Schnell, R. (2016). Randomized Response and Balanced Bloom Filters for Privacy Preserving Record Linkage. Paper presented at the 16th IEEE International Conference on Data Mining Workshop, ICDMW 2016, 12-15 Dec 2016, Barcelona, Spain.

# Randomized Response and Balanced Bloom Filters for Privacy Preserving Record Linkage

Rainer Schnell
Centre for Comparative Social Surveys
City, University of London
London, United Kingdom
Email: Rainer.Schnell@city.ac.uk

Christian Borgs
German Record Linkage Center
University of Duisburg-Essen
Duisburg, Germany
Email: christian.borgs@uni-due.de

***Abstract*—In most European settings, record linkage across different institutions is based on *encrypted* personal identifiers – such as names, birthdays, or places of birth – to protect privacy. However, in practice up to 20% of the records may contain errors in identifiers. Thus, exact record linkage on encrypted identifiers usually results in the loss of large subsets of the data. Such losses usually imply biased statistical estimates since the causes of errors might be correlated with the variables of interest in many applications. Over the past 10 years, the field of Privacy Preserving Record Linkage (PPRL) has developed different techniques to link data without revealing the identity of the described entity. However, only few techniques are suitable for applied research with large data bases that include millions of records, which is typical for administrative or medical data bases. Bloom filters were found to be one successful technique for PPRL when large scale applications are concerned.**

**Yet, Bloom filters have been subject to cryptographic attacks. Previous research has shown that the straight application of Bloom filters has a non-zero re-identification risk. We present new results on recently developed techniques defying all known attacks on PPRL Bloom filters. The computationally inexpensive algorithms modify personal identifiers by combining different cryptographic techniques. The paper demonstrates these new algorithms and demonstrates their performance concerning precision, recall, and re-identification risk on large data bases.**

## I. Background

Linking information on the same micro-unit (persons, institutions, patents) across different data bases is an increasingly important task for administrative and research purposes. Applications can be found in census operations, the health sector, national security, crime detection and prevention [1], official statistics [2] and social science research [3]. However, most applications of record linkage involve information on natural persons. Under many jurisdictions (for example, most European countries), no unique personal identification number is available for Record Linkage. Therefore, Record Linkage has to use unstable and error-prone identifiers such as names or addresses.

Real-world identifiers may show error rates of more than 20% [4]. Examples of these errors include missing or additionally inserted letters, swapped letters, or completely missing attribute values [5].

In most legal settings, names and other personal identifiers have to be encrypted before data linkage across different data sets provided by independent data holders is permitted. This is especially true if potentially sensitive information (health information, criminal records, financial debts) is concerned.

Encrypting unreliable identifiers with standard cryptographic methods such as keyed HMACs (*Hash Message Authentication Code* [6] i.e. SHA-256 or MD5) would result in missing links. From a statistical point of view, unlinked records result in a missing data problem [7]. If a true link is missed, but the link is crucial for variables of interest, this is referred to as *differential linkage error* [8], [9]. Hence, a differential linkage error may result in biased estimates of causal effects and population parameters [10], [11]. The most straight-forward way to reduce differential linkage bias is improving the linkage rate. This problem has given rise to the field of *privacy preserving record linkage* (PPRL). Over the last decade, an increasing number of publications proposing novel PPRL methods have been published [12]. However, few of the proposed techniques are suitable for large scale linkage operations under the restrictions described above [13].

One method using Bloom filters [14] for error-tolerant privacy preserving record linkage [15] has been applied in several different research settings [16], [17],

Although the results on Bloom filter-based PPRL are promising, security concerns remain. So far, four studies by two research groups have been published on attacking PPRL Bloom filters [13]:

The first study [18] attacked basic Bloom filters with a constraint-satisfaction solver (CSS) to assign records to the frequency count given by a voter registration list. Although the technical details of the attack remain unclear it seems to be a variant of what is now being described as a simple rank swapping attack [19].

The second article [20] used composite Bloom filters with the same CSS attack. The authors consider their attack as hardly successful. However, both articles seem to show that basic Bloom filter encodings can be aligned with frequency distributions of unencoded identifiers. This way of attack is impossible if unique encodings can be achieved, for example by using salted encodings [21]. It should be noted, that the CSS attack is based on the entire Bloom filter, therefore it is no decoding, but an alignment. In contrast to that, [21], [22] attempt the decoding (actual revealing of all identifiers as clear text) by a cryptanalysis of individual bit patterns

within the Bloom filters. While [21] were successful with basic Bloom filters, [22] demonstrated partial success with composite Bloom filters. Details of these attacks will be given in section II-A.

To prevent their own attack, [21] suggested the use of different hash functions (random hashing), but this proposal has not been tested so far. Testing this proposal and suggesting two additional techniques for preventing attacks on Bloom filters is subject of this paper.

### A. Our contribution

Random hashing has been suggested by [21], but has not been tested in an attack. We report on applying the cryptanalysis method of [21] on random hashing and compare the results of this attack on two new techniques (Balanced Bloom Filters, BLIP/RAPPOR) suggested here for the first time for PPRL. We simulate the performance of these techniques by comparing them to the current standard practice of Bloom filter-based PPRL in terms of linkage quality and privacy metrics.

### B. Outline of the paper

Section II explains the construction of Bloom filters [15] and composite Bloom filters (CLKs [23]) based on the double hashing scheme [24]. Then, the only known attack on bit patterns within a Bloom filter [21] is described in section II-A.

The following section describes three methods to prevent this attack. The performance of these techniques is studied with regard to linkage quality in section II-D and with regard to privacy metrics in section II-G. We conclude by summing up the current recommendations for Bloom filter-based PPRL.

## II. Methods

Bloom filters storing one identifier as proposed by [15] have been subject to cryptographic attacks [18], [21]. To reduce the filter's vulnerability to cryptographic attacks [23] have proposed storing all identifiers in a single Bloom filter. These are then called Cryptographic Long-term Keys (CLKs). To build a CLK of all identifiers, each identifier is divided into $n$-grams. For instance, the last name `MILLER`, is split into bigrams and would thus yield a vector of $n$-grams containing `_M,MI,IL,LL,LE,ER,R_`. To store each $n$-gram in a Bloom filter (that initially consists only of zeroes) with the length $l$, the original CLKs used the double hashing scheme [24], where $k$ positions in the Bloom filter are set to a value of one. The individual bit positions $h_i$ are then determined by the sum of the integer representations of two different hash functions $f$ and $g$ of the $n$-gram (the original implementation used SHA-1 and MD5) which are mapped to the length of the Bloom filter $l$:

$$h_i = (f + i \cdot g) \mod l.$$

[21] have developed an attack on the resulting bit patterns for this particular scheme exploiting the circumstance that the amount of possible outcomes of the double hashing scheme is very limited. This attack is described in more detail in the following section.

### A. Bit pattern attack details

To attack CLKs, [21] only use $n = 2$-grams (bigrams) with additional attribute information appended (e.g. 'surname:`ER`' and 'name:`ER`' to differentiate between an '`ER`' as a part of a surname or part of a first name). $k = 20$ hash functions were used to map all bigrams to the Bloom filters with a length of $l = 1000$, applying the double hashing scheme described above. The resulting bit pattern of a single bigram is called an atom, which is set to one for up to $k$ bit positions.

Subsequently, a systematic search of all theoretically possible atom patterns is conducted. These patterns are limited, because either some of these patterns are impossible to achieve (as no bigram can be mapped) or they collide due to different $f$- and $g$-values. The patterns are used to build a matrix $D$, which contains the empirical relative frequencies of each atom in the masked data set (the CLK data). A second matrix $E$ consists of the relative frequencies of the bigram combinations over all identifiers from a non-encoded training data set, i.e. unencrypted clear-text data.

The Jakobsen-algorithm [25] is then used to minimize the distance between the matrices $D$ and $E$. The frequency analysis thus determines the correlation between known bigrams and unknown atoms. The algorithm provides a vector containing bigrams that are rearranged in the order of the atoms. In order to determine the maximal similarity assigning the attributes, a list of known names is then sorted in descending order of frequencies, converted into bigrams, and compared using the Dice-coefficient [26].

Finally, we compare the atoms found by the CLK (which are related to a known bigram by the Jakobsen-algorithm [25]) with the personal attributes of a reference list. The assigned value of personal attributes is considered as successfully decoded, if this value matches the personal attribute value of the plain text perfectly. Correspondingly, the decoding rate is defined as the percentage of assigned attribute values that precisely match.

### B. Methods to prevent attacks

*1) Random hashing:* To replace the double hashing scheme with random hashing, $k$ random numbers are drawn for every possible bigram. In comparison to double hashing with $k = 20$ hash functions approximately $10^{41}$ instead of $10^6$ different combinations are theoretically possible. Thus, we do not apply hash functions, as a systematic search requires considerably more computational power and time.

First, the universe of all possible n-grams is constructed separately for each identifier. For each possible n-gram, $k$ random numbers between 1 and the length of the Bloom filter $l$ are drawn with replacements by using a single password as a seed. For each n-gram, $k$ random positions are then set to one in the Bloom filter. This way, the hash functions are no longer required and a pattern-based attack will be much more difficult.

*2) Balanced Bloom Filters:* Since many attacks are based on the Hamming weight of a Bloom filter, data sets with Hamming weight distribution closer to a uniform distribution

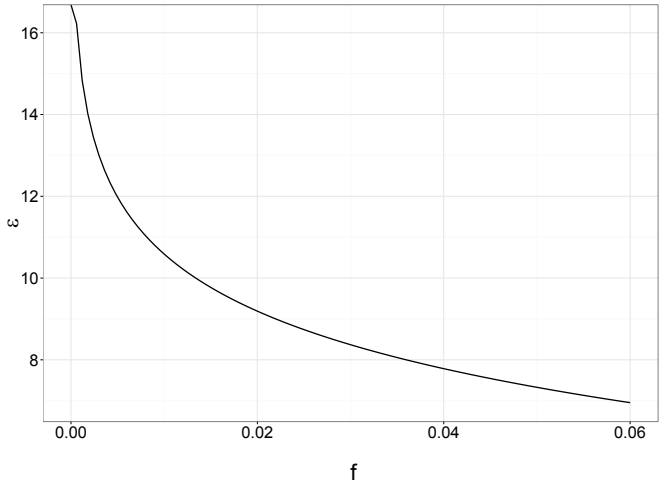


Fig. 1. Privacy level $\epsilon$ dependent on the bit flipping probability $f$. Lower values of $\epsilon$ denote a higher privacy level.

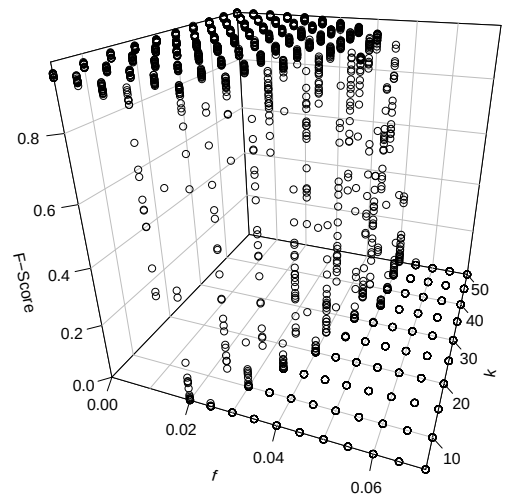


Fig. 2. F-Scores of BLIP CLKs dependent on bit-flipping probabilities $f$ and the number of hash functions $k$. $f$-values between 0.02 and 0.04 are unproblematic, depending on the number of hash functions used. More hash functions tolerate higher values of $f$.

are more difficult to attack than data sets with non-uniform distributions. Of course, data sets with constant Hamming weights for all Bloom filters would be optimal in this regard. Codes with constant Hamming weights are known as balanced codes [27], [28].

Therefore, we suggest the use of *Balanced Bloom filters* with constant Hamming weight for PPRL. Balanced Bloom filters can be constructed by concatenating a Bloom filter with length $l$ with a negated copy of the same Bloom filter. The resulting bit array of length $2 * l$ has to be permuted. This approach seems to prevent all attacks based on Hamming weights of Bloom filters.[1] It should be noted that the increased length of Balanced Bloom filters and their constant hamming weight increases computing time and – for some blocking methods – the required memory.

*3) Permanent Randomized Response CLKs (BLIP):* To increase the level of *differential privacy* [30], [31] and to cover the CLKs in a way that it is impossible for deterministic attacks to be carried out, we implemented the RAPPOR technique [32], which was first proposed for Bloom filters in [30] (as BLIP (for BLoom-and-flIP)). This allows us to use randomized responses in order to flip the values of random bit positions. The permanent randomized response satisfies $\epsilon_{\text{inf}}$-differential privacy [33] if

$$\epsilon_{\text{inf}} = 2k \ln \left( \frac{1 - \frac{1}{2}f}{\frac{1}{2}f} \right)$$

where $k$ is the number of hash functions and $f$ is the probability of switching one bit (see Figure 1).

To implement the randomized response with a Bloom filter of the length $l$, each bit position $B_i$ is treated with a randomized response, which then provides us with the new bit value $B_i^{'}$:

$$B_i^{'} = \begin{cases} 1 & \text{with probability} \quad \frac{1}{2}f \\ 0 & \text{with probability} \quad \frac{1}{2}f \\ B_i & \text{with probability} \quad 1 - f \end{cases} .$$

This way, the new parameter $f$ is the chance of flipping a bit at each bit position of the Bloom filter. To simulate the effect of BLIP on CLKs, we tested $f$-values ranging from 0.01 to 0.2 using $k$ different numbers of hash functions. Figure 2 shows the F-Scores for several different PRR parameters we have tested. For all following calculations, $k = 20$ hash functions and a flipping-probability of $f = 0.02$ were applied. Note that $f = 0.02$ does not guarantee a sufficient level of differential privacy (see Figure 1) [30]. Still, deterministic attacks are countervailed, while the privacy levels are approximately increased by a factor of two (since $\epsilon$ is nearly halved).

*4) Balanced BLIP:* For analysis, we also combined BLIP with balanced codes by applying the balanced code first and using the randomized response on the resulting balanced CLK next. We used $f = 0.02$ as for the BLIP CLKs.

### C. Data

The data set used to assess the quality measures of the record linkage (see section II-F) was generated by independently sampling surnames, names, and sex from a large administrative data base. Dates of birth were sampled uniformly. An error generator was used on a copy of the resulting data file with $n = 10.000$ records, creating at least one error (swapping,

[1] If the Hamming weights of column of the resulting data set of Balanced Bloom filters vary and all rows are identically permuted, a reversal of the balancing might be possible by finding unique pairs of columns (this idea is due to [29]). To make the success of this attack highly unlikely, a different permutation based on an error-free identifier should be used.

replacement, inserts, deletions) in approximately 20% of the rows. The error-free and erroneous files were used to create different alternative Bloom filters, which were finally linked using Multibit trees (see section II-D).

For the bit pattern attack and the calculation of the privacy metrics, we used a second data set. Training data including $n = 1.000.000$ entries containing surnames, first names, dates of birth, and birthplaces was created with an attribute frequency distribution that was similar to a large administrative data base. For privacy reasons, the attributes of the administrative data base were stored separately and contained only the absolute frequencies. Approximately 680.000 different first names, 400.000 different surnames, and 10.400 different locations were sampled independently from the population data base. The resulting sample therefore approximates the population distribution closely. This close approximation is necessary for the bit pattern attack, as the quality of the training data is important for decoding. However, in a real-life attack, the assumption of very similar frequency distributions in both data sets is unlikely to hold.

### D. Linkage

The two data files described in Section II-C were encrypted using the five methods mentioned in the previous section (Standard CLK, CLK with random hashing, BLIP, Balanced CLK, and Balanced BLIP). The resulting Bloom filters were linked using Multibit trees [34]. Multibit trees were used to construct "leaves". This was achieved by finding those bit positions, where approximately half of the records' bits were set to one and the other half of the records' bits were set to zero. This process was repeated until only a few records in each leaf were left – we limited this to three.

Using the information of the match bits, a maximum Tanimoto-similarity between all leaves of a Tree and a candidate Bloom filter could be estimated before we computed similarity. Following this strategy allowed us to exclude a large number of records from the search space. The threshold for the lowest possible Tanimoto-similarity is user-defined. The Tanimoto coefficient $T$ is a similarity measure for bit vectors, which is defined as

$$T(A, B) = \frac{A \wedge B}{A \vee B}$$

for two bit vectors $A$ and $B$. A value of $T = 1$ represents perfectly matching vectors. Lowering the admitted minimal similarity threshold allows tolerating more errors between two vectors, but it may lead to an increase in false positive classifications (see II-F).

### E. Implementation details

This paper analyses the decoding rate of the bit pattern attack, different privacy measures, and the quality of record linkage. Privacy measures were assessed using R 3.3.0, all encoding variants were handled using an R-Package (PPRL) developed by our research group. The attack was implemented in Python 2.7 and C++. Linkage was done with R using a Multibit-tree.

### F. Linkage quality measures

To assess the linkage quality, the standard record linkage criteria (precision, recall, and F-score) were used. Recall is defined as the number of true positive matches divided by the number of factual pairs, including pairs that were falsely classified as non-matches (false negatives *fn*):

$$\text{Recall} = \frac{tp}{tp + fn}.$$

The higher the recall, the better are record pairs found by a given linkage procedure. Precision is defined as the number of correctly classified pairs (true positive classifications $tp$) divided by the number of all classified pairs (*tp* and false positives *fp*):

$$\text{Precision} = \frac{tp}{tp + fp}.$$

The higher the precision, the less likely is the false classification of potential pairs as matches. Finally, F-score is defined as the harmonic mean of recall and precision:

$$\text{F-score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}.$$

All measures range from zero to one.

### G. Privacy Metrics

We evaluated various measures of privacy, calculated mutual information (MI) and entropy as well as the probability of suspicion (Ps) metrics [35]. Additional information, such as entropy and the number of unique bit patterns, were also included. In contrast to [36], we calculated mutual information on logarithms base 2 instead of base 10, so the units of entropy are bits. Encoded plain texts were used as the masked data set. Our training data set was used as the global (reference, in encoded form) data set. Both data sources are described in section II-C.

*1) Mutual information and relative information gain:* Mutual information (MI) was computed as

$$MI = (H(X) + H(Y)) - H(X, Y)$$

where $H(X)$ is the entropy of the clear text variables and $H(Y)$ is the entropy of the corresponding encrypted field (the CLKs). These are calculated as

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i)$$

and

$$H(Y) = -\sum_{i=1}^{m} P(y_j) \log P(y_j).$$

Furthermore, the entropy of the encrypted field (i.e. CLK entropy) together with the file size of the masked data set $n_m$ are used to calculate the mean entropy (ME) for each variation of the encryption:

$$ME = \frac{1}{n_m} H(Y).$$

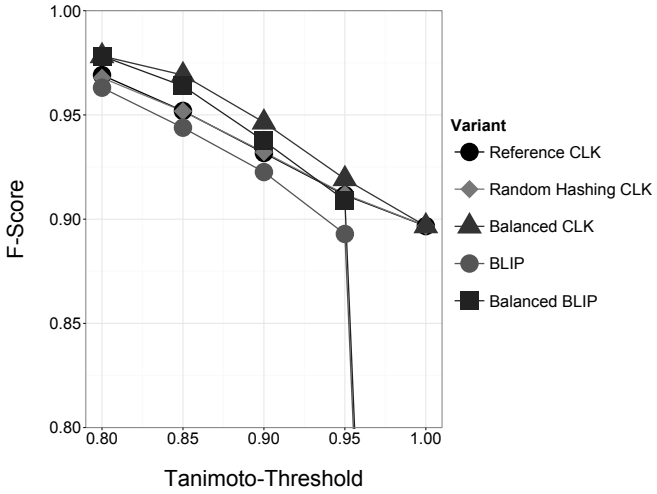


Fig. 3. F-Scores for different encodings dependent on Tanimoto-thresholds between 0.8 and 1.0.

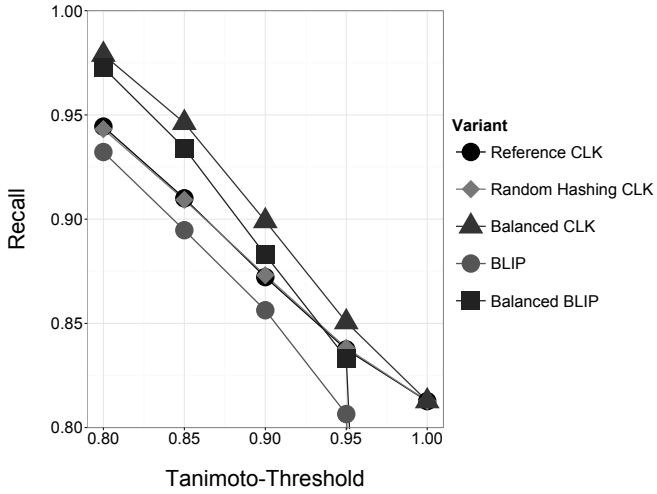


Fig. 4. Recall for different encodings dependent on Tanimoto-thresholds between 0.8 and 1.0.

For the calculation of the mutual information (MI), the joint entropy was computed by

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j)\ \log P(x_i, y_j).$$

Based on the mutual information criterion, the relative information gain (RIG) was calculated by

$$RIG = \frac{MI}{H(X)}.$$

*2) Probability of suspicion:* To calculate the probability of suspicion ($Ps$) [35], we had to classify the result of the linkage of the encrypted global and masked data set first. As no true pair shared the same IDs, true positive matches could be classified easily: if the IDs of the positively classified pairs matched, a true positive pair (TP) was identified. If the pairs did not match, a false positive (FP) was recorded. The difference between the true positives and the actual number of pairs then represents the number of mismatched pairs (false negatives, FN).

With the known number of true positives (TP) and the file size of the masked data ($n_m$), the probability of suspicion was calculated for each record as

$$Ps = \frac{1}{TP} - \frac{\frac{1}{n_m}}{(1 - \frac{1}{n_m})}.$$

## III. Results

Figures 3, 4 and 5 show F-Scores, recall, and precision for each variation of the encryption.

Changing the encryption scheme from double hashing to random hashing showed no change in any of the linkage quality measures. Introducing BLIP with a bit-flipping probability of $f = 0.02$ slightly reduced the F-Score for the standard CLK and the CLK with balanced codes. In addition, balancing the CLKs displayed a marginal increase in linkage quality as a result of improved recall measures.

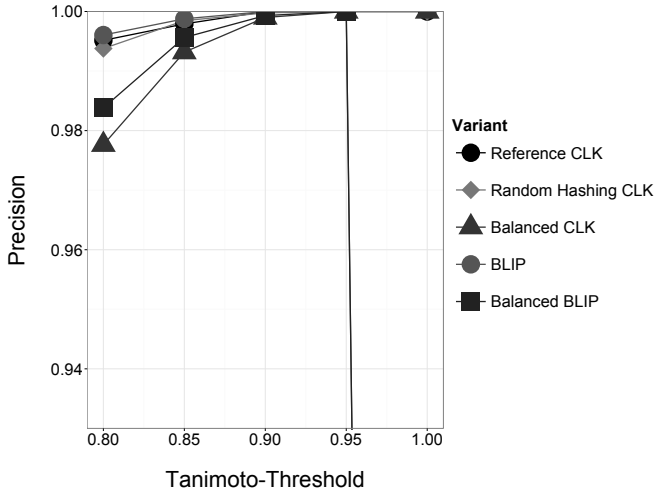


Fig. 5. Precision for different encodings dependent on Tanimoto-thresholds between 0.8 and 1.0.

The BLIP results are in line with the results from adding random bits to Bloom filters [13]. However, using bit flipping changed both zeroes and ones with a probability of $\frac{f}{2}$, while random bits only added a limited number of ones. It has to be noted that all BLIP CLKs show zero true positive matches (consequently, the F-scores, precision, and recall measures have a value of zero) when restricting the Multibit trees to a Tanimoto-similarity of one. However, this is not surprising, as a Tanimoto-similarity requires an exact match, which is highly unlikely when flipping random bits of each record with a probability of $f = 0.02$.

On the basis of the suggested Tanimoto-threshold of 0.85 [13], all proposed modifications yielded similar results as the reference CLK (double hashing scheme, $k = 20$ hash functions, $l = 1000$). Any of the methods tested here detected 90-95% of all true record pairs with F-scores between 0.947 to 0.969.

Table I presents the decoding rates for each variation using the attack by [22], privacy metrics, entropy, and number of unique bit patterns in the data.

The results show that the published attack fails when introducing random hashing. The decoding rate dropped to zero for all methods except for the reference double hashing CLK. The attack is unsuccessful when the number of atoms is too low ($< 300$) to deduce bigrams from the bit patterns.

All privacy metrics (*Ps, RIG, MI*) are constant for all methods. This is due to the fact that all bit patterns are unique in the CLK encrypted data. The linkage attack yielded no feasible results, with a probability of suspicion ($Ps$) of zero. Finally, the metrics were unable to predict the decoding rate using the bit pattern attack.

## IV. Discussion

The findings reported demonstrate that it is possible to achieve good results with Privacy Preserving Record Linkage even under very strict privacy jurisdictions. Neither precision nor recall suffer substantially when any of the proposed techniques are applied.

Frequency attacks require very frequent combinations of identifiers that are not observed in our sample of a hundred thousand records. For most applications, this number of records is not exceeded at all or least not within a block formed by identifiers used for salting. Since the use of CLKs alone produces unique patterns for each unique combination of identifiers, in such settings frequency attacks are impossible on entire bit patterns (CLKs). In such situations, the privacy metrics suggested by the PPRL literature [35], such as MI or $Ps$, are useless as they always result in constant values.

Currently, the only known attack remaining [21] is the identification of bit patterns within a Bloom filter. Applying the decoding algorithms described by Niedermeyer [21] and modified by [22], yielded no successful decoding of bit patterns for any of the newly suggested encoding methods described here. Although random hashing by itself prevents the Niedermeyer attack, a combination of random hashing with balancing codes prohibits any attack based on Hamming weights, including attacks that are – to date – unknown. BLIP or RAPPOR, respectively, is intended to guarantee differential privacy. However, the probability of bit flipping required for conventional privacy levels is too high to be applied to CLKs in order to successfully link records. Therefore, BLIP with lower probabilities of bit flipping should be considered as a variant of random bits as suggested by [15] and discussed by [13]. BLIPs advantage is RRT masking of the bit by either reporting 0 or 1, instead of randomly inserting 1s. BLIP should make deterministic attacks on frequent sub-patterns harder and increases the number of unique full bit patterns. By increasing the number of unique patterns, any frequency attack requires larger data bases or block sizes to be successful. Using a data set with $n = 100.000$ records, we have demonstrated that combining BLIP and balancing codes with random hashing prevents all known attacks.

## V. Conclusion

To sum up the current state of Bloom filter-based PPRL, and considering the results reported here, we recommend:

1) The use of as any many [37] stable [38] identifiers as available,
2) avoiding the use of padding [13],
3) limiting the length of identifiers [13],
4) using random hashing instead of double hashing,
5) using balanced and RRT Bloom filters,
6) using a stable identifier for salting [21],
7) linking Bloom filters using Multibit trees [39], [40].

Studying the impact of these modifications in real-world settings is subject of ongoing research of our group. Furthermore, we are implementing all Bloom filter-based techniques in an upcoming R-library.


## Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments.

This study was supported by the research grant SCHN 586/17-2 of the German Research Foundation (DFG) to the first author.


## References


[1] P. Christen, *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Berlin: Springer, 2012.

[2] O. Abbott, P. Jones, and M. Ralphs, "Large scale linkage for total populations in official statistics," in *Methodological Developments in Data Linkage*, K. Harron, H. Goldstein, and C. Dibben, Eds. Chichester: Wiley, 2016, pp. 170–200.

[3] R. Schnell, "Linking surveys and administrative data," in *Improving Survey Methods*, U. Enge, B. Jann, P. Lynn, A. Scherpenzeel, and P. Sturgis, Eds. New York, London: Routledge, 2015, pp. 273–287.

[4] W. E. Winkler, "Record linkage," in *Handbook of Statistics 29A, Sample Surveys: Design, Methods and Applications*, D. Pfeffermann and C. Rao, Eds. Amsterdam: Elsevier, North-Holland, 2009, pp. 351–380.

[5] P. Christen, "Febrl: a freely available record linkage system with a graphical user interface," in *HDKM '08 Proceedings of the second Australasian workshop on Health data and knowledge management*. Darlinghurst, Australia: Australian Computer Society, 2008, pp. 17–25.

[6] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 6th ed. New Jersey: Pearson, 2014.

[7] X.-H. Zhou, C. Zhou, D. Liu, and X. Ding, *Applied Missing Data Analysis in the Health Sciences*. Hoboken: Wiley, 2014.

[8] J. K. Leiss, D. Giles, K. M. Sullivan, R. Mathews, G. Sentelle, and K. M. Tomashek, "U.S. maternally linked birth records may be biased for Hispanics and other population groups," *Annals of Epidemiology*, vol. 20, no. 1, pp. 23–31, Jan. 2010.

[9] J. T. Lariscy, "Differential record linkage by Hispanic ethnicity and age in linked mortality studies: implications for the epidemiologic paradox," *Journal of Aging and Health*, vol. 23, no. 8, pp. 1263–1284, Dec. 2011.

[10] I. Baldi, A. Ponti, R. Zanetti, G. Ciccone, F. Merletti, and D. Gregori, "The impact of record-linkage bias in the Cox model," *Journal of Evaluation in Clinical Practice*, vol. 16, no. 1, pp. 92–96, 2010.

[11] K. Harron, A. Wade, R. Gilbert, B. Muller-Pebody, and H. Goldstein, "Evaluating bias due to data linkage error in electronic healthcare records," *BMC Medical Research Methodology*, vol. 14, no. 1, p. 36, 2014.

[12] D. Vatsalan, P. Christen, and V. S. Verykios, "A taxonomy of privacy-preserving record linkage techniques," *Information Systems*, vol. 38, no. 6, pp. 946–969, 2013.

[13] R. Schnell, "Privacy preserving record linkage," in *Methodological Developments in Data Linkage*, K. Harron, H. Goldstein, and C. Dibben, Eds. Chichester: Wiley, 2016, pp. 201–225.

TABLE I
CENTRAL MEASURES OF PRIVACY FOR THE ENCRYPTION METHODS PRESENTED.

| | *Decoding rates* | | | | | *Privacy metrics* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Coding method | Surname | Given name | Date of birth | Atoms found | Total decoding rate (%) | $Ps$ | $RIG$ | MI | Entropy | Unique patterns (%) |
| Reference CLK | 70.3 | 67.4 | 42.1 | 2539 | 18.7 | 0 | 1 | 6 | 0.991 | 100 |
| Random Hashing CLK | 0.7 | 0.4 | 3.6 | 189 | 0.0 | 0 | 1 | 6 | 0.991 | 100 |
| Balanced CLK | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0 | 1 | 6 | 1.000 | 100 |
| BLIP | 0.0 | 0.0 | 0.0 | 16 | 0.0 | 0 | 1 | 6 | 0.999 | 100 |
| Balanced BLIP | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0 | 1 | 6 | 0.999 | 100 |

Reference CLKs using $k = 20$ hash functions and the double hashing scheme. Columns show the decoding rate of the Niedermeyer attack for the identifiers surname, given name and date of birth. Total decoding rate is the percentage of records where all identifiers are correctly decoded. Privacy metrics were discussed in section II-G.


[14] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[15] R. Schnell, T. Bachteler, and J. Reiher, "Privacy-preserving record linkage using Bloom filters," *BMC Medical Informatics and Decision Making*, vol. 9, no. 41, 2009.

[16] E. Durham, Y. Xue, M. Kantarcioglu, and B. Malin, "Private medical record linkage with approximate matching," in *Proceedings of the 2010 American Medical Informatics Association Annual Symposium*, 2010, pp. 182–186.

[17] C. E. Kühni, C. S. Rueegg, G. Michel, C. E. Rebholz, M.-P. F. Strippoli, F. K. Niggli, M. Egger, and N. X. von der Weid, "Cohort profile: the swiss childhood cancer survivor study," *International Journal of Epidemiology*, pp. 1–12, 2011.

[18] M. Kuzu, M. Kantarcioglu, E. Durham, and B. Malin, "A constraint satisfaction cryptanalysis of Bloom filters in private record linkage," in *Privacy Enhancing Technologies, 11th International Symposium, PETS 2011*. Waterloo, Canada: Springer Berlin Heidelberg, 2011, pp. 226–245.

[19] J. Domingo-Ferrer and K. Muralidhar, "New directions in anonymization: Permutation paradigm, verifiability by subjects and intruders, transparency to users," *Information Sciences*, vol. 337, no. 338, pp. 11–24, 2016.

[20] M. Kuzu, M. Kantarcioglu, E. A. Durham, C. Toth, and B. Malin, "A practical approach to achieve private medical record linkage in light of public resources," *Journal of the American Medical Informatics Association*, vol. 20, no. 2, pp. 285–292, 2013.

[21] F. Niedermeyer, S. Steinmetzer, M. Kroll, and R. Schnell, "Cryptanalysis of basic bloom filters used for privacy preserving record linkage," *Journal of Privacy and Confidentiality*, vol. 6, no. 2, pp. 59–79, 2014.

[22] M. Kroll and S. Steinmetzer, "Automated cryptanalysis of bloom filter encryptions of health records," in *8th International Conference on Health Informatics*, 2015.

[23] R. Schnell, T. Bachteler, and J. Reiher, "A novel error-tolerant anonymous linking code," German Record Linkage Center, Duisburg, Working Paper WP-GRLC-2011-02, 2011.

[24] A. Kirsch and M. Mitzenmacher, "Less hashing same performance: building a better Bloom filter," in *Algorithms-ESA 2006. Proceedings of the 14th Annual European Symposium: 11-13 September 2006; Zürich, Switzerland*, Y. Azar and T. Erlebach, Eds. Berlin: Springer, 2006, pp. 456–467.

[25] T. Jakobsen, "A fast method for the cryptanalysis of substitution ciphers," *Cryptologia*, vol. 19, no. 3, pp. 265–274, 1995.

[26] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[27] D. E. Knuth, "Efficient balanced codes," *IEEE Transactions on Information Theory*, vol. IT-32, no. 1, pp. 51–53, 1986.

[28] J. M. Berger, "A note on error detection codes for asymmetric channels," *Information and Control*, vol. 4, pp. 68–73, 1961.

[29] F. Niedermeyer, "Analysis of Bloom filters with Constant Hamming Weight," personal communication to the authors, 2016.

[30] M. Alaggan, S. Gambs, and A.-M. Kermarrec, "BLIP: Non-interactive differentially-private similarity computation on bloom filters," in *Stabilization, Safety, and Security of Distributed Systems: 14th International Symposium, SSS 2012, Toronto, Canada, October 1–4, 2012. Proceedings*, A. W. Richa and C. Scheideler, Eds. Berlin: Springer, 2012, pp. 202–216.

[31] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: Simultaneously solving how and what," in *CRYPTO 2008, LNCS 5157.*, D. Wagner, Ed. International Association for Cryptologic Research, 2008, pp. 451–468.

[32] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries," *Computing Research Repository*, 2015.

[33] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014.*, G.-J. Ahn, Ed. New York: ACM, 2014, pp. 1054–1067.

[34] T. G. Kristensen, J. Nielsen, and C. N. Pedersen, "A tree-based method for the rapid screening of chemical fingerprints," *Algorithms for Molecular Biology*, vol. 5, no. 1, pp. 9–20, 2010.

[35] D. Vatsalan, "Scalable and approximate privacy-preserving record linkage," Ph.D. dissertation, Australian National University, 2014.

[36] E. A. Durham, "A framework for accurate, efficient private record linkage," Dissertation. Vanderbilt University, 2012.

[37] R. Schnell and C. Borgs, "Building a national perinatal database without the use of unique personal identifiers," in *Proceedings of the 2015 IEEE 15th International Conference on Data Mining Workshops*, 2015, pp. 232–239.

[38] A. Brown, C. Borgs, S. Randall, and R. Schnell, "High quality linkage using multibit trees for privacy-preserving blocking," IPDLN Conference 2016, 2016.

[39] T. Bachteler, J. Reiher, and R. Schnell, "Similarity filtering with multibit trees for record linkage," German Record Linkage Center, Nuremberg, Working Paper WP-GRLC-2013-02, 2013.

[40] R. Schnell, "An efficient privacy-preserving record linkage technique for administrative data and censuses," *Journal of the International Association for Official Statistics*, vol. 30, no. 3, pp. 263–270, 2014.