dHealth 2020 – Biomedical Informatics for Health and Care
G. Schreier et al. (Eds.)

© 2020 The authors, AIT Austrian Institute of Technology and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/SHTI200070

Encoding of Numerical Data for Privacy-Preserving Record Linkage

Lea DEMELIUS[a,b,1], Karl KREINER[b], Dieter HAYN[b], Michael NITZLNADER[b] and Günter SCHREIER[a,b]
[a] *Graz University of Technology, Graz, Austria*
[b] *AIT Austrian Institute of Technology, Graz / Hall in Tyrol, Austria*

Abstract. Background: Privacy-preserving record linkage (PPRL) is the process of detecting dataset entries that refer to the same individual within two independent datasets, without disclosing any personal information. While applied in different fields, it particularly attained importance in the medical sector. One popular PPRL method are Bloom filters. However, Bloom filters were originally used for encoding strings only. Objectives: This paper evaluates an encoding method specifically designed for numerical data and adjusts it for encoding geocoordinates in Bloom filters. Methods: The proposed numerical encoding of geocoordinates is compared to the string-based method by using synthetic data. Results: The proposed method for encoding geocoordinates in Bloom filters attains a higher recall and precision than the conventional string encoding. Conclusion: Numerical encoding has the potential of increasing the record linkage quality of Bloom filters, as well as their privacy level.

Keywords. algorithms; confidentiality; medical record linkage; medical records system, computerized; personally identifiable information; privacy

1. Introduction

1.1. Privacy-preserving Record Linkage (PPRL)

Nowadays a great quantity of personal health-related data are collected in various data repositories such as hospitals, clinical trials and biobanks. The data collection is mostly conducted for primary use, meaning that it serves the direct care of patients or to clarify a specific research hypothesis. However, the massive accumulation of medical data can also be highly useful for research, as large-scale studies can be conducted without the need of collecting new data. This use beyond the initial intent of direct care is called secondary use [1].

To be able to use medical data for secondary use, it needs to be aggregated from several sources. It is, therefore, required to combine the datasets, i.e. to identify the same patient within these different data sources. This process is referred to as record linkage, data deduplication or duplicate detection [2]. The simplest record linking approach would be to link records using a unique identifier. However, there is no unique patient identifier throughout Europe which could be used for reliable patient identification [3].

[1] Corresponding Author: Lea Demelius, AIT Austrian Institute of Technology, Reininghausstraße 13, 8020 Graz, E-Mail: lea.demelius@ait.ac.at

Apart from that, even if a unique identifier would exist, it could not be used for secondary use due to privacy issues, as the data could not only be linked between different health data repositories, but also to other administrative data, e.g. legal or financial information. Consequently, non-unique identifiers, also referred to as quasi-identifiers or indirect identifiers, are used for record linking. These include name, date of birth, gender and place of birth [4]. Also, when using non-unique identifiers, it is necessary to comply with privacy regulations, such as the General Data Protection Regulation (GDPR) [5] in Europe. Therefore, the data used in the record linking process must be anonymized, or, with the consent of the patient, pseudonymized, to prevent unauthorized access to personal information. Anonymization refers to the complete deletion of all personally identifiable information, making it impossible to retrace the subjects. In pseudonymization the sensitive data are replaced by artificial identifiers called pseudonyms and re-identification is possible somehow. As anonymization interdicts any form of record linking, pseudonymization is applied in this field [1]. Record linkage without the disclosure of sensitive data is called privacy-preserving record linkage (PPRL).

1.2. EUPID

An example of a medical privacy-preserving record linkage system is the EUropean Patient IDentity (EUPID) Service [6] [7], which was developed to provide an identity management system for simplifying the secondary use of medical data, in particular in the context of rare diseases[2]. Collecting sufficient data for rare disease research is particularly challenging, often including several institutions. For this reason, record linking is applied to aggregate the data from different contexts, linking information referring to the same person.

EUPID currently uses three non-unique identifiers for PPRL: first name, last name, and date of birth. However, names are very unevenly distributed e.g. in the USA, where the top 50 first names make up 50% of the population and the top 1% of all last names 90% of the population [8]. In addition, the birthday paradox states that you only need 579 people to have a 99% certainty that someone has the same birth date[3] [9]. Consequently, common name combinations like John Smith are likely to share birth dates, and, henceforth, cannot be uniquely identified by only these three identifiers. Therefore, the place of birth is currently under consideration as an additional identifier for future EUPID versions in order to enhance the identification of individuals.

The performance of record linkage systems depends not only on the underlying identifiers, but also on the applied record linking method. Therefore, the record linkage quality with different PPRL approaches including Bloom filters is currently evaluated for the EUPID services.

1.3. Bloom filters

Bloom filters are bit arrays whose binary digits that are set to 1 encode one (or more) string(s). At first, the string to be encrypted is split into bigrams[4]. Next, k numbers of

[2] A disease is classified as rare if only a small percentage of patients is affected. In Europe the threshold lies at a prevalence of 0.05 percent [3].

[3] Presuming birth dates in the last 100 years.

[4] Bigrams are overlapping sequences of two neighboring characters.

hash functions are applied on each bigram. As proposed by Kirsch et al. [10] and also recommended by Schnell et al. [11], a double hashing strategy can be used to implement k hash functions as a weighted sum of two independent hashing algorithms, e.g. SHA1 and MD5. This is achieved with equation (1).

$$g_i(x) = \big(h_1(x) + ih_2(x)\big) \; mod \; L \tag{1}$$

h_1 corresponds to the first hash function (e.g. SHA1), h_2 to the second (e.g. MD5). The index i ranges from 0 to k-1, while L is the length of the Bloom filter. The modulus of L makes sure that the resulting hash value is not bigger than the length of the Bloom filter.

Next, the empty Bloom filter (an array of zeros with length L) is initiated and every position with index g_i is set to 1.

After encoding the strings, a similarity measure can be applied to the Bloom filters. Therefore, Schnell et al. [11] suggest using the Dice coefficient. The Dice coefficient D of two Bloom filters A and B is computed according to equation (2).

$$D_{A,B} = \frac{q \cdot h}{a + b} \tag{2}$$

When using bigrams q=2; h refers to the number of bits set to 1 in both Bloom filters; a and b represent the number of bits set to 1 on Bloom filter A and B, respectively. The resulting score has a value between 0 and 1. The higher this value is, the greater is the similarity between A and B.

1.4. Cryptographic Longterm Keys (CLKs)

Bloom filters can be used for field-based or record-level record linkage. Field-based means that every field is encoded into a separate Bloom filter e.g. one Bloom filter for the last name, one for the first name etc. Each encoded identifier is then compared separately. By contrast, a record-level Bloom filter first combines the values of all fields to a single string which is then encoded. Therefore, only one comparison per record has to be performed. Record-level Bloom filters were proposed by Schnell et al. [12] and they are known as Cryptographic Longterm Keys (CLKs). In respect to privacy, CLKs are superior to field-based Bloom filters, as it is harder to obtain, from which field the encoded information came from.

1.5. Encoding of numerical data

In record linkage, Bloom filters originally were introduced to encode strings. In the medical field, however, numerical data play an important role. In the context of EUPID, for example, numerical data may appear in the form of date of birth or place of birth.

The place of birth can be represented in different ways, e.g. country name, city name, district name, street name, geocoordinates or some combination of the first four. Using geocoordinates has several benefits compared to the string representations: First, geocoordinates are independent of language. Second, they are stable even if places are renamed or cities merged. Third, they allow distance measurements, which can be helpful e.g. when a place is specified with differing levels of precision (e.g. street vs. district vs. city).

Even though numerical values can be encoded in Bloom filters, simply by handling them as strings, this will not fulfil the purpose as e.g. numbers of similar values like 399

and 400 would not be classified as being close. Therefore, a Bloom filter encoding method is necessary that reflects this property of numerical data.

1.6. Objective of this paper

The objective of this paper was the application of the numerical Bloom filter encoding for similar patient matching proposed by Vatsalan et al. [13] to privacy-preserving record linkage and the adaption of this approach to geocoordinates.

2. Methods

2.1. Test data

Since real patient data were not available due to privacy reasons [14], synthetic data were created with an adapted version of a proprietary tool provided by AIT for generating realistic patient data for telehealth systems [15]. The resulting dataset included a unique ID, which was used for evaluating the record linkage process, gender, first and last name, date of birth, as well as longitude and latitude of the place of birth. The names were random combinations of first and last name typical for one of the following countries: Austria, Germany, France, Croatia, Iceland, Spain and the UK. For the date of birth, a day between 01.01.1900 and 31.12.2001 was randomly selected. The geocoordinates representing the place of birth were generated randomly within the boundaries of each country. Both longitude and latitude were floating point numbers with four decimal places.

For testing the record linkage performance, duplicates were generated by manipulating already existing entries. For names, these modifications included typical typing errors like (a) the abandonment or addition of accents, (b) randomly omitted characters, and (c) an added space at the beginning or end of strings. Moreover, they included (d) the addition of the title ‘Dr.’. Dates of birth were modified by (e) the swapping of day and month, or (f) an error of ± one day. The geocoordinates representing the place of birth were randomly changed within a given radius.

Moreover, similar non-matches were generated to test, if they result in false positives. These similar entries included patients with the same first and last name, patients with the same names as well as date of birth, and patients whose date of birth and place of birth were swapped (with changes to conform to the general data type).

The test dataset consists of 3000 entries, from which 300 are duplicates and 242 are similar non-matches.

2.2. Experimental setup

A Python framework was implemented based on the open source 'Python Record Linkage Toolkit' [16] to compare and evaluate CLKs.

2.2.1. Preprocessing

Before creating the CLKs, names were converted to lowercase letters, and prepended with a space, to recognize the greater significance of the first character while eliminating errors that come from inadvertent capitalization.

2.2.2. Encoding

Three different methods for encoding geocoordinates were tested and compared with each other. For all three, the other identifiers (first name, last name and date of birth) were encoded in a CLK by using the above-described method of hashed bigrams. For splitting the strings into bigrams, the 'ngrams' function of the Python-based 'Natural Language Toolkit' (NLTK) [17] was used. The geocoordinates were added to the CLK using three different approaches:

First (1), the geocoordinates were treated as strings and likewise encoded by the method of hashed bigrams. The second version (2) was like the first, except that the geocoordinates were shortened to two decimal places before encoding. Third (3), the geocoordinates were encoded using a method specifically established for numerical data. The used approach is based on the proposed methods for encoding floating point and modulus data by Vatsalan et al. [13]. Instead of treating the geocoordinate as a string and splitting it into bigrams, the whole value was hashed and included in the CLK. Moreover, several values, which lie in a predetermined range d_r around the original geocoordinate g, were generated and encoded as well (see Figure 1). To determine these additional values, the range d_r was split into 2b intervals, resulting in intervals with the width $d_{intv} = \frac{d_r}{2b}$. Therefore, the list of values $L[i] = x_i$ with length $2b + 1$ can be computed with equation (3).

$$x_i = g + (i - b) \cdot d_{intv} \tag{3}$$

As geocoordinates are floating point numbers, it has to be made sure that the lists of values overlap for close geocoordinates. Generally, this will not hold true as e.g. for b=1 and d_r=2 g=45.5 results in L[i]=[45.0 45.5 46.0] and g=45.6 in L[i]=[45.1 45.6 46.1], with no overlapping elements even though 45.5 is close to 45.6. The solution proposed by Vatsalan et al. [13] is the application of the following alignment (4).

$$new\_x_i = \begin{cases} x_i, & x_i \bmod d_{intv} = 0 \\ x_i - (x_i \bmod d_{intv}), & x_i \bmod d_{intv} < \frac{d_{intv}}{2} \\ x_i + (d_{intv} - (x_i \bmod d_{intv}), & x_i \bmod d_{intv} \geq \frac{d_{intv}}{2} \end{cases} \tag{4}$$

Moreover, geocoordinates have a limited range. For this case, Vatsalan et al. [13] present a calculation specification that handles modulus data, which jump back to the beginning of the range, that only include positive numbers (including zero) e.g. months [… 10 11 12 1 2 3 …]. Geocoordinates, however, include negative numbers, and while the longitude is modulus data, the latitude does not contain a leap but decreases/increases again when reaching the maximum/minimum of ±90° [… 89.9 89.9 90.0 89.9 …]. To comply with these characteristics the calculation specifications (5) and (6) were developed.

For latitude:

$$final\_x_i = \begin{cases} 2 \cdot max - new\_x_i, & new\_x_i > max \\ 2 \cdot min - new\_x_i, & new\_x_i < min \\ new\_x_i, & else \end{cases} \tag{5}$$

For longitude:

$$final\_x_i = \begin{cases} min + new\_x_i \bmod max, & new\_x_i > max \\ new\_x_i \bmod max, & new\_x_i < min \\ new\_x_i, & else \end{cases} \quad (6)$$

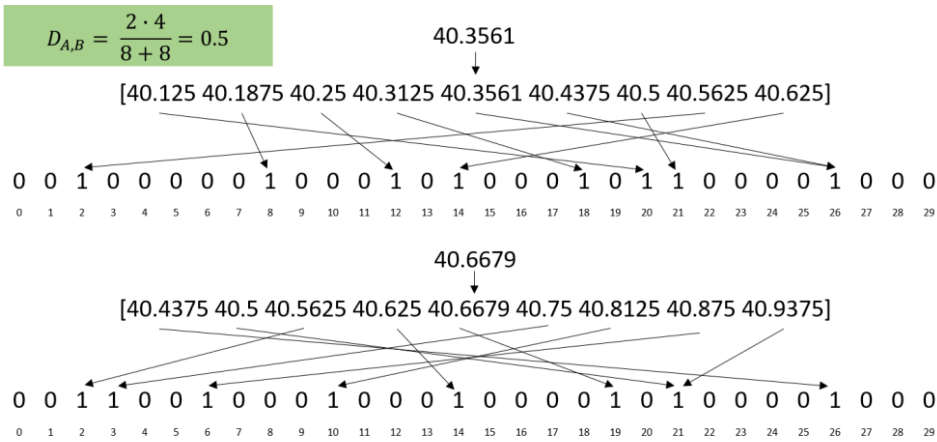

Figure 1. An example of numerical BF encoding of two similar geocoordinates when using b=4, L=30 and d_r=0.5. $D_{A,B}$ is the Dice coefficient of the two depicted Bloom filters.

2.2.3. Parameters

When using Bloom filters, several parameters must be chosen: First, all BFs have a fixed length L. Second, for encoding strings, the number of hash functions k with which every bigram is encrypted must be chosen. For encoding of numerical values according to the above-proposed scheme two more parameters need to be determined: b, which (just as k) regulates the number of created hashes, and the range d_r.

For all test runs, values of L=1,000 and k=10 were adopted from Schnell et al. [12]. To assure comparability between the different tested encodings, b was chosen so that the numerical encoding of the place of birth results in a similar number of hashes compared to the string encoding method. With k=10, this means that b=60. The range d_r was chosen according to empirical knowledge about the sizes of cities: If living in the same city, it is expected that the place of birth only deviates within $d_r = 0.5°$.

2.3. Classification and evaluation

As CLKs are a record-level comparison method, only one Dice coefficient per record pairs was computed which was then directly compared to a threshold value of 0.9.

Three key performance indicators were computed to evaluate the performance of the different encoding techniques: recall, precision and F1 measure.

3. Results

Table 1 lists the results of the three different tested encoding methods described in detail in section 2.2.2.

Table 1. Attained recall, precision, and F1 score for all three test runs with the parameters L=1,000, k=10, b=60, d_r=0.5 and a threshold value of t=0.9.

Test run No.	Test run description	Recall	Precision	F1 score
1	string encoding	0.863	0.977	0.917
2	string encoding with shortened geocoordinates	0.960	0.990	0.975
3	numerical encoding	0.997	0.997	0.997

4. Discussion

As can be observed in Table 1, the first test run resulted in the lowest recall as well as precision. Shortening the geocoordinates to two decimal points instead of four increased both recall and precision. The best result for all three key performance indicators, however, attained test run 3, which used the proposed numerical encoding method. Especially the recall was significantly higher for numerical encoding than string encoding, showing that more true matches could be identified as such. The slightly better precision of numerical encoding may be explained by the usage of different hashing schemes for date of birth and place of birth, which prevents 'overlapping' of these two identifiers. Contrary, in string encoding, a bigram of numbers will result in the same index, no matter if it came from the identifier date of birth or the place of birth.

Apart from inclining a higher precision, the usage of different hashing schemes also increases the privacy of CLKs: As all Bloom filters, CLKs are vulnerable to cryptoanalysis attacks. Even without the knowledge of the encoding parameters, it is possible to re-identify part of the sensitive information by exploiting frequency distributions [18]. Using different hashing schemes for different identifiers decreases exploitable frequency information as the same plaintext coming from two different fields results in a completely different 0-1-pattern. For further strengthening of privacy, also different hashing schemes for names and date of birth should be implemented.

Other methods recommended to prevent (or attenuate) cryptoanalysis attacks are the usage of record-level Bloom filters e.g. CLKs, which were implemented in this research, and advanced hardening techniques like balancing or BLoom-and-flIP (BLIP) [19]. To reach an even higher level of privacy, somewhat homomorphic encryption can be applied to Bloom filters [20]. The approach proposed by Randall et al. [20] is based on ring learning with errors and can prevent frequency attacks. However, further enquiries regarding the drawbacks of these different methods for privacy enhancement need to be made as hardening techniques may have some vulnerabilities, and homomorphic encryption comes with increased computational expense and the need of a fourth independent party.

Regarding the numerical encoding scheme proposed in this paper, further research should be conducted to show, if the results can also be obtained with real data.

References

[1] J. Heurix and T. Neubauer, "Privacy-preserving storage and access of medical data through pseudonymization and encryption," in *International Conference on Trust, Privacy and Security in Digital Business*, Springer, 2011, pp. 186-197.

[2] A. K. Elmagarmid, P. G. Ipeirotis and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Transactions on knowledge and data engineering,* vol. 19, no. 1, pp. 1-16.

[3] M. Maaroufi, P. Landais, C. Messiaen, M.-C. Jaulent and R. Choquet, "Federating patients identities: the case of rare diseases," *Orphanet journal of rare diseases,* vol. 13, no. 1, p. 199, 2018.

[4] S. Setoguchi, Y. Zhu, J. J. Jalbert, L. A. Williams and C.-Y. Chen, "Validity of deterministic record linkage using multiple indirect personal identifiers: linking a large registry to claims data," *Circulation: Cardiovascular Quality and Outcomes,* vol. 7, no. 3, pp. 475-480, 2014.

[5] Regulation, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46," *Official Journal of the European Union (OJ),* vol. 59, no. 1-88, p. 294, 27 April 2016.

[6] H. Ebner, D. Hayn, M. Falgenhauer, M. Nitzlnader, G. Schleiermacher, R. Haupt, G. Erminio, R. Defferrari, K. Mazzocco and J. Kohler, "Piloting the European Unified Patient Identity Management (EUPID) Concept to Facilitate Secondary Use of Neuroblastoma Data from Clinical Trials and Biobanking," *Studies in health technology and informatics,* vol. 223, pp. 31-38, 2016.

[7] M. Nitzlnader and G. Schreier, "Patient identity management for secondary use of biomedical research data in a distributed computing environment," *Stud Health Technol Inform,* vol. 198, pp. 211-8, 2014.

[8] J. R. Barr, C. Stephen and W. Zhao, "The Trouble with Names/Dates of Birth Combinations as Identifiers," 2011. [Online]. Available: https://www.idanalytics.com/media/The_Trouble_With-Names_White_Paper_FINAL.pdf. [Accessed 3 December 2019].

[9] S. Jones, "Same name, same birth date - how likely is it?," 28 September 2011. [Online]. Available: https://www.capgemini.com/2011/09/same-name-same-birth-date-how-likely-is-it/. [Accessed 3 December 2019].

[10] A. Kirsch and M. Mitzenmacher, "Less hashing, same performance: Building a better Bloom filter," *Random Structures & Algorithms,* vol. 33, no. 2, pp. 187-218, 2008.

[11] R. Schnell, T. Bachteler and J. Reiher, "Privacy-preserving record linkage using Bloom filters," *BMC medical informatics and decision making,* no. 1, p. 41, 2009.

[12] R. Schnell, T. Bachteler and J. Reiher, "A novel error-tolerant anonymous linking code," *German Record Linkage Center, Working Paper Series No. WP-GRLC-2011-02,* 16 November 2011.

[13] D. Vatsalan and P. Christen, "Privacy-preserving matching of similar patients," *Journal of biomedical informatics,* no. 59, pp. 285-298, 2016.

[14] "Febrl - Freely extensible biomedical record linkage," Australian National University, 1 July 2005. [Online]. Available: http://users.cecs.anu.edu.au/~Peter.Christen/Febrl/febrl-0.3/febrldoc-0.3/manual.html. [Accessed 27 June 2019].

[15] M. Drobics, K. Kreiner and H. Leopold, "Next Generation ICT Platform to Harmonize Medical, Care and Lifestyle Services," *Advances in Intelligent Systems and Computing,* pp. 275-283, 2016.

[16] "Python Record Linkage Toolkit Documentation," [Online]. Available: https://recordlinkage.readthedocs.io/en/latest/about.html. [Accessed 27 June 2019].

[17] E. Loper and S. Bird, "NLTK: the natural language toolkit," *arXiv preprint cs/0205028,* 2002.

[18] P. Christen, T. Ranbaduge, D. Vatsalan and R. Schnell, "Precise and fast cryptanalysis for Bloom filter based privacy-preserving record linkage," *IEEE Transactions on Knowledge and Data Engineering,* 2018.

[19] R. Schnell and C. Borgs, "Randomized response and balanced bloom filters for privacy preserving record linkage," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 2016, pp. 218-224.

[20] S. M. Randall, A. P. Brown, A. M. Ferrante, J. H. Boyd and J. B. Semmens, "Privacy preserving record linkage using homomorphic encryption," *Population Informatics for Big Data, Sydney, Australia,* vol. 15, 2015.