Recent Developments in Bloom Filter-based Methods for Privacy Preserving Record Linkage

– Public Version –

Rainer Schnell

German Record Linkage Center
University of Duisburg-Essen
Duisburg, Germany

Rainer.Schnell@uni-due.de

Curtin University, Perth, 27.9.2016

German RLC

UNIVERSITÄT DUISBURG ESSEN

Record Linkage

- Linking databases containing information on the same unit is an increasingly popular research strategy in many academic fields, commercial data mining and official statistics.
- The purpose of the linkage operation is the production of a dataset containing information on the same individuals.
- This dataset will be used for detailed statistical analyses, most of them requiring micro-data.

Approaches to Record-Linkage

- Technically, linking with a universally available unique national identification number (NID) is ideal.
- Such NIDs are available in Europe for example, for Denmark, Finland, Norway and Sweden.[1]
- If NIDs are available, linking data bases is technically trivial.
- However, in many research applications, NIDs are either not available at all or their usage is limited by law.
- In many applications, personal identifiers like names or date of birth have to be used.

[1] For a comprehensive review of NIDs, see `https://en.wikipedia.org/wiki/National_identification_number`

Using Personal Identifiers

- Identifiers are not stable and are recorded with errors: Winkler (2009:362) reports that 25% of true matches in a census operation would have been missed by exact matching.
- The use of exact matching identifiers will yield a non-random subset of records.
- Therefore, record-linkage has to use methods allowing for errors in identifiers.
- Many techniques for error-tolerant matching are based on string similarity measures.

String similarity measures

- In many applications, phonetic encodings such as Soundex are used.
- Bigrams or trigrams of letters within names are also widely used.
- Distance measures based on Levenshtein distances are used, but computationally expensive.
- Many studies have shown that the Jaro-Winkler distance seems to be robust and highly successful as distance measure for names.
- Linkage methods are using string similarities for the classification of potential pairs of records from different databases.

Linkage Methods

- All classification methods known to mankind have been suggested for record linkage, for example Cluster Analysis, Neural Nets, Bayes Networks, Tree Augmented Naive Bayes, Random Forests etc.
- The most widely used method for record-linkage is probabilistic record linkage using the Fellegi-Sunter (1969) model.
- The core of the method is a decision based on conditional likelihood ratios. It can be seen as a special version of a naive Bayes classifier.
- For census scale data, probabilistic record linkage is still considered to be state of the art.

Need for Privacy Preserving Record Linkage

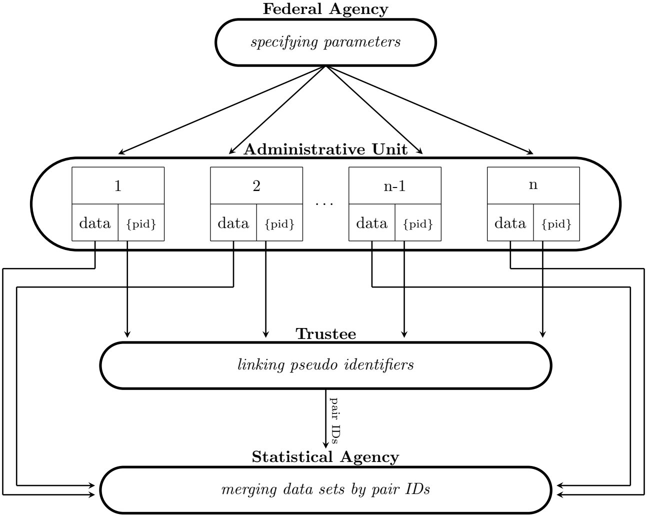
- In practice the data sets belong to different institutions and are controlled by different data custodians.
- Exchange of unencrypted identifiers between such institutions is usually restricted by law.
- Even within the same public administration this exchange may be prohibited by law ('data silos').
- If such linkages are allowed, usually special techniques protecting the identifiers have to be used.
- The set of techniques for record linkage without revealing identifiers is called Privacy Preserving Record Linkage or PPRL.

Implementation problems

- Examples for actual applications:
    - Linking neonatal data to perinatal data without patient ID (Schnell/Borgs 2015).
    - Longitudinal tracking of offenders across different databases.
    - Creation of a national mortality register.
    - Summing all credit debts for natural persons across all financial institutions.
    - Preparing for the Censuses 2021 & 2031.
- The solution has to be available now.
- Certification is needed and takes years.
- Most application need adoption of new laws, adding a few years to the implementation.

Most frequent settings for PPRL

- Required is micro-data, not the computation of summary statistics or a more elaborated statistical model.
- There are hundreds or even thousands of separate administrative units involved.
- The administrative units use different hardware and different software.
- Different data quality between units is most likely.
- Data generation and linkage will probable be a yearly enterprise (at most).
- Protocols requiring iterated computations (such as secure multi-party protocols for linking) will not be approved by the data protection agencies.



Federal Agency

specifying parameters

Administrative Unit

1 data {pid}

2 data {pid}

. . .

n-1 data {pid}

n data {pid}

Trustee

linking pseudo identifiers

pair IDs

Statistical Agency

merging data sets by pair IDs

HBC-Assumption

- In most security analysis in PPRL, the linkage unit acting as trustee is the attacker.
- Since the linkage unit receives only encrypted identifiers, the trustee might try to re-identify the encrypted keys.
- Most of the relevant literature in PPRL discusses ways to prevent re-identification by the trustee.
- Nearly all protocols intended for applications in PPRL assume semi-honest or honest-but-curious parties.
    - Semi-honest parties act according to the protocol, but try to learn secret information of the other parties.
    - They may use additional information or use information they gain during the execution of the protocol.
- This is widely regarded as a realistic assumption.

Privacy considerations

- From a mathematical point of view, it seems impossible to achieve the goal of a dataset, which can not be re-identified at all.
- This is attributed by Dwork/Pottenger (2013) to '. . . the fact that the privacy-preserving database is useful'.
- German jurisdiction defines de facto anonymity as modification of identifiers such as only a disproportionate investment of time, cost and labour would lead to re-identification.
- Therefore, the analysis of a cryptosystem for PPRL might depend on the assumed motives of an attacker (Wan et al. 2015).
- Note: No real-world re-identification attacks on PPRL databases have been reported (El-Emam et al. 2011).

Approaches to PPRL

Trustee: High organizational demands, requires a trustworthy institution with access to plain text identifiers.

Secure Multi-party: Computationally intensive, network access is necessary, typically not suitable for the development of a statistical model.

Encrypted Phonetic Codes: Only limited error-tolerance.

Modern Privacy Preserving Record Linkage: Several protocols suggested (overview: Vatsalan et al. 2013), but most of them are not applicable to the given problem.

Only two approaches have been used in practical applications with large databases within the setting described before:

1. Encrypted phonetic codes
2. Bloom Filter based PPRL

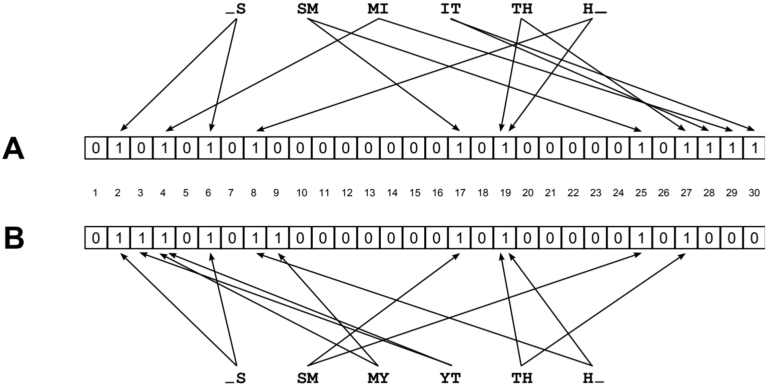
I will concentrate on the second set of methods.

PPRL with Cryptographic Bloom Filters

- Schnell et al. (2009) suggested a new method for the calculation of similarity between two encrypted strings for the use in record linkage procedures.
- Identifiers are split into q-grams and then hashed with several different keyed HMACs (MD5, SHA-1) in a bit vector.
- The mapping of different hash functions into a bit vector is called a Bloom filter (Bloom 1970).
- Given the Bloom filters, the initial string cannot be reconstructed without serious cryptanalysis.
- Only the Bloom filters are used for the linkage.
- The similarity between two strings is approximated by a similarity coefficient of their Bloom filters.

A simplified example

Two Bloom filters A, B with a length of 30
for "‘Smith"’ and "‘Smyth"’ and two HMACs.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	1	1
B	0	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0

Building Basic Bloom Filters

- Usually, between 10 and 20 different hash-functions are used for the encoding of each bigram.
- To speed up computations, we initially decided to use the double-hashing scheme by Kirsch/Mitzenmacher (2006).
- Here, each *n*-gram is encoded by the sum of the numeric representation of MD5 and SHA1 hashes:

  > If L is the length of the Bloom filter and *k* hash functions have to be used, double hashing uses for the *k*-th function:

$$g_k(x) = (SHA1(x) + k * MD5(x)) \mod L \qquad (1)$$

- For our application, this has been a classic example of a seemingly innocent choice with unforeseen cryptographic consequences (details will follow).

Cryptographic Long-term Keys (CLK)

- Due to legal constraints, in some applications in some countries only the use of one single key is allowed.
- Furthermore, frequency attacks of Basic Bloom filters seemed to be possible.
- Schnell et al. (2011) therefore suggested encrypting all identifiers in one single Bloom filter.
- The results produced by the CLK are slightly inferior to those of separate Bloom filters, but harder to attack.
- A variation using separate Bloom filters of different lengths, concatenating and permuting them are denoted as 'composite bloom filters' by Durham et. al (2012) .
- In most simulations, the differences between CLKs and composite Bloom filters are small.

Challenges of PPRL

In order of decreasing importance for actual real-world use (in my settings):

1. Security against attacks
2. Precision and Recall
3. Linkage errors
4. Missing identifiers
5. Scalability
6. Multiple databases

Current state of PPRL

- Precision and recall is already acceptable, many groups are working on small improvements.
- The way to solutions for handling linkage errors seems clear (multiple imputation, weighting).
- We simply have currently no useful proposal for missing identifiers. Most likely, the problem has to be handled in the same way as linkage errors.
- Scalability is in many settings not an issue.
    - Current solutions can handle 1 million records by 1 million records in reasonable time (< 3h).
    - Special hardware is an additional option.
- Some proposals for multiple databases have been presented and there seems to be many ways to attack this problem.
- Therefore, security of PPRL is the central problem for real-world use.

Attacks on Bloom filters

- Bloom filter based PPRL has been attacked by two different techniques:
    1. Constrained Satisfaction Solver (CSS) on frequencies of entire Bloom Filters: Kuzu et al. (2011, 2013).
    2. Cryptanalysis by interpreting the Bloom filter bit patterns as a substitution cipher (Niedermeyer et al. 2014, Kroll/Steinmetzer 2015).

Details on Attack I

- Kuzu et al. (2011, 2013) used a CSS program to align frequencies of simple Bloom Filters and unencrypted identifiers.
- This seems to be a variant of a simple rank swapping attack (Domingo-Ferrer/Muralidhar 2016), but they used the estimated length of the encrypted strings as additional information.
- Kuzu et al. (2013) used composite Bloom filters with the same CSS attack.
- The authors consider their attack on separate Bloom filters as successful, but not their attack on composite bloom filters.

Remarks on Attack I

- It should be noted that the CSS attack is based on the entire Bloom filter, therefore it is no decoding, but an alignment.
- This way of attack is impossible if each case generates a new bit pattern, for example by using salted encodings (Niedermeyer et al. 2014).

Details on Attack II

- Niedermeyer et al. (2014) attempt the decoding (actual revealing of all identifiers as clear text) by a cryptanalysis of individual bit patterns within the Bloom filters.
  - Niedermeyer et al. (2014) were successful with basic Bloom filters.
  - Kroll/Steinmetzer (2015) demonstrated success with CLKs/composite Bloom filters.
- Both attacks are based on the limited number of bit patterns generated by the linear combination of two hash functions in the double-hashing scheme.

Hardening Bloom Filters with Random Hashing

- Niedermeyer et al. (2014) showed that the double hashing scheme is vulnerable to attacks on bit patterns resulting from bigrams.
- Replacing the double-hashing scheme by random hashing should prevent the success of this attack on Bloom filters.
- Random hashing consists in using the bigrams as seed for random number streams.
- Random hashing is implemented using a linear-congruential pseudo-random number generator (LCG, Stallings 2014) to generate a sequence X with the length *k* for each *n*-gram:

$$X_{n+1} = (a * X_n + c) \bmod L.$$

- This increases the number of possible bit patterns (L=1000, k=15) for a $n - gram$ (atoms) from less than 10^6 to more than $6.8 * 10^{32}$.
- Therefore, the Niedermeyer-attack should fail for randomly hashed Bloom filters. This theoretical expectation has been empirically verified by Schnell/Borgs (2016).
- For an actual implementation, a cryptographic random stream generator (for example, Salsa20, Bernstein 2008) might be more secure.

Christen/Schnell/Vatsalan/Ranbaduge (2017)

- This year, we published an attack on Bloom-Filter encodings which is independent of the kind of hash-functions used .
- The attack is based on aligning clear text bigram frequencies and frequent bit positions and then filtering possible assignments.
- The attack is very fast (few seconds).
- Even in the worst-case scenario it yielded more than 40% correct re-identification.
- For this attack, only clear text frequencies of a similar sample of records are required.
- The attack can be used for CLKs and balanced and XOR-BF's as well.

Further improvements

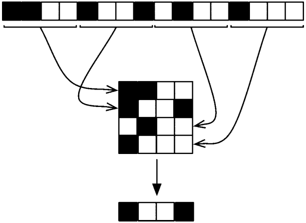
- Christen/Ranbaduge/Vatsalan/Schnell (2017) developed new methods to refine and validate the sets of possible, not-possible, and assigned q-grams.
- This approach is very fast and precise.
- However, frequency information on clear text and a couple of high frequent Bloom filters are needed.
- Currently, Christen et al. are working on a frequent pattern mining approach (based on the Apriori alg
- Therefore, we need methods to protect BFs against this kind of attacks.

General Remarks on protecting Bloom Filters

- The main difference to encryptions of clear text in cryptography is the fact, that PPRL of Bloom filters has to preserve only the similarity of top-k bit patterns in pairwise comparisons.
- A decryption is not necessary.
- This allows simple modifications of identifiers or bit patterns resulting in information loss.

Vector folding

- To speed up data base searches in chemometrics, Baldi et al. (2008) suggested folding bit vectors and applying XOR



- Schnell/Borgs (2016) suggested XOR-Folding for PPRL.
- Using the Niedermeyer-attack, we find 51 instead of 2539 atoms.
- Loss in precision and recall seems to be acceptable.
- We consider vector folding XOR as a simple technique to make some attacks more difficult.

Balanced Bloom filters

- The number of bits set in a Bloom filter (the Hamming weight) can be used for attacks.
- Therefore, Bloom filter with constant Hamming weights seem to be more secure against (unknown) attacks.
- Codes with constant Hamming weights are known as constant-weight codes or balanced codes (Knuth 1986).
- These can be obtained by joining a binary string with its negated copy (Berger 1961, Sayle 1998).
- The resulting binary vector should be permuted.
- This results in Balanced Bloom filters of length $2 * L$, which have the Hamming weight L.
- To prevent reversal of the balancing, using a stable identifier as a salt for the encoding makes the reversal difficult.
- Currently, we are exploring double balanced Bloom filters, where the columns also have the same Hamming weight.

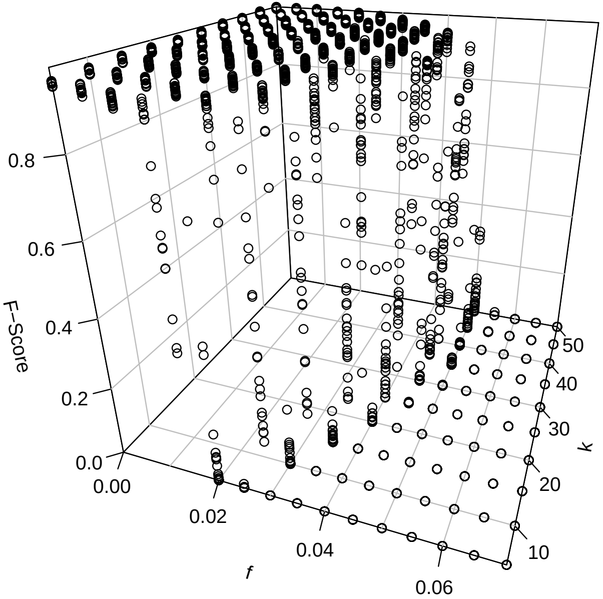
Randomized Response Bloom Filters: BLIP and RAPPOR

- Alaggan et al. (2012) proposed the application of Randomized Response to Bloom filters.
- The later Google publications (Erlingsson 2014, Fanti et al. 2015) used the same idea to propose RAPPOR, which is used within the Chrome browser.
- The idea is to use the randomized response technique on each bit position B_i of a Bloom filter:

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

- Schnell/Borgs (2016) suggested to use Randomized Response Bloom filters for PPRL and they seem to work quite well.

F-Score dependent on *f* and *k*



Summary

- Many modern PPRL methods are based on Bloom filters.
- Basic Bloom filters can be attacked by frequency alignments.
- Using CLKs instead of separate Bloom filters make attacks harder.
- Using salting (stable identifiers as part of the encryption password) defies most frequency attacks.
- Additional hardening techniques are available:
  - BLIP
  - Balancing and Double Balancing
  - Vector folding
  - Rehashing (Schnell 2016)
- We are working on a comparative privacy study of this and other approaches.

Finally . . .

- We have implemented these techniques in a R-library, the first release is due in December.
- We have started to work with a cryptographer (Frederik Armknecht) and already found some new encodings methods.
- We are still searching for an expert in coding theory.
- We are always looking for applications.
- Please contact:
  - `rainer.schnell@uni-due.de`

1 Bachteler, T., Reiher, J., Schnell, R. (2013): Similarity Filtering with Multibit Trees for Record Linkage, Working Paper WP-GRLC-2013-01, German Record Linkage Center, March 2013.

2 Baldi, P., Hirschberg, D. S., & Nasr, R. J. (n.d.). Speeding Up Chemical Database Searches Using a Proximity Filter Based on the Logical Exclusive-OR. J. Chem. Inf. Model, 1367–1378.

3 Bernstein, D. J. (2008). New Stream Cipher Designs. In M. Robshaw & O. Billet (Eds.) (pp. 84–97). Berlin, Heidelberg: Springer

4 Borst, F., Allaert, F.A., Quantin, C. (2001): The Swiss solution for anonymous chaining patient files; in: Patel, V. et al. (eds.): Proceedings of the 10th World Congress on Medical Informatics (MedInfo), London, 1239–1241.

5 Brown, A., Borgs,C. Randall,S., Schnell,R. (2016): High quality linkage using multibit trees for privacy-preserving blocking. IPDLN Conference 2016.

6 Christen, P. (2012): Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Berlin: Springer.

7 Christen,P. , Hand, D. (2016): A note on the F-measure for evaluating record linkage algorithms (and classification methods and information retrieval systems), presentation at INI, Cambridge, 8 September 2016.

8 Christen,P., Schnell,R. Vatsalan,D., Ranbaduge,T. (2017): Efficient Cryptanalysis of Bloom Filters for Privacy-Preserving Record Linkage, PAKDD 2017, Part I, LNAI 10234, pp. 628–640.

9 Christen, P., Ranbaduge, T., Vatsalan, D., Schnell, R. (2017) Precise and Fast Cryptanalysis for Bloom Filter Based Privacy-Preserving Record Linkage, under review.

10 Domingo-Ferrer,J., Muralidhar,K. (2016): New directions in anonymization: Permutation paradigm, verifiability by subjects and intruders, transparency to users, Information Sciences, vol. 337, no. 338, 11- 24

11 Durham, E. A., Kantarcioglu, M., Xue, Y., Toth, C., Kuzu, M., and Malin, B. (2014): Composite bloom filters for secure record linkage." IEEE transactions on knowledge and data engineering 26.12, 2956-2968.

12 Dwork, C., Pottenger, R. (2013): Toward practicing privacy; in: Journal of the American Medical Informatics Association, 20, 102-108

13 Emam, K.E., Jonker,E., Arbuckle,L., Malin,B. (2011): A Systematic Review of Re-identification Attacks on Health Data. PLoS One 6 (12): e28071.

14 Erlingsson, U., Pihur, V., Korolova, A. (2014): RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In: Ahn, G.-J. (ed.) Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014., pages 1054–1067, New York. ACM.

15 Fanti, G., Pihur, V., and Erlingsson, U. (2015). Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. ArXiv e-prints.

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

17 Herzog, T., Scheuren, F., Winkler, W.E. (2007): Data Quality and Record Linkage Techniques. New York.

18 Karmel,R., Gibson, D. (2007): Event-based Record Linkage in Health and Aged Care Services Data, BMC Health Services Research, 7, 154

19 Karmel, R. et al. (2010): Empirical aspects of record linkage across multiple data sets using statistical linkage keys: The experience of the PIAC cohort study; in: BMC Health Services Research, 10 (41).

20 Kirsch, A., Mitzenmacher,M. (2006): Less hashing, same performance: building a better Bloomfilter. 456-467 in: Y. Azar, Erlebach,T. (eds.): Algorithms-ESA 2006. Berlin: Springer.

21 Kristensen,T.G., Nielsen,J., Pedersen, C. N. S. (2010): A tree-based method for the rapid screening of chemical fingerprints. Algorithms for Molecular Biology, 5(9).

22 Kuzu, M. et al., 2011: A Constraint Satisfaction Cryptanalysis of Bloom Filters in Private Record Linkage. p 226-245 in S. Fischer-Hübner & N. Hopper (Eds.), Privacy Enhancing Technologies. Berlin: Springer.

23 Kuzu, M. et al. (2013): A Practical Approach to Achieve Private Medical Record Linkage in Light of Public Resources; in: Journal of the American Medical Informatics Association 20 (2): 285-292.

24 Kroll,M., Steinmetzer,S. (2015): Automated Cryptanalysis of Bloom Filter Encryptions of Health Records; in: Proceedings of the International Conference on Health Informatics, p. 5-13

25 McCallum, A./K. Nigam/L. H. Ungar (2000): Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM: 169-178.

26 Niedermeyer, F. (2016): Analysis of Bloom filters with Constant Hamming Weight. personal communication to the author.

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

28 Ong, T.C., Mannino, M.V., Schilling, L.M., Kahn, M.G. (2014): Improving Record Linkage Performance in the Presence of Missing Linkage Data. Journal of Biomedical Informatics, 52, 43-54.

29 Randall, S. M., Ferrante, A. M., Boyd, J. H., Brown, A. P., & Semmens, J. B. (2016). Limited privacy protection and poor sensitivity: Is it time to move on from the statistical linkage key-581? Health Information Management Journal, 45(2), 71–79.

30 Sayle, R. (1998): Hamming-grey codes for fingerprinting descriptors. Paper for Daylight EuoMUG ’98.

31 Schnell, R., Bachteler,T., Reiher,J. (2009): Privacy-preserving Record Linkage Using Bloom Filters. BMC Medical Informatics and Decision Making 9 (41).

32 Schnell,R., Bachteler,T., Reiher, J. (2011): A Novel Error-tolerant Anonymous Linking Code, German Record Linkage Center, Working Paper Series No. 2.

33 Schnell, R., Richter, A., Borgs, C. (2014): Performance of different methods for privacy preserving record linkage with large scale medical data sets, International Health Data Linkage Conference Vancouver, April 29th 2014.

34 Schnell,R., Thürling,M., Borgs, C. (2016): Explorations of Protective Measures against Cryptanalysis of Bloom filter-based Privacy-preserving Record Linkage, Discussion paper, German Record Linkage Center, Duisburg, 11.5.2016.

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

36 Sehili,Z., Kolb,L., Borgs,C., Schnell,R., Rahm,E. (2015): Privacy Preserving Record Linkage with PPJoin, in: Proceedings BTW, Lecture Notes in Informatics, 85–104.

37 Schnell, R. (2016): Privacy Preserving Record Linkage; in: Harron, K., Goldstein, H., Dibben, C. (eds): *Methodological Developments in Data Linkage*, Hoboken: Wiley, 201-225.

38 Schnell, R., Borgs, C. (2016): Randomized Response and Balanced Bloom Filters for Privacy Preserving Record Linkage, 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, 2016, pp. 218-224.

39 Schnell, R., Borgs, C. (2016): XOR-Folding for hardening Bloom Filter-based Encryptions for Privacy-preserving Record Linkage, Working paper WP-GRLC-2016-03, German Record Linkage Center, Duisburg, 22.12.2016

40 Vatsalan, D., Christen, P., Verykios, V. S. (2013): A taxonomy of privacy-preserving record linkage techniques; in: Information Systems, 38(6), 946-969.

41 Wan, Z., Vorobeychik, Y., Xia, W., Clayton, E. W., Kantarcioglu, M., Ganta, R., Heatherley, R., Malin, B. A. (2015). A Game Theoretic Framework for Analyzing Re-Identification Risk. PLOS ONE, 10(3), e0120592.

42 Winkler, W. E. (2009). Record linkage. In: Pfeffermann, D./Rao, C. (eds.): Handbook of Statistics Vol. 29A, p.351-380. Elsevier: Amsterdam.