Open access Journal International Journal of Emerging Trends in Science and Technology

Impact Factor: 2.838 **DOI:** http://dx.doi.org/10.18535/ijetst/v3i04.10

# Privacy-Preserving Proof Association and Jamming For Huge Records with Cryptographic Keys Utilizing Multibit Trees

Authors
**S. Sridevi, P. Maylvahanan**
VELS University

**ABSTRACT**
*Increasingly, managerial data is being used for statistical purposes, for example registry based census charming. In follow, this generally wants connecting split files containing information on the same unit, without revealing the uniqueness of the part. If the connection has to be complete without a unique identification number, it is necessary to compare keys which are derived from unit identifiers and which are assumed to be similar. When dealing with large files like census data or population registries, comparing each possible pair of keys of two files is impossible. Therefore, special algorithms (jamming methods) have to be utilized to reduce the number of comparisons needed. If the identifiers have to be encrypted due to confidentiality concerns, the amount of available algorithms for blocking is very limited. This project describes the adoption of a newly launched algorithm for this problem and its performance for large files.*
**Key Words:** *Indexing, PPRL, Census, q-gram blocking, Bloom-Filter.*

## INTRODUCTION

Suitable to the growing accessibility of managerial information, linking different databases to determine the overlap of the records or to enhance the data available for a certain unit is a widely used strategy for statistical purposes. For illustration, of the 40 European censuses in 2011 only 22 were traditional censuses, while the rest was based on the relation of registries (Valente 2010). Linking different databases using a set of common identifiers is trivial if a unique personal identification number (PID) can be used. In some countries (for illustration, the Scandinavian nations) a PID is available for all members of the population. In practice, however, most statistical linkage operations are based on personal identifiers such as the name or date of birth. Such identifiers must be combined to yield an identification code. However, the identifiers are usually neither stable nor recorded with-out errors (Winkler 2009). Therefore, the use of exact matching identifiers will only link a non-randomly selected subset of the records. Hence, methods allowing for small variations of identifiers are to be used. Unfortunately, encryption of identifiers usually restricts linking to exact matching identifiers only. Hence, methods for linking with encrypted identifiers allowing for small errors in identifiers have to be used. Appropriate techniques are called "confidentiality preserving proof association methods". A method for confidentiality preserving documentation linkage which has newly become popular is the use of Bloom-Filters.

## USING BLOOM-FILTERS FOR ENCRYPTING IDENTIFIERS

In 2009, we recommended the apply of Bloom-Filters for cryptographic encrypting of identifiers (Schnell et al. 2009). Since then, this move toward has been used in different countries by different research groups and evaluates favorably to other advances (Vatsalan et al. 2013). The essential rule is splitting the string instead of each identifier (for

example the name) into a position of single subsets of length n (n-grams). For illustration, with n = 2 the bigram set of"PETER" is splits P; PE; ET; TE; ER; R. Each bigram of the set is mapped with k different cryptographic one-way hash tasks to a bit vector of duration l (a Bloom-Filter). As hash functions, keyed hash functions (HMACs), usually MD-5 and SHA-1, are utilized (for details on HMACs, see Martin 2012). Figure 1 shows a simple example of mapping names to Bloom-Filters with bigrams. In the instance, 8 identical bit positions are set to 1 in both Bloom-Filters. In total, 11 bits in A and 10 bits in B are set to 1. By the use of the Dice coefficient, the relationship of the two Bloom-Filters is (2 8)=(10 + 11) 0:762.

In general, the relationship between two strings can be approximated by using the Dice-coefficient of their Bloom-Filter. In practice, the utilize of larger Bloom-Filters (500 or 1000 bits) and more hash tasks (typically 15) has been found useful.

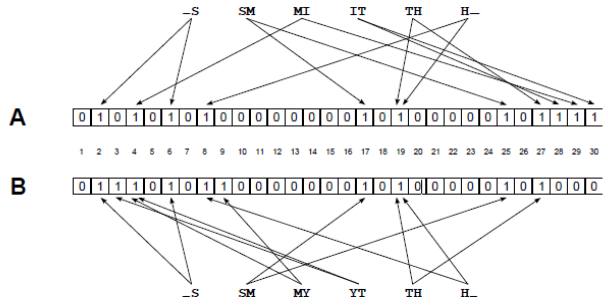


**Figure 1:** Example for the mapping of two names (SMITH, SMYTH) using bigrams and two hash functions to two Bloom-Filters (A, B) with 30 bits each.

In the initial proposal, each identifier was mapped to a separate Bloom-Filter. For the use of record linkage, each identifier encoded in a Bloom-Filter could be used for computing the similarity of two records. However, if a random sample of identifiers in the population is available to the attacker, a cryptographic attack on Bloom-Filters may be victorious for the majority frequent names (see Kuzu et al. 2013). Therefore, the security of separate Bloom-Filter encodings had to be enhanced.

**BLOOM-FILTER BASED SECRECY PRESERVING VERIFICATION ASSOCIATION: CRYPTOGRAPHIC LONG TERM KEYS (CLK)**

If a PID is not available, the number of possible identifiers is quite limited in most administrative databases. Some administrative databases contain unique identifiers. For example, birth registries usually have their own PID, hour of birth, minute of birth, series number in case of twins, birth weight and Apgar-Score. But in general, these special identifiers are not available in further records. Therefore, a key for connecting should be based on those identifiers general to all managerial databases. This set is, of course, specific to local regulations, but in general the basic set of identifiers (BSID) consists of the sure name, first name (at birth), date of birth, sex, and place of birth and country of birth. Additional identifiers are generally not given or even more volatile than those within the BSID. A cryptographic key based on BSIDs first requires the standardization of the identifiers (uppercase, transforming special characters, removing blanks and titles etc.). As a next step, the set of exceptional n-grams of each identifier is formed. Numerical data such as date of birth is also treated as series and discharged into n-grams. Generally, each element of date of birth (day, month, year) is used individually as one string variable and switched separately. Finally, this unique set of each identifier is mapped with a special amount of hash-functions and a special key to the same bit-array. The resulting binary vector (usually 500-1000 elements) is a cryptographic long-term key (CLK), which can be used for linking databases (Schnell et al. 2011). One benefit of CLKs is the reality to a given bit set to 1 may be due to different identifiers. This property makes attacks much more difficult than attacks on separate Bloom-Filters.

A further increase in the difficulty of attacks can be achieved by limiting the number of bigrams per

identifiers. This can be complete with special methods. In addition, random bits may be added. No successful attack on CLKs has been reported at this moment. Given the method CLKs are build, the kind of attacks used for Bloom-Filters will not work for CLKs. Therefore, the main difficulty for the utilize of CLKs in practical applications is the size of databases used for registry-based research.

## CONNECTING HUGE DATABASES WITH CLKs

Discovery two very similar CLKs may also be seen as the difficulty of finding nearest neighbors in a high dimensional binary gap. If we contain a database similar to the size of a census, we have to search the adjacent neighbor between more than 100 million applicants. Therefore, a direct comparison of similarity among all pairs of CLKs is virtually impossible. The number of contrasts has to be reduced to the range usually considered suitable for similarity calculations, for example cluster analysis. Hence, combinations of cases to smaller subsets are needed. Algorithms for producing these kind of combinations are called blocking methods.

## BLOCKING TECHNIQUES FOR CLKs

There are different blocking methods for reducing the amount of record pairs which need to be compared for the use in proof connection (Christen 2012). However, in regard to data structures related to CLKs, the choice of possible techniques is more limited. This project is limited to suitable candidates from the set of best performing techniques in the association study of Christen (2012).

## A NEW BLOCKING TECHNIQUE

In January 2013, I suggested the use of Multibit Trees for similarity filtering in record association in common without reference to confidentiality preserving record linkage (Bachteler et al. 2013). The method described in that project is named q-gram Blocking. The fundamental idea of q-gram Blocking is the transformation of all identifiers in a regular non confidentiality preserving record association problem to a bit array like a CLK as a initial step, tracked by the use of a Multibit Tree to find adjacent neighbors. The transformation to a CLK data structure allows the application of any method for finding nearest neighbors in high dimensional binary space to the problem of finding nearest neighbors to unencrypted insignificant data (or at least data delighted as nominal). Therefore, this approach can be used for blocking or similarity filtering in all record linkage applications.

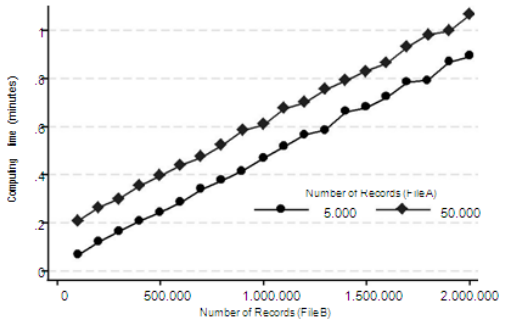


**Figure 2:** Computing time (in minutes) for finding 5.000 and 50.000 CLK records (File A) within a larger database (File B). Exact matches only.

Multibit Trees work in three steps. In the first step, the vectors of the larger file are grouped into bins, which are formed by the size of the vector (denoted by j~Bj), which here is the number of bits set to 1 in ~B. Therefore, all bins satisfying j~Bj t j~Aj or tj~B j j~Aj (1) can be ignored in the searching step, because min(j~B j;j~Aj)/max(j~Bj;j~Aj) constitutes an upper bound of SJ ( ~ A; ~B ).

## A MODEL STUDY OF CONNECTING WITH CLKs

We studied the performance of Multibit Trees in a series of simulations. In the initial publication of q-gram blocking (Bachteler et al. 2013), we reported comparisons inter alia between Multibit Trees, canopy clustering, sorted neighborhood and standard blocking. In most situations, Multibit Trees outperformed the methods performing best in other com-parison studies.

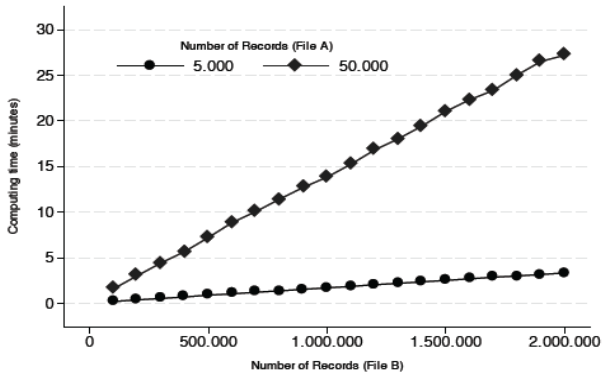


**Figure 3:** Computing time (in minutes) for finding 5.000 and 50.000 CLK records (File A) within a larger database (File B). Similarity threshold 0.95.

After the initial simulations had shown promising results, we implemented Multibit Trees as a R-library using C++. With this version, we observed a decrease in computing time by a factor of more than 5 (see figure 4).

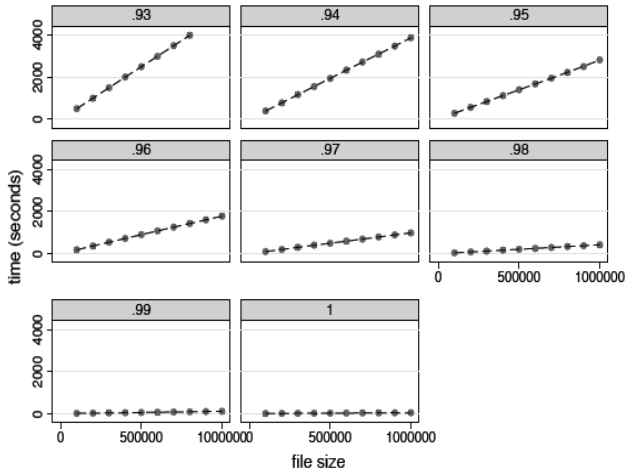


**Figure 4:** Query time in seconds for finding each nearest neighbor by a Multibit Tree (C++ version) in a file with 1 million records for varying file sizes (200.000 – 1.000.000) depending on the similarity threshold (.93 – 1).

For example, a comparison of two files with 1 million cases each was done without additional blocking. The tree was built within 48 seconds. The nearest neighbor was found in 40 seconds with exact matches. If a more reasonable similarity threshold of 0.95 is used the search takes about 48 minutes. Even with an unrealistic low similarity threshold of 0.9 the comparison takes just about 4 hours.

## CONCLUSION

Privacy preserving record linkage for very large files requires blocking methods to reduce the number of comparisons. Here, the use of Multibit Trees has been suggested. Different simulations on large datasets showed superior performance compared to previously used methods even for large datasets as used in practical applications. The suggested method shows a linear increase in computing time with increasing filesize. Therefore, for most statistical applications, the speed and accuracy of Multibit trees will be more than sufficient. Only for very large datasets such as a population census with CLKs additional techniques are required. The most simple option would be external blocking. For large scale problems like census operations, an obvious external block would be year of birth. If the year of birth is encrypted with an HMAC such as MD-5 or SHA-1, the resulting code for a year of birth would form a block and within each block CLKs with Multibit Trees could be used. Given just four machines with current standard hardware and current implementations of CLKs and Multibit Trees the privacy preserving record linkages for each European census can be done within 24 hours. So for the first time, the combination of CLKs, Multibit Trees and external blocking on year of birth would allow PPRL even with datasets as large as a European Census.

## FUTURE WORK

Performance of cryptanalysis based significantly lower than theoretical estimates. The future countermeasure makes resistant to known practical attacks.

## REFERENCES

1. Bachteler, T., Reiher, J., and Schnell, R. (2013), “Similarity Filtering with Multibit Trees for Record Linkage,” Working Paper WP-GRLC-2013-02, German Record Linkage Center, Nuremberg.
2. Christen, P. (2012), “A Survey of Indexing Techniques for Scalable Record Linkage

and Deduplication," IEEE Transactions on Knowledge and Data Engineering, 24, 1537–1555.

3. Hernandez, M. A. and Stolfo, S. S. (1998), "Real-world Data Is Dirty: Data Cleansing and the Merge/purge Problem," Data Mining and Knowledge Discovery, 2, 9–37.
4. Herzog, T. N., Scheuren, F. J., andWinkler,W. E. (2007), Data Quality and Record Linkage Techniques, New York: Springer.
5. Kristensen, T. G., Nielsen, J., and Pedersen, C. N. S. (2010), "A Tree-based Method for the Rapid Screening of Chemical Fingerprints," Algorithms for Molecular Biology, 5.
6. Kuzu, M., Kantarcioglu, M., Durham, E. A., Toth, C., and Malin, B. (2013), "A Practical Approach to Achieve Private Medical Record Linkage in Light of Public Resources," Journal of the American Medical Informatics Association, 20, 285–292.
7. Martin, K. M. (2012), Everyday Cryptography. Fundamental Principles and Applications, Oxford: Oxford University Press.
8. McCallum, A., Nigam, K., and Ungar, L. H. (2000), "Efficient Clustering of Highdimensional Data Sets with Application to Reference Matching," in Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 20–23 August 2000; Boston, eds. Ramakrishnan, R., Stolfo, S., Bayardo, R., and Parsa, I., New York: ACM, pp. 169–178.
9. Schnell, R., Bachteler, T., and Reiher, J. (2009), "Privacy-preserving Record Linkage Using Bloom Filters," BMC Medical Informatics and Decision Making, 9, 1–11.
10. (2011), "A Novel Error-Tolerant Anonymous Linking Code," Working Paper WPGRLC- 2011-02, German Record Linkage Center, Nuremberg