Privacy-Preserving Record Linkage: Past, Present and Yet-to-Come

Lefteris Stetsikas
Department of Informatics and Telecommunications,
National & Kapodistrian University of Athens
Athens, Greece
istetsikas@di.uoa.gr

Dimitrios Karapiperis
School of Science and Technology,
International Hellenic University
Thessaloniki, Greece
dkarapiperis@ihu.edu.gr

George Papadakis
Department of Informatics and Telecommunications,
National & Kapodistrian University of Athens
Athens, Greece
gpapadis@di.uoa.gr

Manolis Koubarakis
Department of Informatics and Telecommunications,
National & Kapodistrian University of Athens
Athens, Greece
koubarak@di.uoa.gr

Abstract

Privacy-preserving record linkage (PPRL) constitutes a critical technique for integrating sensitive data across organizational boundaries without compromising the privacy and confidentiality of personal information. Over the past two decades, PPRL has evolved from simple hash-based exact matching methods to sophisticated approximate matching techniques that address the complex challenges of scalability and linkage quality.

This tutorial provides a comprehensive overview of PPRL, organizing the relevant works in chronological order. We begin with the *fundamental challenges* that motivated PPRL, i.e., the legal restrictions on data sharing (e.g., GDPR), the need for approximate matching in the presence of data errors, and scalability requirements for large databases. Next, we focus on the *past*: we discuss the evolution of PPRL from early secure hash encoding techniques to more advanced privacy-preserving methods (e.g., secure multi-party computation, k-anonymity etc). The *present* section focuses on current state-of-the-art approaches that address the three main challenges of PPRL: scalability, variety and end-to-end privacy preservation. We then focus on the *future*, identifying critical open challenges and promising research directions. A hands-on section demonstrates the open-source software we have developed in Python for integrating the main PPRL tools. We also discuss adversary models, privacy vulnerabilities, and evaluation frameworks for assessing scalability, linkage quality, and privacy protection. We conclude with a discussion about the open challenges and promising research directions. Overall, the tutorial takes special care to synthesize theoretical foundations, current methodologies, and future research trajectories, equipping attendees with comprehensive knowledge to advance PPRL research and deploy privacy-preserving solutions in practice.

Keywords

PPRL, Bloom filters, Blocking, Matching, GDPR

1 Goals and Objectives

Privacy-Preserving Record Linkage (**PPRL**) is a data integration technique that enables the matching of records referring to the same entity across disparate databases without exposing personally identifiable information (**PII**). PPRL employs encryption, hashing, or other obfuscation methods to mask sensitive attributes before the linkage process, allowing organizations to combine data from multiple sources while maintaining privacy compliance with regulations such as HIPAA and GDPR.

PPRL has become increasingly critical as organizations face growing regulatory constraints and privacy concerns when integrating person-level data across institutional and jurisdictional boundaries. The technique enables essential applications in healthcare research, fraud detection, and population health studies that would otherwise be impossible due to privacy regulations. By allowing data linkage without direct PII sharing, PPRL supports privacy-preserving data integration while minimizing the risk of unauthorized access and re-identification, in line with modern data governance principles.

While traditional *Record Linkage* (**RL**) operates directly on plaintext identifiers to match records across datasets, PPRL introduces cryptographic protections that fundamentally alter the matching process. In conventional record linkage, data owners must either share raw PII or rely on a fully trusted third party with access to all sensitive information. PPRL addresses these privacy vulnerabilities by ensuring that matching occurs on encoded or encrypted data, where no single party has access to both the encryption keys and the linkage results.

In general, PPRL works as follows:

- *Preprocessing and Standardization.* Data owners first identify and clean the quasi-identifiers (QIDs) that will be used for linkage, such as name, date of birth, and address. The parties agree on standardization rules to transform these fields into a common format, handling issues like case sensitivity, missing values, and abbreviations. This preprocessing step is performed locally by each data owner on their own systems.
- *Encoding and Encryption.* After preprocessing, the cleaned QID values must be encoded or encrypted using methods that preserve similarity while protecting privacy. A key escrow or trusted configuration service typically provides encryption keys and configuration parameters to all data partners. Each data owner applies the encoding method to transform their plaintext identifiers into encrypted representations that can be compared without revealing the underlying identities.
- *Blocking.* The encoded records are then subjected to blocking or indexing techniques to reduce the number of comparisons required. This step groups potentially matching records together so that only candidates within the same block need to be compared, which is crucial for scalability to large datasets.
- *Matching and Comparison.* A linkage agent or linkage unit performs the actual record matching on the encoded data. The

EDBT '26, Tampere (Finland)

10.48786/edbt.2026.79

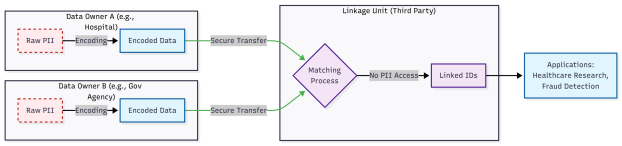

Figure 1: The PPRL workflow, which ensures that matching occurs on encoded data, preventing direct access to PII.

linkage agent compares the encrypted records using adapted similarity metrics and determines which records likely refer to the same entity. Critically, the linkage agent does not have access to the encryption keys, preventing them from decrypting the data and accessing PII.

- *Result Generation and Sharing.* The linkage agent creates unique identifiers (such as Linked IDs) for matched records and returns these identifiers to the data owners. Data owners can then use these shared identifiers to link their analytical data without exposing sensitive information. The final linked dataset can be shared with researchers who receive only the matched records with their unique identifiers.

Figure 1 outlines the workflow where data owners essentially 'lock' their data before transmission, shifting the trust boundary away from the third-party linkage unit.

The goal of this tutorial is to provide a comprehensive overview of PPRL, discussing the seminal works in the field, the current state-of-the-art and directions for future improvements. We begin by examining the three foundational challenges of PPRL:

(1) achieving high linkage quality despite noisy real-world data,
(2) ensuring scalability to large databases with millions of records,
(3) preserving privacy throughout the entire linkage process.

We discuss the end-to-end PPRL pipeline in Figure 1, explaining how each step must be adapted to operate on encoded rather than plaintext data. We formalize the PPRL problem by defining key concepts such as quasi-identifiers (QIDs), database owners, linkage units, and the matching decision model that classifies record pairs without revealing sensitive information. Practical applications are illustrated through compelling real-world scenarios such as public health surveillance (adverse drug reaction studies, infectious disease outbreak detection), crime and fraud detection across law enforcement agencies.

Another objective is to trace the historical development of PPRL techniques from the late 1990s through the mid-2010s, when foundational methods were established. Early encoding approaches relied on simple cryptographic hash functions (MD5, SHA-1) that enabled exact matching of encoded values, but failed to support approximate matching functions, which are necessary for handling real-world data errors. This was a significant limitation that motivated subsequent research. Phonetic encoding techniques such as Soundex and NYSIIS emerged as privacy-preserving blocking methods that inherently protect privacy, while grouping similar-sounding names. On the downside, though, they suffer from language dependence and vulnerability to frequency attacks [13]. Generalization techniques including k-anonymity were adopted to prevent re-identification by ensuring that each encoded value is shared by at least k records. This comes, though, at the cost of reduced linkage accuracy. In the matching phase, non-learning methods dominated, including secure multi-party computation (SMC) protocols for set intersection and scalar product calculations [7, 10], as well as distance-preserving embedding techniques that mapped strings into Euclidean or Hamming spaces. These early approaches established important cryptographic foundations but were computationally expensive and often impractical for large-scale applications. We also present the first learning-based classification methods, that have been adapted from traditional record linkage. We stress, though, that obtaining suitable training data in privacy-preserving settings remained challenging.

We also aim to present a comprehensive overview of the contemporary PPRL techniques. They typically leverage Bloom filter-based encoding [11], which supports approximate matching, while providing computational efficiency. We examine various Bloom filter variants including Cryptographic Long-term Key (CLK) encoding that hashes entire attribute values, Record-level Bloom Filters (RBF) that concatenate separate filters per attribute, and hybrid approaches (CLKRBF) that balance privacy and accuracy. Scalable blocking techniques have evolved significantly. For example, LSH-based methods use random bit sampling from Bloom filters to create locality-sensitive hash codes [5, 6], phonetic blocking enhanced with fake record injection is used to mitigate frequency attacks, and tree-based indexing uses multi-bit tree structures to achieve near-linear complexity for large databases [9]. Filtering and acceleration methods further reduce the comparison costs: for example, length filtering prunes pairs whose token counts differ beyond threshold-compatible ranges, prefix filtering exploits sorted token orders and P4Join combines multiple filters for efficient candidate generation. Parallel and distributed PPRL implementations leverage MapReduce frameworks to partition blocking and comparison workloads across clusters, while GPU-based approaches exploit massive parallelism for Bloom filter similarity calculations. Modern matching techniques apply approximate similarity measures (Dice, Jaccard coefficients) directly on encoded Bloom filters, with threshold-based and probabilistic classification methods adapted to operate without access to plaintext values.

A core target of this tutorial is to discuss the future of PPRL, exploring cutting-edge techniques from traditional record linkage that can be adapted to the PPRL. The first one is Meta-blocking techniques refine the candidate pairs generated by an initial blocking based on Bloom filters: comparison propagation avoids repeated comparisons, while graph-based pruning methods discard candidate pairs with weak co-occurrence patterns and, thus, low weights. The goal of these methods in to substantially increase precision, while maintaining blocking recall close to the initial high levels. The second promising, generic approach that can be applied to PPRL is Approximate Nearest Neighbor Search (*ANNS*) using libraries like FAISS[1] and indices like Hierarchical Navigable Small World (HNSW) [8]. These techniques can be directly applied to Bloom filter vectors, representing an alternative paradigm to traditional PPRL blocking with a tunable trade-off between precision and recall. Finally, clustering techniques can be applied at the outcomes of matching to resolve conflicts in matched pairs, i.e., cases where the same record from Data Owner A is classified as matching with two different records from Data Owner B. Such cases can be addressed by applying bipartite graph matching algorithms to produce consistent entity equivalence classes. We apply these methods to established PPRL datasets using a variety of parameter configurations and methods, performing the first relevant experimental analysis in the literature. We analyze the experimental results and highlight

[1] https://github.com/facebookresearch/faiss

the most interesting insights in terms of effectiveness (precision, recall, F-measure), time efficiency and scalability.

We also discuss the evaluation of PPRL techniques in terms of: (i) privacy protection, (ii) computational efficiency, and (iii) linkage quality. We examine adversary models that formalize threat assumptions, including: (i) the widely-used honest-but-curious (HBC) model where parties follow protocols but attempt to infer sensitive information, (ii) the stronger malicious model where parties may arbitrarily deviate from protocols, and (iii) emerging covert and accountable computing models that detect misbehavior with high probability. We also discuss ways of exploiting vulnerabilities like dictionary attacks (which re-encode known values to match encrypted data), frequency attacks (which exploit value distributions even with keyed hashing), cryptanalysis attacks on Bloom filters using constraint satisfaction solvers, composition attacks combining information from multiple linkages, and collusion between database owners and linkage units. Privacy measures quantify the protection levels through entropy (randomness in encoded values), information gain between plaintext and encoded data, disclosure risk probabilities, and formal security proofs under specific adversary models. Yet, standardized privacy metrics remain an open research challenge. We also discuss ways of assessing linkage quality, namely pairs completeness and pairs quality for blocking as well as precision, recall, and F-measure metrics for the final classification results. Fault-tolerance analysis examines how techniques degrade under various data error types like typographical errors, missing values, and formatting inconsistencies. Finally, we discuss benchmark datasets, contrasting real-world collections with synthetic data generators that enable controlled experiments with known ground truth and reproducible evaluation.

Another goal is to provide participants with direct experience using state-of-the-art open-source PPRL systems that implement the techniques covered in previous sections. We demonstrate Anonlink [2], PRIMAT (Privacy-Preserving Record Linkage Toolbox) [3], AMPPERE [15], Linkja[2], LSHDB [4] and the PPRL Toolkit[3]. For each tool, we discuss its architecture (e.g., two-party vs. three-party protocols) as well as the supported methods for encoding, blocking, and matching. We then introduce our own PPRL Python library that integrates the aforementioned meta-blocking, ANNS-based matching, and clustering techniques, providing participants with a unified end-to-end pipeline framework. The hands-on exercises guide participants through complete PPRL workflows: preprocessing and standardizing input data, selecting appropriate encoding parameters, configuring blocking strategies, executing matching algorithms, and interpreting evaluation results. The goal is to equip attendees with the skills to deploy PPRL solutions in their own infrastructure, while understanding the pros and cons of each tool.

Our final objective is to address critical open challenges and future research directions that will shape the next generation of PPRL techniques. Hardening techniques have emerged in response to cryptanalysis vulnerabilities, including differential privacy mechanisms that add calibrated noise to encoded data, autoencoder-based obfuscation that further obscures Bloom filter patterns, and frequency-based balancing methods that normalize Hamming weights to prevent length-based attacks. The best practices for secure deployment emphasize the separation principle (isolating identifying data from microdata), the secure key exchange protocols on top of the public-key infrastructure, employee confidentiality agreements to mitigate insider threats, and regular security audits of linkage systems. Dynamic data and real-time linking present scalability challenges as databases continuously grow and require incremental updates to linkage results without full re-computation. Multi-party PPRL scalability faces exponential complexity growth when linking more than two databases, requiring novel protocols that balance privacy protection with computational feasibility. Integration with privacy-by-design frameworks ensures that PPRL is embedded within comprehensive data governance strategies that span the entire data lifecycle from collection through analysis to deletion. Deep learning for PPRL represents an emerging frontier, with recent works exploring Siamese neural networks for similarity learning on encoded representations, federated learning approaches that train models without centralizing data, and representation learning methods that transform Bloom filters into semantically meaningful embeddings. We conclude with reflections on the maturation of PPRL from theoretical curiosity to practical necessity, emphasizing that while significant progress has been made, ongoing research is essential to address evolving privacy regulations and emerging attack vectors.

Overall, our tutorial provides researchers with a complete coverage of the state-of-the-art PPRL methods along with a discussion of the main open research problems. Practitioners get a good overview of the benefits of the primary PPRL methods and learn how to use them to improve the productivity of their businesses. They also learn to identify the methods or products that are more suitable for a particular task at hand, or better fit their general needs. Additionally, the audience and especially the developers of information integration tools benefit from the hands-on session, learning how to integrate PPRL into their applications. Developers also become acquainted with novel ideas that could well improve their existing products.

Related Tutorials. While a foundational tutorial [1] has covered the early principles of PPRL, our tutorial provides the first novel, holistic, and systematic view of the field's evolution over the last decade. Unlike previous works, we stress the current state-of-the-art in every step of the end-to-end pipeline, introducing a novel taxonomy that organizes these methods. We notably discuss major advances that have not been covered in depth before, including the application of generic RL techniques—such as Meta-blocking and Approximate Nearest Neighbor Search (ANNS)—to the PPRL context. Furthermore, we address the 'data-driven' generation of PPRL , covering modern deep learning and vector-based matching techniques that are absent from older educational materials. Special care is also taken to present the first large-scale experimental analysis of these techniques , alongside open-source tools for applying them in practice. To the best of our knowledge, no other tutorial comprehensively covers these modern algorithmic advancements and their practical trade-offs.

2 Scope and Coverage

Our tutorial aims to provide an overview of the state-of-the-art techniques for PPRL over the years. It comprises 7 sections, with each one lasting 20 minutes. More emphasis is devoted to the hands-on session, which discusses the main ER tools and demonstrates the latest version of our PPRL Python library (~30 minutes). Together with a couple of minutes for each session, the intended duration of the tutorial is *3 hours*. The content of the individual sessions is outlined below:

I. Introduction and motivation

[2]https://linkja.github.io/
[3]https://github.com/datasciencecampus/pprl_toolkit

- Foundational Challenges of PPRL
- Preliminaries on PPRL
- Fundamental Assumptions, Principles and Definitions
- Practical Applications

II. The Past of PPRL
- Encoding
- Blocking
- Matching (Non-learning & Learning-based Methods)

III. The Present of PPRL
- Bloom Filter Based Techniques
- Scalable Blocking Techniques (LSH, phonetic, tree-based)
- Filtering and Acceleration Methods (length, prefix, position filtering, P4Join)
- Parallel and Distributed PPRL (MapReduce, GPU-based)
- Matching

IV. The Future of PPRL
- Integrating Meta-blocking techniques
- Integrating approximate nearest-neighbor search (ANNS)
- Integrating clustering techniques
- Experimental results

V. Evaluation Methods
- Adversary models
- Privacy attacks (dictionary, frequency, cryptanalysis, composition, collusion)
- Privacy measures (entropy, information gain, RIG, disclosure risk, security proofs)
- Assessing linkage quality
  - Pairs completeness, pairs quality
  - Precision, recall, F-measure
  - Fault-tolerance to data errors
- Benchmark datasets (real & synthetic)

VI. Hands-on Session: PPRL tools
- The state-of-the-art open-source PPRL tools
  - Anonlink, PRIMAT, Linkja, PPRL Toolkit, AMPPERE
- Our PPRL Python library

VII. Challenges and Final Remarks
- Hardening techniques (differential privacy, autoencoders, balancing)
- Best practices for secure deployment
- Dynamic data and real-time linking
- Multi-party PPRL scalability
- Integration with privacy-by-design
- Deep learning for PPRL
- Conclusions

3 Intended Audience and Material

Our tutorial adopts a practical, example-driven approach that emphasizes intuitive understanding over mathematical formalism and formal proofs. The only prerequisite is a fundamental familiarity with data management concepts, making the content accessible to a diverse audience spanning academia, industry, and practice. The tutorial is designed for students, researchers, practitioners, and developers—essentially anyone seeking to understand the core methodologies for building scalable, high-quality, end-to-end PPRL systems. We cover approaches that range from traditional algorithmic techniques to modern graph-based clustering [12] and machine learning methods [14] that improve classification and matching.

In addition to the theoretical background in the state-of-the-art in the field, the tutorial also presents available PPRL-related resources, enabling the participants to directly work on the particular domain. The discussed resources include available data as well as the state-of-the-art tools for performing end-to-end PPRL, like our Python library[4], which can be readily used to tackle PPRL problems via numerous combinations of the most prominent methods.

4 Presenters

The tutorial is given by four presenters:

(1) *Lefteris Stetsikas* is a master student at the at the Department of Informatics of the University of Athens. He will present the hands-on section.
(2) *Dimitrios Karapiperis* is an adjunct lecturer at the School of Science and Technology of the International Hellenic University, Greece. He will present the past and present sections.
(3) *George Papadakis* is a Research Fellow at the Department of Informatics of the University of Athens, Greece. He will present the "Future of PPRL" and the "Evaluation Methods" sections, emphasizing the application of ANNS and meta-blocking techniques on end-to-end PPRL.
(4) *Manolis Koubarakis* is a Professor of Computer Science at the Department of Informatics of the University of Athens. He will present the introductory section.

Acknowledgements. This work was partially funded by the EU Horizon project RECITALS (Grant Agreement 101168490).

References

[1] Peter Christen and Vassilios Verykios. 2012. A Tutorial on Privacy-Preserving Record Linkage. In *PAKDD*.
[2] CSIRO's Data61. 2017. Anonlink Private Record Linkage System. https://github.com/data61/clkhash.
[3] Alexandros Karakasidis, Georgia Koloniari, and Vassilios S Verykios. 2015. PRIVATEER: A Private Record Linkage Toolkit. In *CAiSe Forum*. 197–204.
[4] Dimitrios Karapiperis, Aris Gkoulalas-Divanis, and Vassilios S Verykios. 2016. LSHDB: a parallel and distributed engine for record linkage and similarity search. In *International Conference on Data Mining Workshops (ICDMW)*. IEEE, 1–4.
[5] Dimitrios Karapiperis, Aris Gkoulalas-Divanis, and Vassilios S Verykios. 2017. Distance-aware encoding of numerical values for privacy-preserving record linkage. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 135–138.
[6] D. Karapiperis and V.S. Verykios. 2015. An LSH-based Blocking Approach with a Homomorphic Matching Technique for Privacy-Preserving Record Linkage. *TKDE* 27, 4 (2015), 909–921.
[7] Ibrahim Lazrig, Toan C Ong, Indrajit Ray, Indrakshi Ray, Xiaoqian Jiang, and Jaideep Vaidya. 2018. Privacy preserving probabilistic record linkage without trusted third party. In *Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 1–10.
[8] Y. Malkov and D. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (2018), 824–836.
[9] Thilina Ranbaduge, Peter Christen, and Dinusha Vatsalan. 2014. Tree Based Scalable Indexing for Multi-Party Privacy-Preserving Record Linkage. In *Australasian Data Mining*. Brisbane.
[10] Thilina Ranbaduge, Dinusha Vatsalan, and Peter Christen. 2020. Secure Multi-party Summation Protocols: Are They Secure Enough Under Collusion? *Transactions on Data Privacy* 13, 1 (2020), 25–60.
[11] R. Schnell, T. Bachteler, and J. Reiher. 2009. Privacy-preserving record linkage using Bloom filters. *BMC Med Inform Decision Making* 9, 1 (2009).
[12] Dinusha Vatsalan, Peter Christen, and Erhard Rahm. 2020. Incremental clustering techniques for multi-party Privacy-Preserving Record Linkage. *Data & Knowledge Engineering* (2020), 101809.
[13] Anushka Vidanage, Peter Christen, Thilina Ranbaduge, and Rainer Schnell. 2023. A Vulnerability Assessment Framework for Privacy-preserving Record Linkage. *ACM Transactions on Privacy and Security* 26, 3 (2023).
[14] Wanli Xue, Dinusha Vatsalan, Wen Hu, and Aruna Seneviratne. 2020. Sequence Data Matching and Beyond: New Privacy-Preserving Primitives Based on Bloom Filters. *IEEE Transactions on Information Forensics and Security* 15 (2020), 2973–2987.
[15] Yixiang Yao, Tanmay Ghai, Srivatsan Ravi, and Pedro Szekely. 2021. AMPPERE: A Universal Abstract Machine for Privacy-Preserving Entity Resolution Evaluation. In *ACM International Conference on Information and Knowledge Management*. 2394–2403.

[4] https://github.com/AI-team-UoA/PPRL-Dev