Securing Bloom Filters for Privacy-preserving Record Linkage[*]

Thilina Ranbaduge
The Australian National University
Canberra, Australia
thilina.ranbaduge@anu.edu.au

Rainer Schnell
University Duisburg-Essen
Duisburg, Germany
rainer.schnell@uni-due.de

ABSTRACT

Privacy-preserving record linkage (PPRL) facilitates the matching of records that correspond to the same real-world entities across different databases while preserving the privacy of the individuals in these databases. A *Bloom filter* (BF) is a space efficient probabilistic data structure that is becoming popular in PPRL as an efficient privacy technique to encode sensitive information in records while still enabling approximate similarity computations between attribute values. However, BF encoding is susceptible to privacy attacks which can re-identify the values that are being encoded. In this paper we propose two novel techniques that can be applied on BF encoding to improve privacy against attacks. Our techniques use neighbouring bits in a BF to generate new bit values. An empirical study on large real databases shows that our techniques provide high security against privacy attacks, and achieve better similarity computation accuracy and linkage quality compared to other privacy improvements that can be applied on BF encoding.

CCS CONCEPTS

• **Security and privacy → Data anonymization and sanitization**; • **Information systems → Entity resolution**.

KEYWORDS

Perturbation, hardening, sliding window, random sampling, XOR

ACM Reference Format:
Thilina Ranbaduge and Rainer Schnell. 2020. Securing Bloom Filters for Privacy-preserving Record Linkage. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3412105

1 INTRODUCTION

In today's Big Data era, organisations collect vast quantities of data every day [3]. To improve the quality of decision making, organisations increasingly require to identify matching records from different databases that refer to the same real-world entity [1, 13]. Generally, attributes, such as name, address, or date of birth, are used for the purpose of matching records [14].

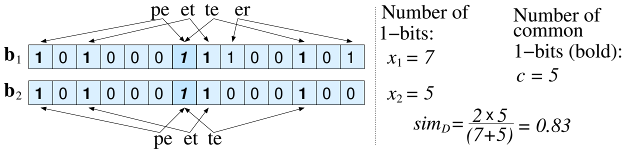

Figure 1: An example Dice coefficient similarity (sim_D) calculation of the two FIRSTNAMES 'peter' and 'pete' encoded in BFs. The italic bit shows a hash collision.

Even though these attributes allow accurate linking of records, due to growing privacy concerns organisations often do not want their sensitive personal data to be revealed to any other party [13]. Privacy-preserving record linkage (PPRL) aims at linking of databases without the need of any sensitive data to be shared between the parties involved in the linkage. This requires data in sensitive databases to be encoded or encrypted before they can be linked [14].

Encoding or encryption methods used in PPRL must facilitate approximate similarity calculations due to data quality issues [14]. The PPRL techniques that have been proposed so far either rely on expensive secure computations to achieve strong privacy guarantees, or use efficient perturbation techniques [13, 14]. However, perturbation techniques can be vulnerable to privacy attacks that can re-identify sensitive values in an encoded database [4].

Bloom filter (BF) encoding is currently the most popular privacy technique employed in different practical applications to link sensitive databases, and has almost become a standard for PPRL [9, 14]. In PPRL, commonly character q-grams extracted from attribute values are hashed into BFs using k hash functions that set certain bit positions to 1. Similarities are then calculated on BFs based on the number of 1-bits they have. Figure 1 shows the encoding of bigrams ($q = 2$) of two string values into 14 bits long BFs using $k = 2$ hash functions, and their Dice coefficient similarity calculation [3].

However, BF encoding can be susceptible to privacy attacks [6, 13]. Sensitive values that occur frequently in an encoded database can lead to frequent bit patterns in BFs that can be identified, and even individual frequent q-grams can be found using pattern mining based privacy attacks [15]. This potentially allows the re-identification of values encoded in BFs [4, 6].

Contribution: We propose two novel techniques that aim to harden BFs in order to make them more resilient to privacy attacks. In the proposed techniques, we apply (1) sliding windows, and (2) re-sampling based methods to select certain bits in the original BFs and apply bit-wise exclusive OR (XOR) upon these selected bits to generate new bit values to be used in the hardened BFs. We evaluate the performance of these techniques by comparing them to existing hardening techniques on BFs in terms of scalability, linkage quality, and privacy. To the best of our knowledge, no such evaluation of hardening techniques has so far been published.

[*] Partially funded by the Australian Research Council under grant DP160101934.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412105

2 RELATED WORK

The vulnerabilities of BFs in PPRL [4, 7, 15] have recently been addressed by the development of *hardening* techniques [12]. Hardening aims to modify the bit patterns in BFs to reduce the frequency information required by privacy attacks [11]. In this section we describe existing hardening techniques that can be applied on BFs.

Balancing: This technique changes any non-uniform Hamming weight distribution of BFs into (near) uniform distribution to avoid any frequency analysis on BFs [11]. Balanced BFs can be constructed by concatenating a BF of length l with its negated copy (all bits flipped) and then permuting the $2 \times l$ bits. Balanced BFs potentially make the identification of certain frequent bit patterns more difficult. However, balancing BFs requires more space than basic BFs and the additional set bits can always lead to more false matches.

XOR Folding: Schnell and Borgs [12] proposed a BF hardening technique that uses vector folding [2] combined with a bit-wise XOR operation. A BF $\mathbf{b}$ of length l is first divided into two halves $\mathbf{b}_1$ and $\mathbf{b}_2$ each of length $l/2$, where $\mathbf{b}_1[i] = \mathbf{b}[i]$ and $\mathbf{b}_2[i] = \mathbf{b}[i+l/2]$, with $0 \leq i < l/2$. Then, the two BFs $\mathbf{b}_1$ and $\mathbf{b}_2$ are combined into a new hardened BF $\mathbf{b}_H$ of length $l/2$ by applying the bit-wise XOR operation $\oplus$ on each bit $0 \leq i < l/2$, where $\mathbf{b}_H[i] = \mathbf{b}_1[i] \oplus \mathbf{b}_2[i]$. An XOR operation applied on a bit position ensures it is not possible to recover the original two input bits values, which improves privacy.

Salting: Salting is a hardening technique proposed by Niedermeyer et al. [7] to avoid privacy attacks on BFs by adding an extra (string) value to each q-gram before it is hashed, where these string values are very specific for an individual and do not change over time. Examples of salting values can be the year of birth or the birth place of an individual. So instead of hashing q-grams into BFs, salted q-grams are hashed. Thus, with salting, most q-grams that are frequent will become much less frequent once concatenated with a salting value which potentially improve privacy.

Rule 90: Rule 90 is a cellular automata [16] that is based on the bit-wise XOR function of two bits in a bit array used to generate a new bit array. Schnell and Borgs [10] first proposed to use Rule 90 as a hardening technique for BF encoding because it is non reversible. Each bit position p, with $0 \leq p \leq l-1$, in a BF of length l is modified by XORing the bits at positions $(p-1)$ *mod* l and $(p+1)$ *mod* l, where the modulo (*mod*) function is used to 'wrap around' the input bits of the first and last bit in a BF.

Markov chaining: This technique [10] avoids frequency attacks on q-grams encoded into BFs by adding extra q-grams randomly based on their frequent co-occurrences. For each unique q-gram q, c other q-grams are randomly selected to be encoded with q based on their probability to occur after q, where c is the *chain length* parameter that is used in the hardening technique. However, larger values of c will result in more distorted frequency distributions and thus improve privacy, but will reduce linkage quality.

Bloom and Flip (BLIP): Schnell and Borgs [12] proposed this approach which flips bit values at certain positions in a BF according to a differential privacy mechanism [5]. Here, the bit value at position p in a BF $\mathbf{b}$ is flipped randomly to either 1 or 0 with a flip probability f. However, depending upon the percentages of 1-bits in BFs, BLIP can result in an increase or decrease of the similarities calculated between hardened BFs.

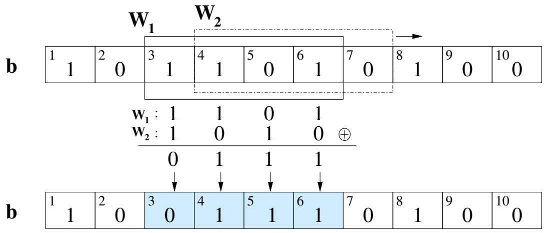

Figure 2: An example of the windowing based hardening technique with a window size $w = 4$. $\oplus$ represents bit-wise XOR operation.

Re-hashing: This technique [8] uses several bits from a BF $\mathbf{b}$ to generate a new set of bits. The idea is to slide a window of width w bits over $\mathbf{b}$, where the window is moved s bits (step size) forward in each step. The bits in the window are used to calculate an integer value which is then used as the seed for a pseudo random number generator (PRNG). A sequence of k_{re} random numbers $[p_1, \ldots, p_{k_{re}}]$, each in the range $0 \leq p_i \leq l-1$, where l is the length of the hardened BF $\mathbf{b}_H$, is then generated by the PRNG. These numbers p_i will be used as the bit positions to set to 1 in $\mathbf{b}_H$.

3 OUR NOVEL HARDENING TECHNIQUES

As we experimentally show in Sect. 4, existing hardening techniques can potentially reduce linkage quality while improving privacy. To this end, we propose two techniques that use bits in the original BF to generate a hardened BF using XORing. XORing of bits ensures it is not possible to recover the original bit values. We next describe our two hardening techniques in more details.

3.1 Windowing based XORing (WXOR)

This method focuses on applying a sliding window approach to compute hardened BFs. In this approach two sliding windows W_1 and W_2 of a certain window size w are iteratively moved along the original BF $\mathbf{b}$ of length l. In each iteration the window W_1 starts at position p, with $0 \leq p < l - w$, and the window W_2 is positioned at bit position $(p+1)$ *mod* l, where the modulo (*mod*) function is used to 'wrap around' the last position of the second window W_2.

In a given iteration i a bit pattern from each window is extracted. These two bit patterns are then XORed together. The bit values in the computed XORed bit pattern are used to set the corresponding positions in the BF $\mathbf{b}$ according to the window W_1. After the XORing process the two windows W_1 and W_2 are moved by one bit position (step). We repeat these steps until we processed the whole original BF. This updated BF $\mathbf{b}$ is then used as the hardened BF $\mathbf{b}_H$. Fig. 2 shows an example of this hardening technique.

In this hardening technique each window is moved $(l - w + 1)$ iterations and each bit position in the hardened BF $\mathbf{b}_H$ is updated w times. If the window size w is small then longer runtime is required to apply the hardening since more iterations are required to process the original BF. However, as we also show in Sect. 4, a small w potentially allows more accurate similarity calculations upon the hardened BF since bit positions are updated according to the XORed value of a smaller number neighbouring bits. On the other hand, a large window size w will make the hardening process more efficient, but will be less accurate due to the large number of bits that will

Table 1: The average number of unique values in the different attributes of the different data set pairs used in the evaluation.

Data set name	Number of records	First Name	Last Name	Birth Year	Street address	City	Zip code
CLN-500K	500,000	40,405	77,160	101	469,139	756	838
CLN-100K	100,000	14,699	26,544	88	97,193	706	792
DRT-100K	100,000	14,870	26,796	87	96,289	699	784
Eurostat	24,978	2,169	1,029	104	3,022	–	605

be changed. Furthermore, if this hardening technique is to be used in a linkage protocol, the database owners only need to agree on the window size w that will be used in the hardening process.

3.2 Re-sampling based XORing

Our second hardening technique uses re-sampling of bit positions from the original BF **b** and applies XORing upon these bits. In this approach we use a random sampling process with replacement where we apply the sampling step l times to generate a hardened BF of length l. In each sampling step k, with $0 \leq k \leq l-1$, we randomly select two bit positions p_i and p_j from the original BF **b**, with $0 \leq i, j \leq l-1$. Next, the bit values $\mathbf{b}[p_i]$ and $\mathbf{b}[p_j]$ are XORed (⊕) and the resulting bit value is used to set the position k in the hardened BF $\mathbf{b}_H$, where $\mathbf{b}_H[k] = \mathbf{b}[p_i] \oplus \mathbf{b}[p_j]$.

In this approach, random sampling with replacement ensures the bit positions are selected with equal probability and the selection of two bit positions are independent. However, in this technique if the original BFs are hashed with more q-grams, i.e. contain more 1-bits, then the hardened BF will have more 0-bits due to XORing. This can potentially lower similarities thus leading to false negatives. In a PPRL protocol, the only parameter that needs to be agreed by the database owners for applying this hardening technique is a random seed value to ensure they apply the same random sampling of bit positions. We next compare our proposed techniques with the existing hardening techniques reviewed in Sect. 2.

4 EXPERIMENTAL EVALUATION

We used four data set pairs for experiments as summarised in Table 1. The first three data set pairs are based on the real North Carolina Voter Registration database (NCVR) (see: http://dl.ncsbe.gov/), where we extracted records from the April 2018 and October 2019 NCVR snapshots. CLN-500K and CLN-100K are 'clean' data sets that have at most two attribute values that are different for a given voter across the two NCVR snapshots. In the DRT-100K (dirty) data set each pair of records of the same voter has between one and three attribute values that are different between the two NCVR snapshots. The last data set pair, Eurostat, is a synthetic European census database (available at: https://ec.europa.eu/eurostat/cros/content/job-training_en) which is generated to represent the real observations of the decennial census.

A unique record identifier in these data sets allows us to identify true matching records that refer to the same entity across two data sets. The first three data set pairs in Table 1 have 100% overlap between entities while Eurostat has 97% overlap between entities with 62% of entities having errors and missing values in their records.

To evaluate linkage quality we simulated a three-party PPRL protocol [14] where we used Soundex [3] based phonetic blocking on FirstName and LastName attributes. Following [9], in the classification step we used Dice coefficient with a threshold t ranging from 0.1 to 1.0, in 0.1 steps, and assessed precision and recall [3].

Table 2: Average runtimes for different numbers of attributes encoded using the CLN-500K data sets. We show total encoding plus hardening runtimes in seconds, and the overhead of hardening techniques in percentages compared to no hardening.

Hardening technique	Runtime (sec)					
	One attribute		Two attributes		Four attributes	
No hardening	121		341		832	
Balancing	384	(217%)	812	(138%)	1,325	(59.2%)
XOR folding	134	(10.7%)	353	(3.5%)	844	(1.4%)
Salting	125	(3.3%)	593	(73.9%)	984	(18.3%)
Rule 90	392	(224%)	878	(157%)	1,475	(77.2%)
Markov chain	782	(546%)	1,595	(368%)	3,527	(324%)
BLIP	263	(117%)	593	(73.9%)	984	(18.3%)
Re-hashing	412	(240%)	906	(165%)	1,526	(83.4%)
WXOR	398	(228%)	880	(158%)	1,483	(78.3%)
Re-sampling	148	(22.3%)	384	(12.6%)	862	(3.6%)

To evaluate the privacy we used the cryptanalysis attack by Christen et al. [4]. This attack aligns frequent BFs and plain-text values in a public database to allow re-identification of the most frequent values encoded in these BFs. We conducted this attack assuming one file in a data set pair is the encoded BF database while the other represents the public database. We evaluate the re-identification accuracy in terms of the percentages of (1) correct guesses with 1-to-1 matching (1-1 corr), (2) correct guesses with 1-to-many (1-m corr) matching, (3) wrong guesses (Wrong), and (4) no guesses (No), where these four percentages sum to 100.

As in earlier PPRL work [14], we set the BF parameters as l =1000 bits, $k = 30$, $q = 2$, and used cryptographic long-term key [9] as BF encoding. We used BirthYear values for salting and set the chain length $c = 1$ in Markov chain based hardening. Following [12] and [8], we set flip probability $f = 0.05$ in BLIP, and 8, 16, 3 for w, s, and k_{re}, respectively, in re-hashing. For our window based hardening (WXOR) technique we set w to [1, 5, 10, 20, 50, 100].

We used Python (version 2.7) for implementations. We ran all experiments on a server with 64-bit Intel Xeon (2.4 GHz) CPUs, 128 GBytes of memory and running Ubuntu 14.04. The programs and data sets are available from the authors.

Results and Discussion Table 2 shows runtime results for BF encoding with the different hardening techniques applied. As can be seen, for the hardening techniques that work on the actual BFs (such as balancing, XOR folding, Rule 90, and BLIP), including our proposed techniques, the relative overhead becomes less as more attributes, and therefore more q-grams, are being hashed because these techniques are independent of the number of q-grams that are hashed. Markov chaining requires significantly longer runtimes because the total number of q-grams that are being encoded is increased through the selection of extra q-grams.

Figure 3 shows q-gram based similarities versus corresponding similarities on our hardening techniques. As can be seen, the WXOR and re-samping based hardened BFs allow accurate similarity calculations even when BFs are encoded with increasing numbers of attributes. However, we noted that in WXOR a larger window size ($w > 50$) can decrease the corresponding BF similarities of q-gram similarities (> 0.7) by 20% (plots not shown due to limited space).

Table 3: Area under the curve (AUC) values for different hardening techniques with different data sets and different numbers of attributes.

Data set	Number of attributes	No hardening	Balancing	XOR folding	Salting	Rule 90	Markov chain	BLIP	Re-hashing	WXOR	Re-sampling
CLN-100K	3	0.876	0.851	0.852	0.872	0.856	0.525	0.839	0.856	0.873	0.872
	4	0.957	0.961	0.967	0.957	0.973	0.683	0.937	0.878	0.973	0.971
DRT-100K	3	0.743	0.731	0.732	0.706	0.741	0.488	0.714	0.707	0.743	0.740
	4	0.831	0.873	0.876	0.830	0.882	0.472	0.785	0.711	0.886	0.884
Eurostat	3	0.987	0.984	0.983	0.982	0.983	0.707	0.975	0.981	0.985	0.985
	4	0.988	0.988	0.987	0.989	0.987	0.511	0.986	0.987	0.989	0.989

Table 4: Average percentages of 1-1 corr, 1-m corr, Wrong, and No guesses, shown respectively, with different numbers of attributes.

Number of Attributes	No hardening	Balancing	XOR folding	Salting	Rule 90	Markov chain	BLIP	Re-hashing	WXOR	Re-sampling
1	5 / 8 / 11 / 76	3 / 2 / 6 / 89	2 / 1 / 6 / 91	2 / 2 / 4 / 92	2 / 8 / 12 / 78	4 / 12 / 6 / 78	2 / 12 / 2 / 84	3 / 11 / 27 / 59	1 / 1 / 3 / 95	1 / 2 / 4 / 93
2	1 / 8 / 2 / 89	0 / 8 / 4 / 88	0 / 3 / 8 / 89	0 / 2 / 2 / 96	0 / 3 / 8 / 89	0 / 12 / 2 / 86	0 / 12 / 4 / 84	0 / 8 / 4 / 88	0 / 1 / 1 / 98	0 / 1 / 2 / 97

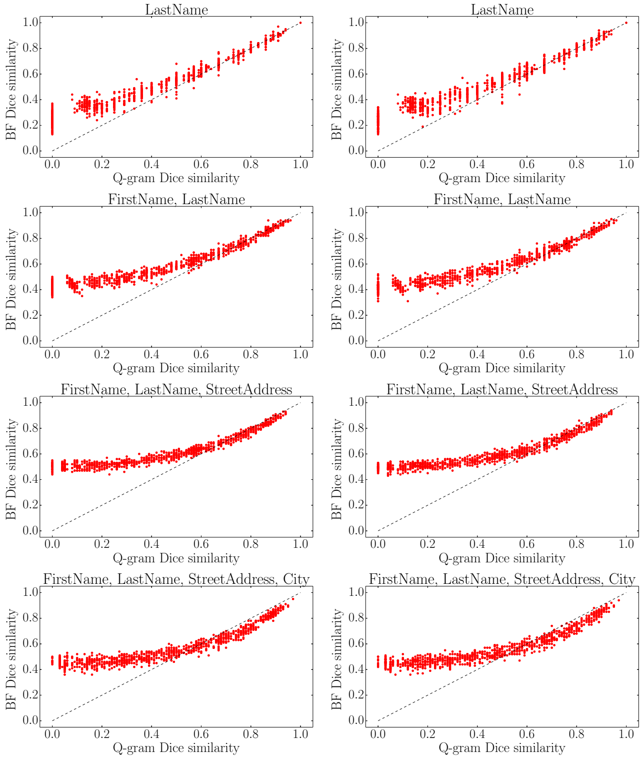

Figure 3: Similarity plots of (left) WXOR (w = 1) and (right) re-sampling hardening techniques for the CLN-100K data set.

Table 3 shows linkage quality results in terms of area under the curve value (AUC) calculated based on precision and recall for different threshold values. We noted for different numbers of attributes our hardening techniques resulted in higher AUC values compared to other hardening techniques. Markov hardening shows the lowest AUC values due to the addition of extra q-grams which potentially increases BF similarities leading to many false positives.

Finally, Table 4 shows average re-identification results for different hardening techniques. As can be seen, our proposed hardening techniques resulted in the lowest correct 1-1 re-identifications even when BFs are encoded with one attribute value. This indicates the XORing of bit values will likely distort the frequency distribution of BFs making a frequency based cryptanalysis attack more difficult. In the attack for three and four attributes all hardening techniques resulted in 100% No re-identification guesses.

5 CONCLUSION AND FUTURE WORK

We have introduced two novel hardening techniques for Bloom filter (BF) encoding based privacy-preserving record linkage (PPRL). Our techniques used two different bit XORing methods to harden BFs. Our experimental evaluation showed that our proposed techniques can outperform existing hardening techniques in terms of linkage quality and privacy. As future work, we aim to theoretically analyse our techniques and to assess privacy with other privacy attacks. Extending our proposed hardening techniques to other PPRL encoding techniques is another future research avenue.

REFERENCES

[1] J. Boyd et al. 2015. Accuracy and completeness of patient pathways–the benefits of national data linkage in Australia. *BMC health services research* (2015).
[2] J. Chen, S. Swamidass, et al. 2005. ChemDB: a public database of small molecules and related chemoinformatics resources. *Bioinformatics* 21, 22 (2005), 4133–4139.
[3] P. Christen. 2012. *Data Matching–Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Springer.
[4] P. Christen, T. Ranbaduge, et al. 2018. Precise and Fast Cryptanalysis for Bloom Filter based Privacy-Preserving Record Linkage. *IEEE TKDE* 31, 11 (2018).
[5] C. Dwork. 2006. Differential privacy. *ICALP* (2006), 1–12.
[6] M. Kuzu, M. Kantarcioglu, et al. 2011. A Constraint Satisfaction Cryptanalysis of Bloom Filters in Private Record Linkage. In *PET*. Waterloo, Canada, 226–245.
[7] F. Niedermeyer, S. Steinmetzer, et al. 2014. Cryptanalysis of Basic Bloom Filters Used for Privacy Preserving Record Linkage. *JPC* 6, 2 (2014), 59–79.
[8] R. Schnell. 2015. Privacy-preserving Record Linkage. In *Methodological Developments in Data Linkage.* John Wiley & Sons, Inc., UK, 201–225.
[9] R. Schnell, T. Bachteler, and J. Reiher. 2009. Privacy-preserving record linkage using Bloom filters. *BMC Med Inform Decis Mak* 9 (2009).
[10] R. Schnell and C. Borg. 2018. Hardening encrypted patient names against cryptographic attacks using cellular automata. In *ICDMW DINA*.
[11] R. Schnell and C. Borgs. 2016. Randomized Response and Balanced Bloom Filters for Privacy Preserving Record Linkage. In *ICDMW*. Barcelona, 218–224.
[12] R. Schnell and C. Borgs. 2016. XOR-Folding for Bloom Filter-based Encryptions for Privacy-preserving Record Linkage. *German Record Linkage Center* (2016).
[13] D. Vatsalan, P. Christen, and V. Verykios. 2013. A Taxonomy of Privacy-Preserving Record Linkage Techniques. *Information Systems* 38, 6 (2013).
[14] D. Vatsalan, Z. Sehili, P. Christen, and E. Rahm. 2017. *Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges.* Springer.
[15] A. Vidanage, T. Ranbaduge, et al. 2019. Efficient Pattern Mining based Cryptanalysis for Privacy-Preserving Record Linkage. In *IEEE ICDE*.
[16] S. Wolfram. 2002. *A new kind of science.* Wolfram media Champaign.