essential to bring PPRL into applications. In order to support various use cases, all components must be individually configurable to comply with the specific needs of each application scenario. Moreover, additional or future techniques should be easy to integrate.

Enabling of batch and incremental linkage. Typically, PPRL is executed in batch mode where *all* records are linked *at once*. However, in practice it is also desired to deploy PPRL in an incremental fashion such that new records can be added continuously without having to repeat linkage for previously known records. Adding individual records and querying their existence must be very fast [11].

Support for multiple data owners. While most previous PPRL approaches focus on only two data sources, it is essential to support multi-party approaches with two or more data owners. In this case, the matching records should not only be linked but also clustered so that all records in a cluster match with each other. Incremental linkage has to determine whether a new record belongs to an existing cluster or whether it represents a new cluster.

Ease of use. PPRL workflows should need minimal parameter tuning effort or deep knowledge of underlying techniques. Moreover, the effort for integrating PPRL in existing system environments should be minimized.

Evaluation. It is important for both practitioners and researchers to test and evaluate different PPRL methods and parameter settings to determine effective configurations and appropriately balance efficiency and privacy. For this purpose, tool support to generate and use realistic synthetic test data is highly beneficial to determine effective PPRL workflows. Furthermore, providing analysis and measurement facilities is helpful to evaluate different approaches and configurations under uniform conditions.

3. RELATED WORK

Many PPRL approaches have been proposed in the last years as summarized in [20, 21]. PPRL is strongly related to traditional record linkage, but focuses on privacy and thus raises new challenges. Recent approaches are mostly relying on Bloom-filter-based encoding techniques [18] and make use of a trusted linkage unit that centrally conducts the linkage of encoded records. A *Bloom filter* is a bit vector of fixed size where initially all bits are zero. A set of cryptographic hash functions is used to hash (map) features of a record, e.g., q-grams [1] drawn from record attributes, in the Bloom filter. More precisely, each hash function is applied on each record feature and returns an index within the bit vector. Finally, the bits at these index positions are set to one.

While some PPRL tools have been implemented, they do not meet all requirements collected in Sec. 2. MergeToolBox (MTB) [17] is a record linkage systems that has recently been extended to support current privacy-preserving techniques.[2] However, MTB mainly focuses on encoding techniques while providing only limited support of linkage methods. Mainzelliste [11] is a pseudonymization and identity management system designed for multi-site medical applications. While it has already been used in medical joint research projects in Germany, it does not support blocking nor current privacy-enhancing encoding techniques [21], thus lacking both scalability and sufficient privacy protection. SOEMPI [19] provides PPRL functionality and is also designed for medical applications. SOEMPI offers a variety of methods including current encoding and blocking techniques. However, it does not provide evaluation facilities nor it is optimized for incremental linkage. PRIVATEER [8] is a PPRL research prototype that lacks support of private blocking and does not provide incremental linkage. Finally, LSHDB [9] is another prototype, that offers parallel and distributed processing, privacy-preserving blocking as well as both batch and incremental linkage. However, incremental linkage is only supported with limits, e.g., there is no clustering mechanism for more than two records referring to the same entity. Besides, LSHDB does not provide any encoding, hardening or evaluation facilities. Additionally, all current PPRL implementations lack of any pre-processing support to weaken typical data quality problems. As a consequence, each listed implementation fails to cover the entire PPRL process. Moreover, no tool is actually used in real-world PPRL applications due to the aforementioned problems.

4. DESCRIPTION OF OUR TOOLBOX IMPLEMENTATION

In Fig. 1 the overall architecture of Primat is depicted. We differentiate between two main roles for participating in a PPRL process:

Database owners (DOs) manage sensitive data in form of person-related database records, e.g., patient records, that should be linked to the other DOs records in a privacy-preserving manner. Any database record consists of specific attributes and has an unique record identifier that is no entity identifier. We do not make any assumptions about the status of the databases to be linked regarding inconsistencies or their deduplication, i.e., if there is more than one record per entity in a single database.

Trusted third parties can have multiple responsibilities within a linkage process. For Primat, we assume that a trusted **linkage unit** (LU) performs the actual linkage of encoded records submitted by the DOs. Using a LU is beneficial since it requires less complex protocols and low communication costs. Secondly, we assume a **data analyst** that wants to combine the DOs data for analysis or research. Other third parties may be involved to coordinate

[2]See `https://CRAN.R-project.org/package=PPRL`

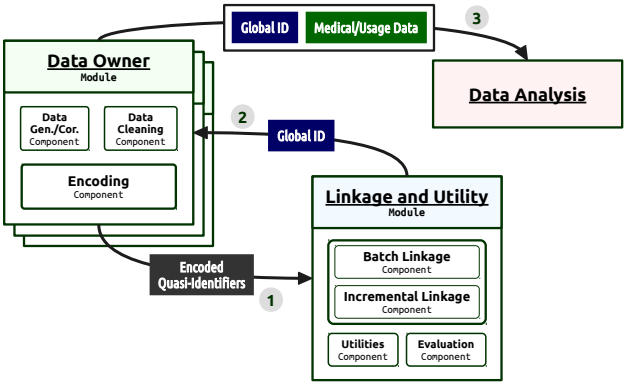

Figure 1: PRIMAT architecture.

communication between DOs or provide parameter recommendations.

From a high-level perspective, each PPRL process consists of three steps (see Fig. 1): (1) Each DO prepares their records for linkage and encodes the identifying attributes (quasi-identifiers) of each entity. The encoded quasi-identifiers are then send to the LU. (2) The LU conducts the linkage and returns global IDs (pseudonyms) to the DOs. Thereby, every pair of record that has the same global ID is considered as match. (3) The DOs send the medical/usage data together with the entities' global ID to the data analyst where the data for the same ID can be combined for enhanced analysis.

PRIMAT consists of several key components that cover each step of the PPRL process pipeline. Based on the two aforementioned roles typically occurring in PPRL protocols, the components are separated into two modules, namely a *DO module* for pre-processing and encoding of database records and a *linkage and utility module* which mainly provides various linkage techniques for use at the LU. Moreover, each module provides functions for sending/receiving and generating/parsing data and parameter files. In the following subsections, we give a description of each component in the two modules. In Tab. 1 the current implementation status of each component is listed. Several of the planned extensions are already existing, e.g., for distributed linking [3, 7], and only have to be integrated into PRIMAT. For multi-party and incremental linkage we plan to add PPRL variations of our clustering approaches in [13, 15] using Bloom filters instead of unencoded records.

4.1 DO Module

The DO module provides procedures that are required for DOs to appropriately prepare their records for linkage. It consists of the following three key components (see Fig. 1):

D_1: Data Generation and Corruption. This component allows DOs to generate realistic synthetic datasets, possibly based on real-world data, which can be used for probing and balancing PPRL workflows. By inspecting the linkage results it thus supports selection of appropriate methods and fine-tuning of parameter settings.

D_2: Data Cleaning. The data cleaning tool allows extensive pre-processing of the DOs' data. In particular, it provides common operations to clean and standardize data, e.g., removing/replacing of values or splitting/merging of attributes. Furthermore, it is planned to support traditional record linkage to remove intra-source duplicates.

Table 1: PRIMAT current status of implementation.

Comp.	Function / Feature	Status
D_1	Data generation	Implemented
	Data corruption	Planned
D_2	Split/merge/remove attributes	Implemented
	Replace/remove unwanted values and stopwords	Implemented
	OCR transformation	Implemented
	Intra-source record linkage	Planned
D_3	Bloom filter encoding and hardening techniques	Implemented
	Client-side standard blocking	Implemented
$D_1 - D_3$	Graphic user interface	Planned
L_1	Pre-processing templates	Planned
	Private schema matching	Planned
L_2	Standard & LSH-based blocking; metric space filtering	Implemented
	Threshold-based classification	Implemented
	Post-processing	Implemented
	Multi-threaded processing	Implemented
	Distributed processing	Planned
	Multi-party support	Planned
L_3	Incremental linkage	Planned
L_4	Quality & scalability evaluation	Implemented
	Privacy-preserving match result visualization	Planned

D_3: Encoding. The encoding component works as follows: At first, the identifying attributes of each entity are transformed into a set of relevant record features. Secondly, these record features are encoded. We focus on Bloom-filter-based encoding techniques as they are widely used in the PPRL domain [20, 21]. Since standard Bloom filters have a relatively high disclosure risk, also recent hardening techniques are provided to enhance privacy.

4.2 Linkage and Utility Module

The linkage and utility module mainly provides a wide-range of methods and procedures required for linkage. It offers the following four components:

L_1: Utilities. For DOs it is challenging to select and agree on appropriate methods and parameters. Hence, the utility component provides methods to automatically determine parameters based on masked sample data. In particular, private schema matching [2] and pre-processing templates [16] support DOs to consistently prepare their data for linkage.

L_2: Batch Linkage. The batch linkage tool is the main component of this module as it implements various linkage techniques. In particular, standard and LSH-based blocking as well as metric space filtering approaches based on our previous work [3, 7] are provided to reduce the complexity of the linkage. For classification, we provide frequently used binary similarity measures, e.g., Jaccard or Dice similarity [20]. Furthermore, we included our recent post-processing methods [4] that can significantly increase linkage quality by selecting the best match candidates if there are multiple records exceeding a given similarity threshold.

L_3: Incremental Linkage. This component is similar to L_2 but provides database support to store and retrieve previous match results, e.g., match clusters. Hence, new records can be added continuously without repeating linkage for every record.

L_4: Evaluation Tool. It is important for both practitioners and researchers to evaluate PPRL methods and workflows. For researchers, such evaluation is straightforward since ground truth information, i.e., the correct linkage result, is usually available. Thus, quality metrics like precision, recall and F-measure can be determined. In practice, however, ground truth information is usually not available and, in addition, manual inspection of records or linkage results is normally prohibited due to privacy constraints. Nevertheless, some metrics, like runtime, reduction ratio or number and average size of blocks, can still indicate bad parameter or method choices. Furthermore, privacy-preserving visualization approaches can be utilized to grant authorized experts insights into linkage results without degrading privacy [10, 14].

5. DEMONSTRATION DESCRIPTION

During the demonstration we will show how PRIMAT can be employed for PPRL. In a first scenario we focus on usability and will illustrate how users can define and execute typical PPRL workflows. For this purpose, we provide an example program for batch linkage of two persons' address books to identify persons commonly known by both persons. We demonstrate how to construct a PPRL workflow by selecting methods from the components provided by PRIMAT. The workflow can then be executed and the linkage result can be inspected. In the second scenario we demonstrate how PRIMAT is used to comparatively evaluate PPRL approaches using datasets with distinct characteristics, e.g., size, corruption level, number of duplicates. The datasets are drawn from the North Carolina voter registration (see `https://www.ncsbe.gov/`) or generated by PRIMATs data generator component. We focus on evaluating the linkage quality (precision, recall, F-measure) as well as efficiency (number of record comparisons, runtimes) of pre-configured PPRL workflows that utilize different blocking and post-processing methods. The audience may interactively manipulate certain parameters, e.g., similarity threshold, number of blocking keys, to assess robustness of the strategies and to identify trade-offs between linkage quality and efficiency.

6. CONCLUSION

We presented our PPRL toolbox PRIMAT that allows a flexible definition, execution and evaluation of PPRL workflows. PRIMAT provides various state-of-the-art encoding and linkage techniques covering the entire PPRL process and thus drastically reduces the effort to deploy PPRL in practice.

7. ACKNOWLEDGMENTS

This work is supported by the German Federal Ministry of Education and Research under grant `BMBF 01IS18026B`.

8. REFERENCES

[1] P. Christen. *Data matching*. Springer, 2012.
[2] T. P. da Nobrega et al. Blind attribute pairing for privacy- preserving record linkage. In *ACM SAC*, pages 557–564, 2018.
[3] M. Franke et al. Parallel privacy-preserving record linkage using LSH-based blocking. In *IoTBDS*, pages 195–203, 2018.
[4] M. Franke et al. Post-processing methods for high quality privacy-preserving record linkage. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 263–278. Springer, 2018.
[5] M. Franke et al. ScaDS research on scalable privacy-preserving record linkage. *Datenbank-Spektrum*, 2019.
[6] A. Gibberd et al. Lung cancer treatment and mortality for Aboriginal people in New South Wales, Australia: results from a population-based record linkage study and medical record audit. *BMC Cancer*, 16(1):289, 2016.
[7] M. Gladbach et al. Distributed privacy-preserving record linkage using pivot-based filter techniques. In *ICDEW*, pages 33–38, 2018.
[8] A. Karakasidis et al. PRIVATEER: A private record linkage toolkit. In *CAiSe Forum*, pages 197–204, 2015.
[9] D. Karapiperis et al. LSHDB: A parallel and distributed engine for record linkage and similarity search. In *ICDMW*, pages 1–4. IEEE, 2016.
[10] H.-C. Kum et al. Privacy preserving interactive record linkage (PPIRL). *JAMIA*, 21(2):212–220, 2014.
[11] M. Lablans et al. A RESTful interface to pseudonymization services in modern web applications. *BMC Medical Informatics and Decision Making*, 15(1):2, 2015.
[12] Q. Luo et al. Cancer-related hospitalisations and unknown stage prostate cancer: a population-based record linkage study. *BMJ Open*, 7(1), 2017.
[13] M. Nentwig and E. Rahm. Incremental clustering on linked data. In *ICDMW*, 2018.
[14] E. D. Ragan et al. Balancing privacy and information disclosure in interactive record linkage with visual masking. In *ACM CHI*, page 112. ACM, 2018.
[15] A. Saeedi, E. Peukert, and E. Rahm. Using link features for entity clustering in knowledge graphs. In *ESWC*, pages 576–592. Springer, 2018.
[16] K. Schmidlin et al. Privacy preserving probabilistic record linkage (P3RL): a novel method for linking existing health-related data and maintaining participant confidentiality. *BMC Medical Research Methodology*, 15(1), 2015.
[17] R. Schnell et al. A toolbox for record linkage. *Austrian Journal of Statistics*, 33(1-2):123–133, 2004.
[18] R. Schnell et al. Performance of different methods for privacy preserving record linkage with large scale medical data sets. In *Int. Health Data Linkage Conf.*, 2014.
[19] C. Toth et al. SOEMPI: A secure open enterprise master patient index software toolkit for private record linkage. *AMIA Symp.*, pages 1105–1114, 2014.
[20] D. Vatsalan et al. A taxonomy of privacy-preserving record linkage techniques. *Inf. Syst.*, 38(6):946–969, 2013.
[21] D. Vatsalan et al. Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges. *Handbook of Big Data Technologies*, 2017.