European Conference on Quality in Official Statistics (Q2016)
Madrid, 31 May-3 June 2016

Available Methods for Privacy Preserving Record Linkage on Census Scale Data

Rainer Schnell[1], Christian Borgs[2]

[1] *City University London, London, UK; Rainer.Schnell@city.ac.uk*
[2] *University of Duisburg-Essen, Duisburg, Germany; Christian.borgs@uni-due.de*

Abstract
The trend towards register based censuses is obvious in Europe. For these kind of censuses, record linkage across administrative data bases is essential. Without unique personal identifiers (PIDs) record linkage is laborsome, but routine. If the jurisdiction requires the use of encrypted identifiers for record-linkage, the task is challenging. The set of problems associated with this task has created an academic subfield called Privacy Preserving Record Linkage (PPRL). PPRL requires the use of advanced encryption techniques. Most often, standard encryptions such as MD5 or SHA-1 are applied on phonetic codes of identifiers. In real world settings, such codes miss many true links due to excessive errors in identifiers. Furthermore, missing identifiers usually require the use of hierarchical matching schemes. Therefore, new techniques such as embedding or Bloom Filters have been suggested. However, most encryption techniques can be attacked with cryptographic methods. All applications of PPRL have to demonstrate that their encryptions can resist at least a reasonable amount of cryptographic efforts. Based on recent cryptographic research, recommendations on the practical applications for PPRL will be given. Furthermore, in order to handle census scale datasets, special techniques for finding nearest approximate neighbors of encrypted records (blocking techniques) have to be used. The state of the art on blocking within PPRL will be described briefly.

Keywords: Administrative data, big data, record linkage, entity resolution, blocking.

1. Introduction

Register-based censuses are becoming more and more common in Europe (Valente, 2010). In countries where unique personal identifiers (PIDs) are not available, linking real-world entities across administrative data requires the use of identifiers such as names or birth dates (Abbott et al., 2016). Since these identifiers are prone to error, they can lead to non-linked pairs, which can in turn lead to biased estimates (Harron et al., 2014, Bohensky, 2016).

If the jurisdiction does not allow the use of unencrypted identifiers for record linkage, different linkage techniques are required allowing for errors in encrypted identifiers. This field of research is known as Privacy Preserving Record Linkage (PPRL, Vatsalan et al. (2013)). Many different approaches to PPRL have been suggested, for example, phonetic codes (Borst et al., 2001, Karmel et al., 2010), embedding (Scannapieco et al., 2007), reference lists (Pang and Hansen, 2006), secure multi-party protocols (Vaidya and Clifton, 2003) and Bloom Filters (Schnell et al., 2009). In current practice, only variants of phonetic codes and Bloom Filters are in use with census-scale data. Although phonetic codes are tried and tested for many years, especially in medical applications, simple implementations will miss many true links. Therefore, the application of phonetic codes usually requires a series of linkages using different encrypted identifiers (Abbott et al., 2016). The consequences of these procedures for inconsistent linkage decisions are hardly understood so far. Given this unsatisfactory state, research for alternative procedures is ongoing. Currently, the only alternative to phonetic encodings are Bloom Filter-based approaches. Therefore, we will concentrate on these. For a full overview of other relevant PPRL techniques, see Vatsalan et al. (2013).

2. Using Bloom Filters and Cryptographic Long-term Keys (CLKs) for encoding identifiers

Bloom (1970) suggested rapidly checking set membership by hashing keys to a bit vector. Its application for PPRL was suggested by Schnell et al. (2009). Currently, Bloom Filter-based encryptions are regularly used for private linking of personal data (Randall et al., 2013, Schnell et al., 2014, Schmidlin et al., 2015, Schnell and Borgs, 2015), to store ordinal data

(Vatsalan and Christen, 2016) and even for masking geographical data (Farrow, 2014, Farrow and Schnell, Under review).

Initially, a Bloom Filter is a bit array of the length l with all elements set to zero. To encrypt an identifier, the corresponding string is split into subsets of the length n. Most often, $n=2$ is used, these subsets are called bigrams. The bigrams are stored in a Bloom Filter by mapping the numeric representation of a linear combination of their SHA1- and MD5-strings to the Bloom Filter:

$$pos_{ngram} = h_{SHA1}(ngram) + i * h_{MD5}(ngram) \bmod l \qquad (1)$$

The resulting position in the Bloom Filter is set to one. This mapping is repeated k times. This kind of linear combination is the "double hashing" scheme proposed by Kirsch and Mitzenmacher (2006).

For practical applications, we recommend using $10 \leq k \leq 20$ hash functions for a Bloom Filter length of $l = 1000$, depending on the average amount of n-grams in the data (we use smaller values of k for higher average n-grams of all identifiers). Due to collisions, each n-gram sets up to k positions in the Bloom Filter to one. The resulting array of zeroes and ones is a Bloom Filter encryption of a single identifier. The desirable property of Bloom Filters is that they are similarity-preserving: an appropriate similarity function will find similar Bloom Filters despite small errors in the identifiers. The initial proposal for Bloom Filter-based Record Linkage used one separate Bloom Filter for each identifier. Schnell et al. (2011) showed that instead of separate Bloom Filters, one common Bloom Filter for all identifiers can be used. The resulting combined Bloom Filter is called a Cryptographic Long-term Key (CLK). In practice, CLKs seem to perform slightly inferior to separate Bloom Filters, but in many settings, CLKs are easier to apply. Furthermore, CLKs are harder to attack by simple frequency attacks than separate Bloom Filters. It should be kept in mind that all kinds of encryptions of identifiers (including phonetic keys) are prone to frequency attacks: a very common combination of personal identifiers will still be very common after an encryption. Therefore, encryptions of identifiers should employ the cryptographic principle of diffusion.

For practical considerations regarding such protective measures against attacks, see Schnell (2016).

In a first cryptographic study of Bloom Filter encryptions beyond frequency attacks of the entire bit pattern, Niedermeyer et al. (2014) showed that the double hashing scheme for Bloom Filters is vulnerable to cryptographic attacks of individual bit patterns resulting from bigrams. Recently, the double-hashing scheme has been successfully attacked within CLKs (Kroll and Steinmetzer, 2015). However, Niedermeyer et al. (2014) suggested to use full random hashing as a replacement of the double hashing scheme. For simulations, random hashing is implemented using a linear-congruential pseudo-random number generator (LCG, Stallings, (2014)) to generate a sequence *X*:

$$X_{n+1} = (a * X_n + c) \bmod l \qquad (2)$$

where *a* and *c* are carefully chosen constants. The sum of the index of the bigram in a bigram table and a secret cryptographic key is used as an initial value X_0. For each bigram, *k* subsequent numbers of the LCG are generated. For an actual implementation in practice, we would recommend using a cryptographic pseudo-random number generator instead of the LCG. Of course, a shared table of physically generated true random numbers could be used instead. Research on the cryptographic properties of CLKs is ongoing, but up to now, no successful attack on CLKs using random hashing is known.

3. Linking large databases with CLKs

While CLKs enable error-tolerant linkage through similarity measures, calculating the similarity is computationally expensive. If the full Cartesian product of two data sets has to be computed, the amount of computations grows too fast for practical applications with census-scale data. To reduce the number of comparisons, special techniques for finding nearest neighbors (*blocking*) have to be used.

3.1. Finding best matching CLKs

There are several methods for finding nearest neighbors in high dimensional binary space. In practice, most often Canopy Clustering (CC, McCallum et al. (2000)) and Sorted nearest neighborhood blocking (SNN, Hernandez and Stolfo (1998)) are used.

Although tree-based approaches are widely regarded as not suitable for finding nearest neighbors in binary space, Bachteler et al. (2013) suggested Multibit Trees for linking CLKs. Multibit Trees were introduced by Kristensen et al. (2010) for applications in chemoinformatics. During the search for nearest neighbors with Multibit Trees, the search is restricted to sub-trees with the same amount of bits set to one. For each cardinality of a query bit vector, we can estimate an upper similarity bound for all trees built from a second data set (Swamidass and Baldi, 2007). Only trees with an upper similarity bound higher than a given threshold will be considered for the calculation of the similarity. This reduces the search space considerably. Due to its fast computation, most often the Tanimoto similarity is used. A Tanimoto-coefficient is defined as the ratio of the bit positions set to one in both vectors A and B to the total number of bits set to one in A and B:

$$T(A,B) = \frac{\Sigma_i(A_i \wedge B_i)}{\Sigma_i(A_i \vee B_i)}. \qquad (3)$$

Lower thresholds will consider trees with lesser similarities, thereby increasing the search space and the risk of false positive classifications. Conversely, the amount of true matches will increase as well. For applications with CLKs, simulations indicate a Tanimoto-threshold of 0.85 to be optimal (Schnell, 2016).

The currently most promising approaches to find nearest neighboring CLKs (e.g. Canopy Clustering (CC), Sorted nearest neighborhood blocking (SNN) and Multibit Trees (MBT)) were compared by Schnell (2014). While the SNN blocking was shown to be faster than both other methods, its resulting linkage quality for CLKs was unacceptable. Furthermore, if the number of ones in a Bloom Filter (the Hamming weight) is identical for all CLKs, SNN

blocking will fail. The restriction of identical Hamming weights might be used to impede cryptographic attacks (Schnell et al., 2016).

While both CC and MBT showed virtually the same linkage quality, Multibit Trees outperform Canopy Clustering by large. In a recent comparison, a state-of-the-art parallel algorithm (PPJoin) was also outperformed by Multibit Trees (Sehili et al., 2015).

3.2 Quality and Speed of CLKs with Multibit Trees

CLKs in conjunction with Multibit Trees are able to achieve high-quality linkage results. Using Australian health data with standard CLKs and Multibit Trees, up to 97% of all possible matches were found, while keeping the amount of false positives under 5% (Brown et al., 2016). It should be noted, that no blocking was used at all. For example, by using date of birth as an external block or as a salt (Niedermeyer et al. 2014), precision can be improved.

The critical dependence of CLK performance on the similarity threshold is easily demonstrated by an example using German cancer registry data, shown in **Fig.** . Using a Tanimoto-threshold of 0.85 (as recommended earlier), we are able to achieve a F-Score of 0.975.

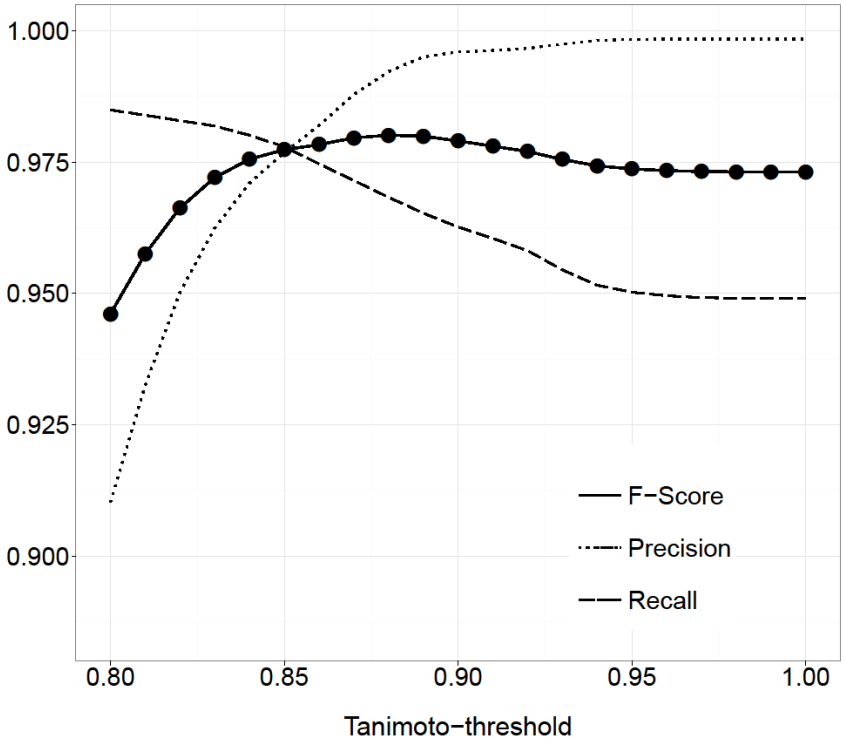

Fig. 1: Precision, Recall and F-Score for linking cancer registry data ($n_1 = 138131, n_2 = 73184$) with CLKs using k=10 hash functions and Multibit Trees by varying thresholds.

Multibit Trees are able to effectively link even very large amounts of data. Using a threshold of 0.85, files up to 10 million records can be linked without additional blocking in less than100 hours on standard hardware. External blocks for Multibit Trees on CLKs for census-sized data have been studied by Schnell (2014). For example, if the year of birth is encrypted (e.g. with SHA-256) and used to form blocks of CLKs in which the Multibit Trees are used separately, the maximum block size will be smaller than one million records for nearly all censuses. Using this approach, a full Census can be linked privately on a small server-cluster in less than a week.

4. Conclusion

Privacy Preserving Record Linkage, as shown here, works well despite errors in identifiers. False positive links can be further reduced by including additional identifiers, for example, place of birth or geographical distances between possible record pairs. However, simulations

have shown that the performance of Bloom Filter-based PPRL is dependent on the chosen parameters (Schmidlin et al., 2015). Therefore, we are currently developing an automatic method of choosing optimal parameters for Bloom Filter-based PPRL. Of course, exploring the cryptographic properties of CLKs to increase their resilience against cryptographic attacks is of primary importance for the wide-spread use of this PPRL approach. Finally, speeding up Multibit Trees by using GPUs and implementing these methods on Hadoop will be further steps in this research program.

References

Abbott, O., P. Jones and M. Ralphs (2016), Large-scale linkage for total populations in official statistics, Methodological Developments in Data Linkage, K. Harron, H. Goldstein and C. Dibben, Chichester, Wiley, pp. 170-200.

Bachteler, T., J. Reiher and R. Schnell (2013), Similarity Filtering with Multibit Trees for Record Linkage, Nuremberg, German Record Linkage Center.

Bloom, B. H. (1970), Space/time trade-offs in hash coding with allowable errors, Communications of the ACM, 13 (7), pp. 422-426.

Bohensky, M. (2016), Bias in data linkage studies, Methodological Developments in Data Linkage, K. Harron, H. Goldstein and C. Dibben, Chichester, Wiley, pp. 63-82.

Borst, F., F. A. Allaert and C. Quantin (2001), The Swiss solution for anonymous chaining patient files, Proceedings of the 10th World Congress on Medical Informatics: 2-5 September 2001; London, Amsterdam, IOS Press.

Brown, A., C. Borgs, S. Randall, R. Schnell (2016), High quality linkage using Multibit Trees for privacy-preserving blocking, IPDLN Conference 2016, accepted presentation.

Farrow, J. (2014), Privacy Preserving Distance-Comparable Geohashing, Internation Health Data Linkage Conference, Vancouver.

Farrow, J. and R. Schnell (Under review), Locational privacy preserving distance computations with intersecting sets of randomly labelled grid points, Journal of the Royal Statistical Society A.

Harron, K., A. Wade, R. Gilbert, B. Muller-Pebody and H. Goldstein (2014), Evaluating bias due to data linkage error in electronic healthcare records, BMC Medical Research Methodology, 14 (1), pp. 36.

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

Karmel, R., P. Anderson, D. Gibson, A. Peut, S. Duckett and Y. Wells (2010), Empirical aspects of record linkage across multiple data sets using statistical linkage keys: the experience of the PIAC cohort study, BMC Health Services Research, 10 (1), pp. 41.

Kirsch, A. and M. Mitzenmacher (2006), Less hashing same performance: building a better Bloom filter, Algorithms-ESA 2006, Proceedings of the 14th Annual European Symposium, Berlin, Springer.

Kristensen, T. G., J. Nielsen and C. N. S. Pedersen (2010), A Tree-based Method for the Rapid Screening of Chemical Fingerprints, Algorithms for Molecular Biology, 5 (1), pp. 9-20.

Kroll, M. and S. Steinmetzer (2015), Automated Cryptanalysis of Bloom Filter Encryptions of Health Records, 8th International Conference on Health Informatics.

McCallum, A., K. Nigam and L. H. Ungar (2000), Efficient clustering of high-dimensional data sets with application to reference matching, Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, ACM.

Niedermeyer, F., S. Steinmetzer, M. Kroll and R. Schnell (2014), Cryptanalysis of Basic Bloom Filters Used for Privacy Preserving Record Linkage, Journal of Privacy and Confidentiality, 6 (2), pp. 59-79.

Pang, C. and D. Hansen (2006), Improved record linkage for encrypted identifying data, Proceedings of HIC 2006 and HINZ 2006, Brunswick, Health Informatics Society of Australia.

Randall, S. M., A. M. Ferrante, J. H. Boyd, J. K. Bauer and J. B. Semmens (2013), Privacy-preserving Record Linkage on Large Real World Datasets, Journal of Biomedical Informatics, pp. 205-212.

Scannapieco, M., I. Figoti, E. Bertino and A. Elmagarmid (2007), Privacy preserving schema and data matching, Proceedings of the SIGMOD Conference, New York, ACM.

Schmidlin, K., K. M. Clough-Gorr and A. Spoerri (2015), Privacy Preserving Probabilistic Record Linkage (P3RL): a novel method for linking existing health-related data and maintaining participant confidentiality, BMC Medical Research Methodology, 16 (1), pp. 46-56.

Schnell, R. (2014), An efficient Privacy-Preserving Record Linkage Technique for Administrative Data and Censuses, Journal of the International Association for Official Statistics, 30 (3), pp. 263-270.

Schnell, R. (2016), Privacy Preserving Record Linkage, Methodological Developments in Data Linkage, K. Harron, H. Goldstein and C. Dibben, Chichester, Wiley, pp. 201-225.

Schnell, R., T. Bachteler and J. Reiher (2009), Privacy-preserving record linkage using Bloom filters, BMC Medical Informatics and Decision Making, 9 (41).

Schnell, R., T. Bachteler and J. Reiher (2011), A Novel Error-Tolerant Anonymous Linking Code, German Record Linkage Center.

Schnell, R. and C. Borgs (2015), Building a national perinatal database without the use of unique personal identifiers, 2015 IEEE 15th International Conference on Data Mining Workshops, Atlantic City, IEEE.

Schnell, R., A. Richter and C. Borgs (2014), Performance of different methods for privacy preserving record linkage with large scale medical data sets, 2014 International Health Data Linkage Conference: 28.04.-30.04.2014, Vancouver.

Schnell, R., M. Thürling and C. Borgs (2016), Explorations of Protective Measures against Cryptanalysis of Bloom filter-based Privacy-preserving Record Linkage, University of Duisburg-Essen.

Sehili, Z., L. Kolb, C. Borgs, R. Schnell and E. Rahm (2015), Privacy Preserving Record Linkage with PPJoin, Proceedings of the 16. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW), 2015, Hamburg, pp. 85-104.

Stallings, W. (2014), Cryptography and Network Security: Principals and Practice, 6th edition, New York, Pearson.

Swamidass, S. J. and P. Baldi (2007), Bounds and Algorithms for Fast Exact Searches of Chemical Fingerprints in Linear and Sublinear Time, Journal of Chemical Information and Modeling, 47 (2), pp. 302-317.

Vaidya, J. and C. Clifton (2003), Privacy-preserving k-means clustering over vertically partitioned data, Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, ACM.

Valente, P. (2010), Census taking in europe: how are populations counted in 2010?, Population and Societies, 467, pp. 1-4.

Vatsalan, D. and P. Christen (2016), Privacy-preserving matching of similar patients, Journal of Biomedical Informatics, 59, pp. 285-298.

Vatsalan, D., P. Christen and V. S. Verykios (2013), A Taxonomy of Privacy-preserving Record Linkage Techniques, Information Systems, 38 (6), pp. 946-969.