Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik

Master Thesis Bioinformatics

Privacy-Preserving Record Linkage Using Secure Multi-Party Computation

Noah JM Dietrich

21.01.2024

Reviewers

Dr. Mete Akgün
(Methods in Medical Informatics)
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Prof. Dr. Nico Pfeifer
(Methods in Medical Informatics)
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Dietrich, Noah JM:
Privacy-Preserving Record Linkage Using Secure Multi-Party Computation
Master Thesis Bioinformatics
Eberhard Karls Universität Tübingen
Thesis period: 21.07.2023-21.01.2024

Abstract

Write here your abstract.

Zusammenfassung

Bei einer englischen Masterarbeit muss zusätzlich eine deutsche Zusammenfassung verfasst werden.

Acknowledgements

I want to thank the following people, who either made my thesis possible or greatly improved my time working on it:

- Mete Akgün for supervising my thesis
- Nico Pfeifer for being the second reviewer
- Ali Burak Ünal, Mete Akgün, Şeyma Selcan Mağara and everyone who contributed to CECILIA
- Şeyma Selcan Mağara for helping me with CECILIA and listening to me rubberducking on many occasions
- Cem Ata Baykara, Arjhun Swaminathan and Şeyma Selcan Mağara for welcoming me in their office and many nice coffe breaks
- Jonas Fischer for all the lunch breaks and trips to the Mensa
- Jonas Fischer for proof-reading my thesis

Contents

List of Figures

List of Tables

List of Abbreviations

CECILIA Comprehensive Secure Machine Learning Framework

FCM fuzzy c-means

HMAC hashed message authentication code

IDAT identity data

MainSEL Mainzelliste SecureEpiLinker

PPRL privacy-preserving record linkage

SMPC secure multi-party computation

SSN social security number

XOR bitwise exclusive-OR

Chapter 1

Introduction

Start with a comprehensive introduction about the questions of your thesis.

The thesis could include a background section, which also could become one or two separate chapters.

Do not forget to also give a short overview of the structure of the thesis in this chapter, for example as follows:

This thesis is structured as follows:

Chapter 2

Background

This chapter aims to provide a comprehensive overview of the key concepts, challenges, and advancements in privacy-preserving computation, specifically secure multi-party computation (SMPC), and record linkage.

2.1 Secure Multi-Party Computation

2.1.1 Adversary Models

2.1.2 Yao's Garbled Circuits

2.1.3 Arithmetic Sharing

2.1.4 Boolean Sharing

2.2 Record Linkage

2.2.1 Concept and Application

Record linkage is the process of matching data points (records) related to the same entity, i.e. human, but originating from different datasets. If a commonly shared unique identifier exists, this process is trivial. For instance, the social security number (SSN) in the United States of America, in theory, serves this purpose. In practice, even this is not perfect due to erroneous data. Unique identifiers may have spelling mistakes or typos, so that a match is no longer possible. More importantly, there are many instances where such a unique identifier is not readily available. In these instances, the matching must be done on features of the entity that are found in both datasets. The term record linkage in its broadest definition may be used to refer to any such process of linking datasets, such as In this work, the term record linkage pertains to the linking of datasets using identity data (IDAT). These include information such as name, date of birth and place of residence of a patient. In medical research, record linkage is commonly employed to match patient data from various healthcare institutions, creating a dataset that combines information from different sources. Individually, each IDAT point is insufficient to provide an unambiguous match, which is why they are often referred to as quasi-identifiers. However, when combining enough quasi-identifiers, such a match can be achieved. Issues in record linkage based on IDAT arise when the data is either non-unique or inaccurate, which leads us to error types and their sources.

IDAT inaccuracies arise through various means, primarily during manual data entry. Human input, typically via keyboard, introduces errors stemming from multiple sources. Spelling mistakes may result from transcribing orally received information or copying text-based information. Oral communication may introduce discrepancies due to similar-sounding words or varied spellings of the same name. Copying text can lead to confusion between visually similar letters (e.g., I and l). Typographical errors, induced by slips during typing, contribute to insertions, substitutions, transpositions, or deletions of letters. Lastly, misplacement of individual attributes, such as swapping a patient's first name and surname, represents another source of error.

2.2.2 Approaches

To tackle the complexities of record linkage, three main classes of record linkage have emerged: Rule-based (deterministic) record linkage, probabilistic record linkage and more advanced forms using machine learning classifiers.

Deterministic record linkage describes the procedures where a binary outcome is obtained. Records are either classified as matching or not matching. Because this approach requires high data quality due to its inability to deal with errors, it is not reported often. One example where such an approach has been used appears to be Surveillance, Epidemiology and End Results-Medicare linked dataset created by the US National Cancer Institute as reported by Dusetzina et al. [1], though the primary sources reporting this are not available anymore. Here, records were first matched on the SSN and combinations of first and last name, month of birth and sex. Then, a second round of linkage is carried out for records without a SSN or which were not matched in the previous round. In this round, first name, last name, month of birth and sex are matched, in addition to either a part of the SSN or a combination of day of birth, year of birth, middle initial and date of death. It is noteworthy that despite being an example of deterministic record linkage, the first and last name allow for fuzzy matches to account for nicknames. This is an example of data preprocessing, which will be described in more detail later.

Probabilistic record linkage is a more lenient approach in that rather than matching records based on hard rules, a match probability or record similarity metric is computed for pairs of records. This metric is subsequently thresholded to categorise record pairs into matches and non-matches. Probabilistic record linkage is generally better for dealing with low quality data with many errors. It has the benefit of being flexible in weighing up specificity against sensitivity. Namely, a researcher requiring very high sensitivity may set a low threshold to catch as many true matches as possible, at the detriment of increasing false positive matches. In contrast, a researcher interested only in the highest quality matches may set a high threshold, thus filtering out false positives at the cost of sensitivity. There are a number of algorithms employing probabilistic record linkage.

Perhaps the most classical example of a probabilistic record linkage model is the Fellegi-Sunter model [2]. To compute the probability of a match between a given pair of records, Fellegi and Sunter start with a prior probability that two randomly drawn records match. Each record attribute is compared individually and is given a partial match weight based on how similar the attribute is between the two records. The sum of all match weights is used together with the prior probability to compute a specific probability that the two records match. Each partial match weight is estimated via m and u probabilities. The relation between an attribute pair may be match/no match or a similarity metric, depending on the specific implementation of the Fellegi-Sunter model. m is the probability of observing the attribute pair relation on hand given the two records are a match. Likewise, u is the probability of the relation given the two records are not a match. m and u can be computed in various ways, either from the given datasets themselves or from prior knowledge about the populations the records are drawn from. When implemented well, m and u will incorporate information about error rates of an attribute as well as individual frequencies of the given attributes. Since Fellegi and Sunter only describe a general mathematical framework, the actual result depends on the specific implementation and probability computation, of which Fellegi and Sunter give a few suggestions.

2.2.3 Blocking

In the most naive implementation of record linkage, each record in set A has to be compared to each record in set B, giving a quadratic complexity of $|A| \times |B|$ comparisons. To reduce this number, *blocking* is employed. Here, records are grouped into blocks by differing criteria, so only those records within each block have to be compared. In its most basic form, blocks may simply be defined based on a specific attribute such as the city or year of birth. This technique is extremely simple to implement, but it has a significant downside – if there is an error such as a misspelling or typo in a record's attribute selected for blocking, it becomes impossible to match this record. Therefore, it is only appropriate if the researcher can be certain a specific attribute is relatively error-free across all records. Therefore, more advanced blocking techniques have been developed. Blocks may be defined by clustering the records.

Clustering is the process of dividing data points into groups such that within-variance of each group is minimised, while between-variance between groups is maximised. This variance is computed based on a distance or similarity function. Many such functions exist – distance functions include the Pearson correlation distance, the standardised Euclidean distance and the Minkowski distance; for similarity functions, these may be the Jaccard similarity or the Hamming similarity. Clustering algorithms can be divided into nine categories according to Xu and Tian [3]. These categories describe what the algorithm is based on. Partition-based algorithms One such algorithm is K-means clustering, where k cluster centers are defined. Data points are assigned to the cluster with the nearest center, which is then updated to include the data point [4]. Hierarchy-based algorithms try to establish a hierarchy of clusters. This is either achieved bottom-up or top-down. In bottom-up algorithms such as Clink [5], the algorithm starts with each data point having its own cluster. Iteratively, the two closest clusters are merged until just a single one remains. In contrast, top-down algorithms such as DIANA [6] start with a single cluster and iteratively split each cluster into two to create the hierarchy. fuzzy c-means (FCM) clustering [7] is an example of a fuzzy theory-based algorithm. These algorithms use fuzzy cluster membership, so that a single data point may belong to multiple clusters and cluster membership is not binary but a continuous number between zero and one. FCM clustering otherwise works similarly to k-means clustering. The other clustering categories are distribution-based, density-based, graph theory-based, grid-based, fractal theory-based and model-based clustering [3]. When using a clustering technique as a precursor to record linkage, it is important to choose one that does not have a higher complexity than the record linkage itself. The aforementioned hierarchical clustering algorithm Clink, for example, has a time complexity of $O(n^2)$ [5], which is not an improvement in comparison to naive record linkage. On the other hand, clustering methods with a time complexity of $O(n)$, such as the hierarchical clustering algorithm BIRCH [8], may greatly increase the runtime of record linkage when used for blocking.

2.2.4 Privacy-Preserving Techniques

Privacy regulations such as the European Union's regulation on the protection of personal data or Health Insurance Portability and Accountability Act of the USA restrict disclosing sensitive personal data. These restrictions necessitate algorithms for record linkage that do not require the exchange of IDAT. To accommodate these requirements, privacy-preserving record linkage (PPRL) algorithms have been developed. Generally, these algorithms encode IDAT so that the encoded versions can still be compared but do not reveal information about the original data. This may be done via a hashing function such as MD5. However, with hashes, only a binary match can be obtained, which can only identify string equivalence, not similarity. The reason for this is that similarity measures between two differing hashes do not correlate with the similarity of the strings from which the hashes were derived. Therefore, to the author's knowledge there is no PPRL algorithm employing hashes only. Hashing-based algorithms can also fall prey to *dictionary attacks* - given a hash, an adversary can compute hashes for many different words until they find the word that produces a given hash. To remedy this,a hashed message authentication code (HMAC) may be used. Here, a secret key is exchanged between the parties involved in PPRL, and two padding strings *ipad* and *opad* of length B are defined. When a party needs to compute a hash, the following procedure takes place:

The secret key is padded with zeros to the length B. Two bitwise exclusive-OR (XOR) computations are carried out – between the padded secret key and either *ipad* or *opad*. Then, the string to be hashed is appended to the *ipad* XOR result, and the hashing function is applied to the resulting string. The *opad* XOR result is appended to the hash, and the hashing function is applied to this string to obtain the final result.

The final hash cannot be replicated without knowing the secret key, rendering dictionary attacks void. This technique was originally developed to authenticate messages, where the hash was computed from the original message [9].

An extension of the concept of hashes is using Bloom filters. Bloom filters were originally invented to easily test whether an item is a member of a set. A Bloom filter consists of a zero-initialised bit array of length n. For each item in the set, l hashes are computed using hash functions that have an output range of n values. For each computed hash h, the hth bit in the bit array is set to 1. To test an item for membership of the set, one can simply compute the l hashes of the item and see if the Bloom filter was set to 1 in all l positions given by the hashes. This method has a false negative error rate of 0%, though its false positive error rate is larger. Since Bloom filter-based PPRL algorithms also rely on hashes, HMACs are required to prevent dictionary attacks. Additionally, Bloom filters are weak to *frequency attacks*.

One PPRL algorithm employing Bloom filters was proposed by Schnell et al.

Chapter 3

Methods and Implementation

3.1 Mainzelliste SecureEpiLinker (MainSEL)

3.2 Comprehensive Secure Machine Learning Framework (CECILIA)

3.3 Implementation

3.4 Performance Considerations

3.5 Synthetic Data Generation

Chapter 4

Results

In this chapter which also could be more than one chapter, depending on the nature of the thesis, the results of the thesis are presented. Make sure you illustrate your results with appropriate figures and tables, but do not discuss the results here. This should be done in a separate discussion chapter.

4.1 Blocking

Chapter 5

Discussion and Outlook

Of course very important! You need to discuss the informatics as well as bio part of your thesis topic.

Take your time for writing the discussion, besides the introduction chapter it is the most important chapter of your thesis. Also do not subsection the discussion too heavily.

At least 5 pages.

Outlook can become an extra chapter.

Appendix A

Further Tables and Figures

Viele Arbeiten haben einen Appendix. Besondere Sorgfalt muss beim Nummerieren der Tabellen und Abbildungen gewährleistet sein.

Nummer	Datum
1	1.1.80
2	1.1.90

Table A.1: Erste Appendix-Tabelle

Nummer	Datum
1	1.1.80
2	1.1.90

Table A.2: Zweite Appendix-Tabelle

Bibliography

[1] Stacie B. Dusetzina, Seth Tyree, Anne-Marie Meyer, Adrian Meyer, Laura Green, and William R. Carpenter. An Overview of Record Linkage Methods. In *Linking Data for Health Services Research: A Framework and Instructional Guide [Internet]*. Agency for Healthcare Research and Quality (US), September 2014.

[2] Ivan P. Fellegi and Alan B. Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, December 1969.

[3] Dongkuan Xu and Yingjie Tian. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165–193, June 2015.

[4] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

[5] D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, January 1977.

[6] *6. Divisive Analysis (Program DIANA)*, pages 253–279. John Wiley & Sons, September 2009.

[7] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57, January 1973.

[8] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, pages 103–114, New York, NY, USA, June 1996. Association for Computing Machinery.

[9] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: Keyed-Hashing for Message Authentication. Request for Comments RFC 2104, Internet Engineering Task Force, February 1997.

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum

Unterschrift