German Record Linkage Center
WORKING PAPER SERIES NO. WP-GRLC-2011-02 | NOVEMBER 16, 2011

A Novel Error-Tolerant Anonymous Linking Code

Rainer Schnell | Tobias Bachteler | Jörg Reiher

A Novel Error-Tolerant Anonymous Linking Code

Rainer Schnell[1,2], Tobias Bachteler[2], and Jörg Reiher[2]

[1] Methodology Research Unit, Department of Social Sciences
University of Duisburg-Essen, D-47057 Duisburg, Germany
rainer.schnell@uni-due.de, http://www.uni-due.de

[2] German Record Linkage Center
University of Duisburg-Essen, D-47057 Duisburg, Germany
recordlinkage@iab.de, http://fdz.iab.de/linkage

Abstract.
An anonymous linking code is an encrypted key for linking data from different sources. So far, quite simple algorithms for the generation of such codes based on personal characteristics as names and date of birth are in common use. These algorithms will yield many non matching codes when facing errors in the underlying indentifier values. We suggested the use of Bloom filters for calculating string similarities in a privacy-preserving manner. Here, we claim that this principle can also be used for a novel error-tolerant but still irreversible encrypted key. We call the proposed code Cryptographic Longterm Key. It consists of one single Bloom filter into which identfiers are subsequently stored. Tests on simulated databases yield linkage results comparable to non encrypted identifiers and superior to results from hitherto existing methods. Since the Cryptographic Longterm Key can be easily adapted to meet quite different prerequisites it might be useful for many applications.

1 Introduction

In many epidemiological studies record linkage is required for patient follow-up or for combining cross-sectional data from different sources. Whenever feasible, researchers merge the databases using a common unique identification number. Otherwise, they most frequently resort to probabilistic record linkage based on personal identifiers like surnames, given names, date of birth, and address information [1-4]. In many countries legal regulations prescribing to ensure patient privacy in medical research, however, complicate record linkage on personal identifiers. There are three generic types of approaches to deal with this requirement. The first involves the use of a highly reliable data trustee [5,6]. The trustee performs the record linkage on the plain text identifiers previously transferred to him from the data sources. Afterwards, the trustee assigns some sort of pseudonyms to the records. Using these pseudonyms researchers are subsequently able to merge the epidemiological data. Approaches of the second type are commonly labelled "privacy-preserving record linkage" in the literature [7,8]. Each identifier is encrypted separately before, usually by a semi-trusted third party, the record linkage is performed. In contrast, the principle of the third class of approaches is at first to compose one single data item from the identifiers which is subsequently encrypted and used as an anonymous linking code (ALC). To some extent though made obsolete by the improvement of privacy-preserving record linkage methods, ALCs are still in common use [9].

Submitted to Methods of Information in Medicine, 22 September 2011.
Previous versions of this work were presented at the ASA Spring Methodology Conference at Tillburg University (19 May 2011), the Fourth Conference of the European Survey Research Association in Lausanne (21 July 2011), and the International Conference on Exploiting Existing Data for Health Research, University of St Andrews (11 September 2011).

Their design principle forms the basis of well-reputed commercial software products [10]. Additionally, there are instances in which the use of just a single data item for anonymous linkage is actually prescribed by law [11,12]. ALCs hence might find a wide range of applications in the future, either.

In [13] we suggested the use of Bloom filters for privacy-preserving record linkage. The favourable reception of the paper [14-16] as well as practical need in the course of the establishment of a German Mortality Register made us thinking about whether our idea could possibly used to build an encrypted key for linking patient data to epidemiological registers.

In this paper, we describe the design and the properties of a novel ALC. By exploiting the cryptographic properties of Bloom filters it preserves the privacy of identifier values while enabling error-tolerant comparisons. That is, it allows for typographical and other errors in the identifier values the linkage key is composed of. We additionally report on the results of empirical comparisons with previously used methods and discuss some properties of the novel ALC.

2 Related work

The basic principle of hitherto existing ALCs is to begin with a single string built from some identifier values. Prevailing are combinations of identifiers regarded as relatively stable as First name, Surname, Date of birth and Sex. This string serves as input to a cryptographic hash function. The resulting hash value constitutes the linking code.[1] If two ALCs agree exactly, the corresponding records are regarded as to represent the same person. Due to the cryptographic hash function it is impossible to re-engineer the original string. Moreover, if a keyed one-way hash function is used and the key of the hash function remains secret, a dictionary attack on the ALCs is impossible as well.

The most basic and common ALC consists of three simple steps [9]. First, the values of all constituent identifier values are concatenated. Secondly, the resulting string is standardized using some set of pre-processing rules[2]. Thirdly, this standardized string is put into a cryptographic hash function. [22-28] describe examples of the application of this *Basic ALC*.

An obvious drawback of this approach is that it does not tolerate any errors in the identifiers. Due to the design of cryptographic hash functions, the slightest input variation results in a completely different hash value. As a consequence, due to the notorious frequency of spelling and typographical errors in patient identifiers there will be many false negative classifications. In addition, since patients with variations of identifiers may have different characteristics than patients with exact matching identifiers, restricting the linkage in this manner is not an option.

The design of more elaborate ALCs allows for some variation or errors in the identifier values. In order to link data on subsequent hospitalizations of patients in Switzerland, the Swiss Federal Office for Statistics mandated the Encryption Section of the Swiss Military Department to find a linking method that preserves the confidentiality of the patients' medical records [29,30]. The result was an error-tolerant ALC based on Soundex encodings, called "anonymer Verbindungskode" resp. "code de liaison anonyme", that works as follows [29]: First, identifiers Surname and First name are Soundex encoded. Secondly, the Soundex codes are concatenated with the full Date of birth and Sex. Thirdly, this string is put into a cryptographic hash function. Applications with and discussions of this *Swiss ALC* can be found in [31-34].

Another path to an error-tolerant ALC is followed by the Australian Institute of Health and Welfare (AIHW). The basic idea is to make up the key just of certain characters instead of full

[1] Thereby, ALCs meet the definition of one-way [17-19] or irreversible pseudonyms [20,21]. ALCs can therefore be regarded as instances of the latter.

[2] Customary pre-processing routines involve the removal of non alphabetical characters from names, removal of non digits from dates, and bringing all characters to upper case.

last and first names. [35] tested various variants and identified the combination of the second, third and fifth character of Surname plus second and third of First given name, Date of birth and Sex as best performing[3]. This string forms the so called "Statistical Linkage Key" (SLK) which is routinely included in data sets of the AIHW [37]. If the SLK is additionally encrypted, it forms an ALC. This is routinely done within the Supported Accommodation Assistance Program (SAAP) National Data Collection, the resulting key is called the *Encrypted SLK* [38].[4]

3 Methods

The core problem to solve when constructing a Longterm identification key is how to reconcile privacy and error-tolerance. In [13] we suggested to use Bloom filters for calculating string similarities between two strings in a privacy-preserving manner. We believe that this principle can be used for a ALC as well. Since our ALC is intended to be persistent over long time periods we call it Cryptographic Longterm Key (CLK).

3.1 Bloom filters

A Bloom filter is a data structure proposed by Bloom [39] for checking set membership efficiently [40]. It consists of a bit array of length l with all bits initially set to 0. Furthermore, k independent hash functions $h_1, \dots, h_k$ are defined, each mapping on the domain between 0 and $l-1$. In order to store the set $S = \{x_1, x_2, \dots, x_n\}$ in the Bloom filter, each element $x_i \in S$ is hash coded using the k hash functions and all bits having indices $h_j(x_i)$ for $1 \leq j \leq k$ are set to 1. If a bit was set to 1 before, no change is made.

To store set elements in Bloom filters, we apply the double hashing scheme proposed by [41]. They show that only two independent hash functions are necessary to implement a Bloom filter with k hash functions without any increase in the asymptotic false positive probability [41]. Therefore, k hash values are computed with the function

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

where i ranges from 0 to $k-1$ and l is the length of the bit array. For security reasons, we use two keyed hash message authentication codes (HMACs), namely HMAC-SHA1 (h_1) and HMAC-MD5 (h_2) [42] to create the CLKs.

Bloom filters can also be used to determine whether two sets approximately match [43]. If two sets contain a common element, the bits having the same indices will be set to one when the sets are stored in Bloom filters under the same parametrization. Consequently, if two sets have many elements in common, their Bloom filters will have a large number of identical bit positions set to 1. Since the proportion of zeros in a Bloom filter for n elements is approximately [44]:

$$p' = \left(1 - \frac{1}{l}\right)^{kn}, \tag{2}$$

[3] During the setup, non-alphabetic characters are ignored. When a name fails to have a sufficient length, the digit '2' serves as a placeholder. For any other missing data, the digit '9' is used [36]

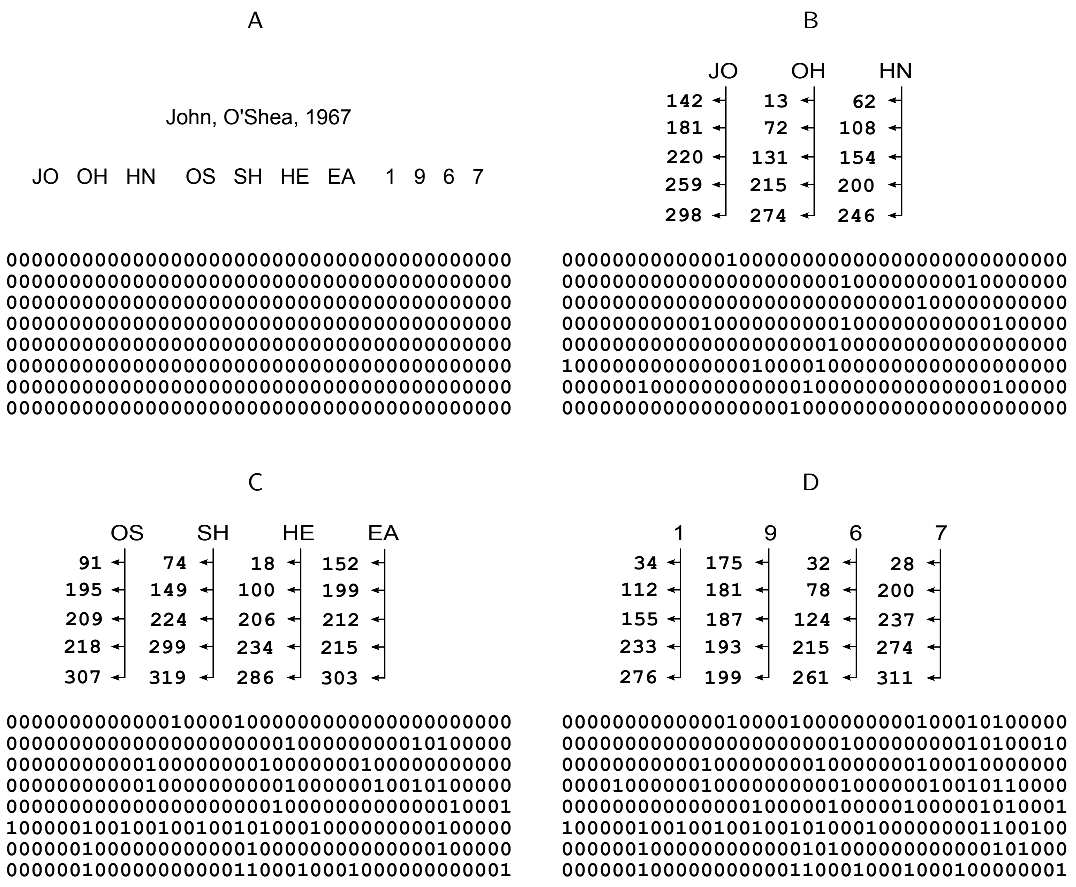
[4] From personal communication we are aware that other versions of this ALC variant are in actual use. The Swiss Federal Statistical Office, for example, links criminal statistics using a code composed of the first three letters of Birth name, the first two letters of First name, Sex, Date of birth, Birth canton and Citizenship canton. This and many other analogical codes unfortunately are *ad hoc* setups and consequently not well documented or published.

a long Bloom filter will contain mostly zeros. To assess the similarity of Bloom filters, a coefficient insensitive to many matching zeros is desirable. We chose the Dice-coefficient [45] therefore. For comparing bit strings, the Dice-coefficient can be defined as

$$D_{A,B} = \frac{2h}{(a+b)} \tag{3}$$

where h is the number of bit positions set to 1 in both bit strings, a is the number of bit positions set to 1 in A and b the number of bit positions set to 1 in B.

Fig. 1: **Building a Cryptographic Longterm Key**
Subfigure (A) starts with raw identifiers First name, Surname, and Birth year and an empty Bloom filter. The line breaks are inserted for graphical convenience only. After pre-processing, First Name and Surname are split into Bigrams, Birth year into Unigrams. (B) depicts the storing of the first n-gram subset (First name). Each Bigram is hashed five times. Bits having indices corresponding to the hash values are set to one. In (C) the hashing of the second n-gram subset (Surname) in the Bloom filter is shown. (D) illustrates the storing of the Unigram-set stemming from Birth year. The resulting Bloom filter forms the CLK for John O'Shea, born in 1967.



```
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
```

```
0000000000000100000000000000000000000000
0000000000000000000000100000000010000000
0000000000000000000000000000100000000000
0000000000010000000000100000000000100000
0000000000000000000001000000000000000000
1000000000000001000010000000000000000000
0000001000000000000100000000000000100000
0000000000000000001000000000000000000000
```

```
0000000000000100001000000000000000000000
0000000000000000000000100000000010100000
0000000000010000000010000000100000000000
0000000000010000000000100000010010100000
0000000000000000000001000000000000010001
1000001001001001001010001000000000100000
0000001000000000000100000000000000100000
0000001000000000000110001000100000000001
```

```
0000000000000100001000000000100010100000
0000000000000000000000100000000010100010
0000000000010000000010000000100010000000
0000100000010000000000100000010010110000
0000000000000001000001000001000001010001
1000001001001001001010001000000001100100
0000001000000000000101000000000000101000
0000001000000000000110001000100010000001
```

3.2 The Cryptographic Longterm Key

Suppose a CLK is to be made up from m identifiers. After pre-processing, each identifier is split into its set of constituent n-grams. Each of the m n-gram subsets is subsequently stored in the very same Bloom filter of length l using k hash functions as described in the previous section. The resulting Bloom filter constitutes the CLK.

The Dice coefficient of the two corresponding CLKs approximates the n-gram similarity [46] between two records of identifiers. Since we use keyed one-way functions to hash the n-grams, the identifier values cannot be re-engineered from a given CLK. The CLK therefore is an error-tolerant anonymous linking code. Figure 1 depicts the building process for three identifiers using a bit array of length $l = 320$ and $k = 5$ hash functions. In Figure 2 the comparison or matching of two CLKs is illustrated.

Fig. 2: **Comparing Cryptographic Longterm Keys**
To determine the similarity between two CLKs the Dice Coefficient as defined in section 3.1 is calculated. CLKs shown in subfigures (A) and (B) stem from exactly the same identifier values, consequently $D_{a,b} = 1.00$. The identifiers corresponding to CLKs from (A) and (C) are completely different. Since there are hardly any bits conjointly set to one, the Dice Coefficient results in $D_{a,c} = 0.23$. Regarding CLKs in (A) and (D), there is one typo in Surname. Apart from that the identifiers agree exactly and the Dice Coefficient results in $D_{a,d} = 0.80$. Note that of the Basic ALC, the Swiss ALC, and the Encrypted SLK described in section 2 neither would match given the identifier values from (A) and (D).

A

John, O'Shea, 1967

```
0000000000000100001000000000100010100000
0000000000000000000000100000000010100010
0000000000010000000010000000100010000000
0000100000010000000000100000010010110000
0000000000000001000001000001000001010001
1000001001001001001010001000000001100100
0000001000000000000101000000000000101000
0000001000000000001100010001000100000001
```

B

John, O'Shea, 1967

```
0000000000000100001000000000100010100000
0000000000000000000000100000000010100010
0000000000010000000010000000100010000000
0000100000010000000000100000010010110000
0000000000000001000001000001000001010001
1000001001001001001010001000000001100100
0000001000000000000101000000000000101000
0000001000000000001100010001000100000001
```

C

Nancy, Smith, 1982

```
0010100000001010000010000000000001100000
0000000010001000100000000000010000010000
0000000000000010110000000010000010000000
0010010100001000100100000001110000010110
0100010000100101000001000101000001000001
1000000010000100000100100100000001000010
0100000000000000000000000000000000001000
0010000001000000010001000000000010000000
```

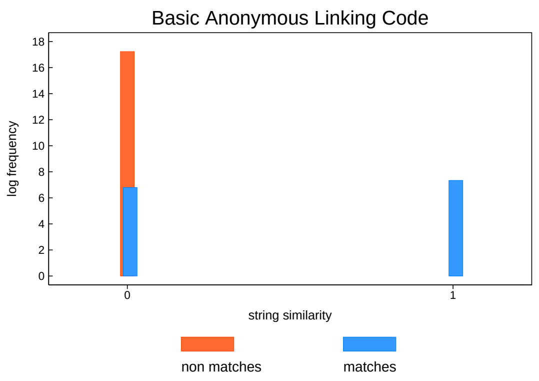
D

John, O'Dhea, 1967

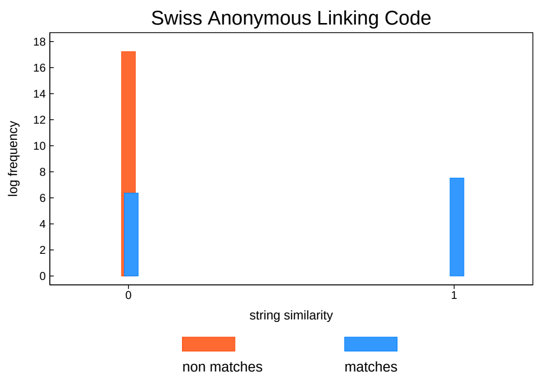
```
0000000010000100001000000000100010100000
0000000000000000000000101000000010000010
0000000000000000000010000000100010000000
1000100000010000000000100100000110110100
0000000000000001000001000001000001001001
1000001000001001000010000000000001100100
0000001000001000000101000000000000101000
0000001000000000001000010000100100100000
```

Fig. 3: **Comparison of the CLK with different ALCs and Plain Text Comparison**
The plots (A-E) show the conditional frequency distributions of matches and non matches over similarity scores. The CLK is set up with $l = 1000$. First Name and Surname were padded with spaces before being split into Bigrams [47]. The other identifiers were split into Unigrams. Each subset of n-grams is hashed using 10 HMACs and a different cryptographic key either. The lines in (D) and (E) indicate the optimal thresholds of classification. The table in (F) lists the true positive, false positive, true negative and false negative classifications of the methods. For the CLK and the plain text comparison optimal thresholds for classification as a link ($.84$, resp. $.83$) are adopted.

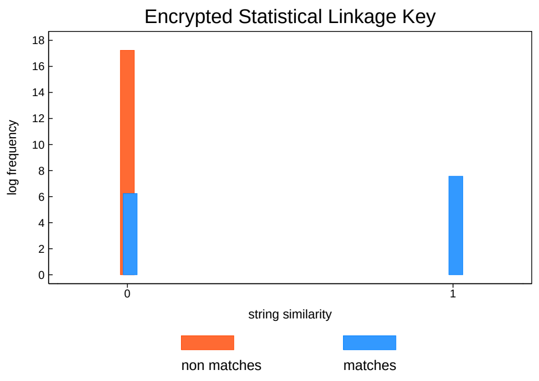
A



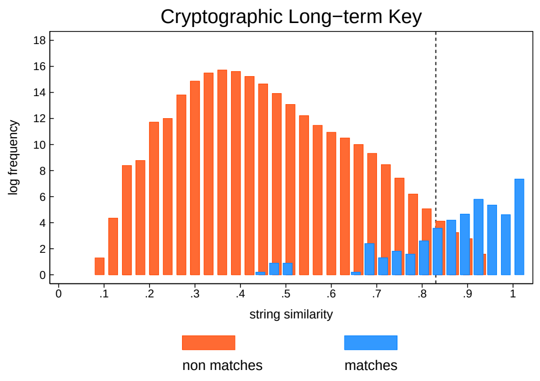
B



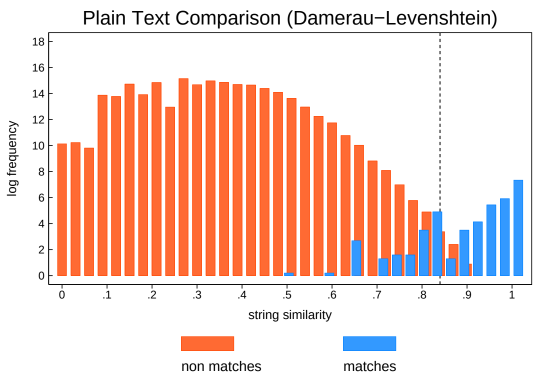
C



D



E



F

Method	TP	FP	TN	FN
Basic ALC	1,265	0	24,998,000	735
Swiss ALC	1,521	0	24,998,000	479
Encrypted SLK	1,580	0	24,998,000	420
CLK	1,953	50	24,997,950	47
Plain text comparison	1,945	22	24,997,978	55

4 Results

We compared the CLK with the Basic and Swiss ALCs, as well as with the Encrypted SLK for evaluation. Additionally, we put the CLK to a test against the plain text comparison of identifiers using the Damerau-Levenshtein distance function. We simulated records comprising the identifiers First name, Surname, Birth day, Birth month, Birth year and Sex based on a German phone book.[5] The test data base consisted of two files: First, we simulated file A with $n_A = 2,500$ records. From them we took $2,000$ records and introduced errors at random to them.[6] Next, we put these $2,000$ garbled records into a second file B and appended $8,000$ freshly simulated records, making up file B with $n_B = 10,000$. There were $25,000,000$ record pairs hence, $2,000$ of them matches and $24,998,000$ non matches. Crossing this true matching state with the classifications from the methods compared, we got the usual true positive (TP), false positive (FP), true negative (TN) and false negative (FN) classifications.

Results are shown in Figure 3. The conditional frequency distributions of matches and non matches demonstrate that the CLK outperforms the other ALCs clearly (subfigures A-D). As the table in subfigure (F) reveals, it exhibits a far smaller total number of false classifications. As shown by the numbers, the CLK performs only slightly less well as the plain text comparison. The quite similar conditional distributions of matches and non matches in subfigures (D) and (E) support this reasoning.

5 Discussion

Security Consider a setting with several data sources (e.g. hospitals) and a central data registry. At the data sources, personal identifiers are replaced by CLKs and records subsequently sent to the registry. At the registry, records are linked using the CLKs. According to Kerkhoffs' principle, we assume public knowledge of the design principles of the CLKs and parameters l, n, and k. The cryptographic key of the HMACs however remains private to the data sources.

The registry is not able to re-engineer the identifier values because one-way functions are used. Also, the registry is not able to mount a dictionary attack on the CLKs because it is not aware of the cryptographic key. The registry might however undertake a frequency attack since the frequencies of the bit positions set to 1 are of course related to the frequencies of n-grams in the original identifiers. In [13] we discussed a frequency attack on Bloom filters containing just one identifier, a more thorough analysis is contained in [16]. To sum up, the success of a frequency attack depends on the ratio of the number of hash functions used to the length of the Bloom filter $\frac{k}{l}$. The CLK however takes several instead of one identifier only[7], and each can be hashed with a different cryptographic key k. We regard a frequency attack as extremely difficult therefore. In any case, it is possible to enhance security by raising $\frac{k}{l}$.

[5] We took First name and Surname from the phone book. A Birth year was then simulated according to the actual age distribution of Germany in 2006. Lastly, we used a precompiled list recording first names with sex. If a first name did not appear in the list, we assigned Sex with $p = .5$.

[6] For generating errors we built an error simulation tool following the design principles of "generate2.py" [48]. We introduced a variety of error types, including typos, phonetic errors, and ocr-errors. Error probabilities were independently applied according to "gold standard" data stemming from the Panel Study "Labour Market and Social Security" [49]. We adopted for First name $p = .2$, Surname $p = .15$, full Date of birth $p = .05$, and Sex $p = .05$.

[7] If one stores the Bigrams of a surname of average length, say 7 characters, in a Bloom filer of length $l = 1000$ using $k = 30$ hash functions, there are at most 180 Bits set to one. If one hashes First name (e.g. 6 characters), Surname (7 characters), Birth day, Birth month, Birth year, and Sex as in the evaluation study ($l = 1000$, $k = 10$), there are more, at most 200, Bits set to one.

If one of the data sources acquires the CLKs from the registry, the possibility of a dictionary attack emerges. This could be prevented if the registry simply rotates the incoming CLKs by a secret number of positions. The ability to link the CLKs is thereby preserved whereas the possibility of a dictionary attack by a data source perishes.

Collisions An important feature of any ALC and pseudonym system is collision-freeness. For example, in [29] a test for collisions of the Swiss ALC is described. Swiss ALCs of 220, 020 unique combinations of patients' identifiers (First name, Surname, Date of birth, and Sex) from a hospital data base were created. Among them, there were 304 double Swiss ALCs and 1 triple Swiss ALC. This means a false-positive (or collision) rate of $\frac{(2\times304+3\times1)}{222020} = .003$, which was considered acceptable.

We put the CLK to an analogous test. Due to lack of real test data, we took 20, 931, 406 combinations of First name and Surname from a German phone book and added simulated Date of birth and Sex as described in Footnote 5. Among the resulting records, there were 20, 916, 246 unique combinations of identifiers. We generated CLKs from them as described in the caption text of Figure 3. We found just 526 doublets, making a collision rate of $\frac{(2\times526)}{20916246} = .00005$.

Efficiency Regarding ALCs, time resources needed comparing them is of relevance only.[8] Table 1 displays the runtimes of the different runs in our evaluation study. Whereas it took 19 minutes more to compare the CLKs than the other ALCs, this certainly does not imply that time complexity is prohibitive. Actually, the CLK exhibits a runtime comparable to plain text similarity functions. After all, it took additional 39 minutes to determine the Damerau-Levenshtein distances. Since this would be an unfair confrontation, we determined the runtime when applying n-grams on the plain text values as well. It turns out that it took 6:30 minutes more to calculate the plain text n-grams than to compare the CLKs.

Table 1: **Runtimes of comparison in minutes.**
Each computer run involved 25, 000, 000 record pairs. All computations were performed using a 2.80GHz Pentium D Processor running Windows XP, with 2GB of RAM. All routines were implemented in Java.

Method	Runtime
Basic ALC	53:36
Swiss ALC	53:10
Encrypted SLK	53:34
CLK	72:08
Plain text comparison (Damerau-Levenshtein)	111:32
Plain text comparison (n-grams)	76:42

[8] Compuationally, coding the identifier values is a cheap task. For example, the creation of the 21 Million CLKs used for the collision test took us 50 minutes only.

Flexibility The design and application of the CLK might be modified in several respects to meet certain practical requirements. That is, it is more flexible than hitherto existing ALCs in various regards.

First, the CLK allows for relative weighting of included identifiers. For example, if Surname is to be of greater relevance than First Name, simply use relatively more hash functions to store Surname in the CLK.

Second, the CLK allows for adding additional identifiers *ex post*. Suppose a registry is accumulating CLKs. After years, there are more and more collisions and it would be advantageous to add an identifier. This identifier could, in principle, be added to the CLKs without taking recourse to the original values of the identifiers already incorporated. There is no need to know the cryptographic keys whatsoever since each identifier is hashed using a different one. When the counting variant of Bloom filters [50] is used to set up the CLKs, there would be even the possibility of a removal of identifiers.

Third, the design of the CLK allows for meaningful comparisons even in the event that some CLKs only contain additional identifiers. Suppose in some CLKs an indicator of vital status ("dead/alive") is added for some reason. Depending on the data quality of the other identifiers this disturbs the similarity approximation to some extent, but makes it not obsolete as would be the case with other ALCs.

Fourth, as discussed above, the security of the CLKs can be enhanced by raising $\frac{k}{l}$. On the other hand, lowering $\frac{k}{l}$ will result in an improved record linking quality. The CLK hence allows for fine-balancing security and linking ability under the terms of a specific application.

Fifth, whereas the other ALCs return simply "ALC matches" or "ALC does not match", the CLK returns continuous similarity scores (see Figure 1). This implies that the CLK allows for economizing on the costs of false linking classifications by adopting specific loss functions in different studies. Suppose the cost of a false positive classification is relatively high as compared to the cost of a false negative classification. By raising the threshold for classification (i. e. it takes higher a similarity to be classified as a link), false positive cases tend to be avoided. Lowering the threshold would work the other way around.

6 Conclusion

In this paper, we presented the Cryptographic Longterm Key, a new method for building an anonymous linking code. Hitherto suggested ALCs consist of a single string built from some identifier values which is put into a cryptographic hash function. If two ALCs agree exactly, the corresponding records are considered to be linked, otherwise to be not linked. In contrast, the CLK consists of a single Bloom filter in which n-gram sets of a several identifiers are stored. It allows to approximate the n-gram similarity between two sets of identifier values. It is a novel, error-tolerant ALC therefore. Using simulated data, we compared its performance with previously suggested ALCs. It results to be clearly superior in matching effectivity being as efficient as a n-gram comparison of plain text identifier values.

As a future work, we intend to test the CLK in a real-world setting. Additionally, we plan to investigate the security properties more thoroughly. Finally, we want to put the applicability of Counting Bloom filters for building CLKs to the test.

References

1. William E. Winkler. Record linkage. In Danny Pfeffermann and C.R. Rao, editors, *Handbook of Statistics 29A, Sample Surveys: Design, Methods and Applications*, pages 351–380. Elsevier, North-Holland, Amsterdam, 2009.
2. Thomas N. Herzog, Fritz J. Scheuren, and William E. Winkler. Record linkage. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):535–543, 2010.
3. Douglas P. Jutte, Leslie L. Roos, and Marni D. Brownell. Administrative record linkage as a tool for public health. *Annual Review of Public Health*, 31:91–108, 2010.
4. Xiaochun Li and Changyu Shen. Linkage of patient records from disparate sources. *Statistical Methods in Medical Research. Published online before print June 10, 2011.*, 2011.
5. Christopher W. Kelman, A. John Bass, and C. D'Arcy J. Holman. Research use of linked health data: a best practice protocol. *Australian and New Zealand Journal of Public Health*, 26(3):251–255, 2002.
6. Tim Churches. A proposed architecture and method of operation for improving the protection of privacy and confidentiality in disease registers. *BMC Medical Research Methodology*, 3(1), 2003.
7. Rob Hall and Stephen E. Fienberg. Privacy-preserving record linkage. In Josep Domingo-Ferrer and Emmanouil Magkos, editors, *Proceedings of the 2010 International Conference on Privacy in Statistical Databases: 22–24 September 2010; Corfu, Greece*, pages 269–283, Berlin, 2010. Springer.
8. Alexandros Karakasidis and Vassilios S. Verykios. Advances in privacy preserving record linkage. In Tokuro Matsuo and Takayuki Fujimoto, editors, *E-Activity and Intelligent Web Construction: Effects of Social Design*, pages 22–34. Information Science Reference, Hershey, PA, 2011.
9. Thomas N. Herzog, Fritz J. Scheuren, and William E. Winkler. *Data Quality and Record Linkage Techniques.* Springer, New York, 2007.
10. Andrew Friedrich. IBM DB2 Anonymous Resolution: Knowledge discovery without knowledge disclosure. IBM DB2 Anonymous Resolution Whitepaper, 2005.
11. Gesetz über Krebsregister (Krebsregistergesetz-KRG) vom 4. November 1994.
12. Act LVIII of 2001 on the Magyar Nemzeto Bank, in force as of July 1, 2011. Download: 13-08-2011, http://english.mnb.hu/A_jegybank/organisation.
13. Rainer Schnell, Tobias Bachteler, and Jörg Reiher. Privacy-preserving record linkage using bloom filters. *BMC Medical Informatics and Decision Making*, 9(41), 2009.
14. Elizabeth Durham, Yuan Xue, Murat Kantarcioglu, and Bradley Malin. Private medical record linkage with approximate matching. In *Proceedings of the 2010 American Medical Informatics Association Annual Symposium*, pages 182–186, 2010.
15. Elizabeth Durham, Yuan Xue, Murat Kantarcioglu, and Bradley Malin. Quantifying the correctness, computational complexity, and security of privacy-preserving string comparators for record linkage. *Information Fusion*, Accepted Manuscript, 2011.
16. Mehmet Kuzu, Murat Kantarcioglu, Elizabeth Durham, and Bradley Malin. A constraint satisfaction cryptanalysis of bloom filters in private record linkage. In *The 11th Privacy Enhancing Technologies Symposium: 27–29 July 2011; Waterloo, Canada*, 2011.
17. Simone Fischer-Hübner. *IT-Security and Privacy: Design and Use of Privacy-Enhancing Security Mechanisms.* Springer, Berlin, 2001.
18. Rita Noumeir, Alain Lemay, and Jean-Marc Lina. Pseudonymization of radiology data for research purposes. *Journal of Digital Imaging*, 20(3):284–295, 2007.
19. Bernice S. Elger, Jimison Iavindrasana, Luigi Lo Iacono, Henning Müller, Nicolas Roduit, Paul Summers, and Jessica Wright. Strategies for health data exchange for secondary, cross-institutional clinical research. *Computer Methods and Programs in Biomedicine*, 99(2):230–251, 2010.
20. Khaled El Emam and Anita Fineberg. An overview of techniques for de-identifying personal health information. Ottawa: CHEO Research Institute, 2009.
21. Health System Use Technical Advisory Committee, Data De-Identification Working Group. 'Best Practice' Guidelines for Managing the Disclosure of De-Identified Health Information. Ottawa: Canadian Institute for Health Information, 2010.

22. Elizabeth van Eycken, Karin Haustermans, Frank Buntinx, Ann Ceuppens, Joost Weyler, Etienne Wauters, Herman Van Oyen, Monique De Schaever, Dirk Van den Berge, and Margareta Haelterman. Evaluation of the encryption procedure and record linkage in the belgian national cancer registry. *Archives of Public Health*, 58(6):281–294, 2000.
23. Matthias Wjst. Anonymizing personal identifiers in genetic epidemiologic studies. *Epidemiology*, 16(1):131, 2005.
24. Boonchai Kijsanayotin, Stuart M. Speedie, and Donald P. Connelly. Linking patients' records across organizations while maintaining anonymity. *Proceedings of the 2007 American Medical Informatics Association Annual Symposium*, page 1008, 2007.
25. Eugen Schülter, Rolf Kaiser, Mark Oette, Claudia Müller, Norbert Schmeißer, Joachim Selbig, Niko Beerenwinkel, Thomas Lengauer, Martin Däumer, and Daniel Hoffmann. Arevir: A database to support the analysis of resistance mutations of human immunodeficiency. *European Journal of Medical Research*, 12(Supplememt III):10–11, 2007.
26. Stephen B. Johnson, Glen Whitney, Matthew McAuliffe, Hailong Wang, Evan McCreedy, Leon Rozenblit, and Clark C. Evans. Using global unique identifiers to link autism collections. *Journal of the American Medical Informatics Association*, 17(6):689–695, 2010.
27. Shinya Kimura, Toshihiko Sato, Shunya Ikeda, Mitsuhiko Noda, and Takeo Nakayama. Development of a database of health insurance claims: Standardization of disease classifications and anonymous record linkage. *Journal of Epidemiology*, 20(5):413–419, 2010.
28. Antje Tessmer, Tobias Welte, Ruprecht Schmidt-Ott, Sonja Eberle, Grit Barten, Norbert Suttorp, and Tom Schaberg. Influenza vaccination is associated with reduced severity of community-acquired pneumonia. *European Respiratory Journal*, 38(1):147–153, 2011.
29. Office fédéral de la statistique. La protection des données dans la statistique médicale. Technical report, Neuchâtel, 1997.
30. David-Olivier Jaquet-Chiffelle and Jean-Paul Jeanneret. How to protect the rights of patients to medical secrecy in official statistics. *Information Security Bulletin*, 6(8):41–44, 2001.
31. François Borst, François-André Allaert, and Catherine Quantin. The Swiss solution for anonymous chaining patient files. In V. Patel, R. Rogers, and R. Haux, editors, *Proceedings of the 10th World Congress on Medical Informatics: 2–5 September 2001; London*, pages 1239–1241, Amsterdam, 2001. IOS Press.
32. Alberto Holly, Lucien Gardiol, Yves Eggli, Tarik Yalcin, and Tiago Ribeiro. Ein neues gesundheitsbasiertes risikoausgleichssystem für die schweiz. *G+G Wissenschaft*, 5(2):16–31, 2005.
33. Yves Eggli, Patricia Halfon, Mehdi Chikhi, and Till Bandi. Ambulatory healthcare information system: A conceptual framework. *Health Policy*, 78:26–38, 2006.
34. Anas El Kalam, Carlos Melchor, Stefan Berthold, Jan Camenisch, Sebastian Clauß, Yves Deswarte, Markulf Kohlweiss, Andriy Panchenko, Lexi Pimenidis, and Matthieu Roy. Further privacy mechanisms. In Jan Camenisch, Ronald Leenes, and Dieter Sommer, editors, *Digital Privacy*, pages 485–555. Springer, Berlin, 2011.
35. Trish Ryan, Bella Holmes, and Diane Gibson. A national minimum data set for home and community care. Canberra, AIHW, 1999.
36. Rosemary Karmel. Data linkage protocols using a statistical linkage key. Technical report, Canberra: AIHW, 2005.
37. Rosemary Karmel, Phil Anderson, Diane Gibson, Ann Peut, Stephen Duckett, and Yvonne Wells. 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(41), 2010.
38. Australian Institute of Health and Welfare. Saap national data collection collectors manual. Canberra: AIHW, 2005.
39. Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.
40. Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: a survey. *Internet Mathematics*, 1(4):485–509, 2003.
41. Adam Kirsch and Michael Mitzenmacher. Less hashing same performance: building a better bloom filter. In Yossi Azar and Thomas Erlebach, editors, *Algorithms-ESA 2006. Proceedings of the 14th Annual European Symposium: 11-13 September 2006; Zürich, Switzerland*, pages 456–467, Berlin, 2006. Springer.

42. Hugo Krawczyk, Mihir Bellare, and Ran Canetti. Hmac: keyed-hashing for message authentication. Internet RFC 2104, 2009.04.10 1997.
43. Navendu Jain, Mike Dahlin, and Renu Tewari. Using bloom filters to refine web search results. In AnHai Doan, Frank Neven, Robert McCann, and Geert Jan Bex, editors, *Proceedings of the Eight International Workshop on the Web and Databases: 16-17 June 2005; Baltimore*, pages 25–30, 2005.
44. Michael Mitzenmacher and Eli Upfal. *Probability and computing: an introduction to randomized algorithms and probabilistic analysis*. Cambridge University Press, Cambridge, 2005.
45. Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
46. Graham A. Stephen. *String Searching Algorithms*. World Scientific, Singapore, 1994.
47. Alexander M. Robertson and Peter Willett. Applications of n-grams in textual information systems. *Journal of Documentation*, 54(1):48–67, 1998.
48. Peter Christen and Angus Pudjijono. Accurate synthetic generation of realistic personal information. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu-Bao Ho, editors, *Advances in Knowledge Discovery and Data Mining. Proceedings of the 13th Pacific-Asia Conference: 27–30 April 2009; Bangkok*, pages 507–514, Berlin, 2009. Springer.
49. Mark Trappmann, Stefanie Gundert, Claudia Wenzig, and Daniel Gebhardt. Pass: A household panel survey for research on unemployment and poverty. *Schmollers Jahrbuch*, 130(4):609–622, 2010.
50. Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000.

IMPRINT

Publisher

German Record-Linkage Center
Regensburger Str. 104
D-90478 Nuremberg

Template layout

Christine Weidmann