Generating-Set Evaluation of Bloom Filter Hardening Techniques in Private Record Linkage

Karin Mortl and Rinku Dewri

Department of Computer Science, University of Denver, CO, USA
karin.mortl@du.edu, rdewri@cs.du.edu

Abstract. Private record linkage is an active field of research targeted towards linking data sets from two or more sources, while preserving the privacy of contained sensitive content. With computation and communication efficiency as other two important requirements in such a process, much attention has been given to use Bloom filters for fast encoding of data records, while maintaining privacy of the records at the same time. A number of techniques to modify a typical Bloom filter have also appeared and addresses the need to harden them against known attacks. However, the field significantly lacks quantitative measures of the privacy level introduced by such techniques. In this work, we motivate and propose the *generating-set amplification factor* measure to bridge some of this gap. This privacy measure aims to capture the level of uncertainty that a hardening technique introduces between its output and the input used to create a Bloom filter. We provide algorithms to compute the measure and provide an empirical assessment of the state-of-the-art Bloom filter hardening techniques with respect to the measure. Our assessment shows that current techniques may still be retaining much of the characteristics of the input, although attacks to exploit them are yet to appear.

Keywords: Private record linkage · Bloom filter · Generating set · Privacy evaluation

1 Introduction

Private record linkage is the process of linking records of individuals in the absence of unique identifiers, with a high accuracy, and while preventing access to clear text information about an individual. Quasi-identifiers built from demographic information are often used in such a process. Further, clear text access to such information is prevented through the use of different encoding techniques [13, 21, 26] or secure protocols [6, 17, 27]. Due to the possible presence of data entry errors, a record linkage method must be able to link records based on approximated similarity rather than exact matching.

Bloom filter based methods have become actively researched and adopted mechanisms in this domain [1, 7, 11, 22, 28]. Bloom filters provide a succinct representation of a data record and can be tailored to provide indirect measurements

of the similarity of two data records. However, their vulnerability to many attacks [4, 16, 15, 18, 19, 31–33] have also resulted in a number of hardening techniques [14, 20, 23–25]. A hardening technique operates on a Bloom filter encoding with the objective of diffusing frequency information. Unfortunately, hardening techniques themselves have been shown to be vulnerable, raising the need to have better evaluation of the privacy preserving capabilities of proposed methods.

Quantitative privacy measures for Bloom filters in record linkage are rare. While there exists techniques that are yet to be shown vulnerable to an attack, there also does not exist much assessment on why hardening methods can resist an attack. Recent studies have used measures based on the frequency distribution of bits in a Bloom filter, but they are mostly useful in a comparative setting [10]. In this work, we propose a privacy measure that can inform whether a hardening technique can produce an output that detaches the relationship between a data record and its Bloom filter representation. More specifically, we analyze if the output of a standard hardening technique can be generated from a Bloom filter created from a much larger input set than was actually used. To this end, we make the following contributions in this work. We introduce the notion of a *generating-set amplification factor* that indicates the extent to which a Bloom filter's input set can be amplified and still result in the same hardened output. We then provide algorithms to compute this measure for standard hardening techniques such as balancing [23], XOR-folding [24], Rule90 [25], and random noise addition [8, 20, 23]. This is detailed in Section 3. Since the efficacy of these methods often depend on parametric choices (e.g. probability of distortion), we supplement our work with a theoretical analysis on when probabilistic methods are likely to fare well in terms of maintaining high linkage quality (Section 4). Finally, we provide empirical evidence using real world data records that, except for a select few scenarios, most hardening techniques still retain a strong correlation between the input set of a filter and the hardened output (Section 5). For the ones that do provide some level of privacy (in terms of the amplification factor), we show that a high linkage quality can be obtained, but requires careful setup of the similarity matching thresholds. Besides the aforementioned sections, Section 2 provides a brief background on record linkage using Bloom filters and hardening techniques, and puts related work in context, and Section 6 concludes the paper with references to future work.

2 Background and Related Work

A private record linkage aims to join two databases of records without clear text access to the attributes of either database [12, 29]. The presence of typographic errors in data records creates an additional challenge in this task. Therefore, the standard approach is to follow approximate matching principles where a record is converted into multiple q-grams (q sized sequences of contiguous characters), and the q-gram sets are instead compared privately to obtain a similarity score [5]. Bigrams ($q = 2$) and trigrams ($q = 3$) are typically used. Approximate matching

relies on a provided similarity scoring function and a similarity threshold, and matches are considered valid only if the similarity score crosses the threshold.

The use of Bloom filters to perform private comparison of two q-gram sets has received much attention due to their low communication and computation costs. A Bloom filter is a m-bit binary array, initialized with all zeros. An element is inserted into it using k pseudo-random hash functions, each of which maps the element to one of the m positions, which is then set to one. An element can therefore be tested for membership in the Bloom filter by checking if the bits at its hashed positions are set or not. Bloom filters can have false positives (an element passes the membership test although it was not inserted), but the rate can be controlled by proper choice of m and k. For a Bloom filter B, we use the notation $B[i]$ to indicate the bit at position $i = 0...m-1$ of the binary array.

2.1 Linkage with Bloom filters

A Bloom filter can be used to obtain an encoding of the q-gram set of a record by treating each q-gram as an element to be inserted in the filter [21]. Similarity comparisons are then performed on the Bloom filters instead of the q-gram sets. The primary objective here is to hide the exact q-grams of a record, and yet be able to determine the extent of overlap between two such sets. A frequently used similarity function is the Dice coefficient metric, given as

$$\mathsf{Dice}(B, B') = \frac{2 \times \mathsf{bitsum}(B \text{ AND } B')}{\mathsf{bitsum}(B) + \mathsf{bitsum}(B')}, \tag{1}$$

where AND is the bitwise-AND operation, and bitsum is the sum of all bit values in a binary array. A Dice score always falls in $[0, 1]$, where a score of zero implies that there are no positions where the bit is set in both filters, while a score of one implies that the two filters have the same bit value in all positions. Another used measure is the Jaccard coefficient, which is related to the Dice coefficient as $\mathsf{Dice}(B, B')/(2 - \mathsf{Dice}(B, B'))$. Similarity thresholds are often set in an ad-hoc manner, with a value such as 0.8 being a common choice [2, 10]. It is often assumed that data set holders will create the Bloom filters from their records using an agreed upon configuration (m, k, hash functions, and hardening method), and send it to a third party for the scoring and final matching.

2.2 Hardening Bloom filters

Although Bloom filters are efficient constructions and have been found to be useful in capturing the q-gram similarities, they are also vulnerable to frequency attacks. Frequency attacks often leverage the distribution of the set bits across the multiple Bloom filters created from a data set [3, 16]. Therefore, a number of hardening methods have been proposed to obfuscate or distort a Bloom filter further. We focus on the following common techniques in this work.

Balancing. A balanced Bloom filter is created by doubling its size, copying the original bits into the additional positions, inverting the extended bits, and

then applying a permutation on the extended filter [23]. Balancing produces a filter with equal number of zeros and ones.

XOR-folding. XOR-folding performs a bitwise XOR operation using the two halves of a Bloom filter, and outputs the resulting bit string [24]. This results in a string that is half the size of the input Bloom filter, and scoring is performed on the XOR-folded strings. We refer to the locations of two XOR-ed bits as *dual locations* of each other, and use the notation $\bar{i}$ to imply the dual location of position i. XOR-folding can lose set bit information if two XOR-ed bits are both one.

RULE90. RULE90 also creates a resulting bit string using an XOR operation, but uses positions $(i-1) \mod m$ and $(i+1) \mod m$ to obtain the resulting bit for position i [25]. As such, the output of RULE90 is of the same length as the original Bloom filter.

The above three methods are deterministic in nature and always produce the same output for a given Bloom filter. The following methods are probabilistic, and parameterized by a probability value p. These methods independently transform each bit of a Bloom filter to obtain a distorted version.

Random Bit Set (RBS). Irrespective of the original bit value at a position, RBS sets a position with probability p and retains the original value with probability $(1-p)$ [20]. RBS can therefore increase the number set bits in a Bloom filter, possibly creating more false positives.

Random Bit Flip (RBF). At each position, RBF inverts the existing bit with probability p and retains it with probability $(1-p)$ [23]. If a Bloom filter is set in more than half the positions, RBF can be expected to reduce the number of total set bits in a filter.

Permanent Randomized Response (RAPPOR). At each position, RAPPOR sets the bit to zero with probability $\frac{p}{2}$, sets it to one with probability $\frac{p}{2}$, and retains it with probability $(1-p)$ [8]. RAPPOR is proven to be able to provide ϵ_∞-differential privacy, with $\epsilon_\infty = 2k \ln(\frac{2}{p} - 1)$, with respect to two different Bloom filters producing the same output string. However, small p values must be used with a small number of hash functions (as small as $k = 1$) to obtain reasonable differential privacy guarantees in this method.

2.3 Privacy measures

While a number of hardening methods have been proposed for use with Bloom filters, metrics that evaluate their efficacy in preserving the privacy of the records are rare [30]. Bloom filter hardening methods are mostly relied upon based on a "test of time" where modifications are proposed after a method is found to be vulnerable to an attack. We list here two metrics based on the frequency distribution of set bits that have been used in a recent study [10]. Both of these metrics operate on a set of bit strings, possibly generated by a hardening method from records in a data set. Given a set of bit strings, each of length l, we use t_i to denote the total number of ones at position i across all the strings. Further, let $t = \sum_i t_i$ and $q_i = t_i/t$. We also include a distortion measure based on the number of set bits before and after a Bloom filter transformation.

Normalized Shannon entropy. Normalized Shannon entropy compares the entropy in each position relative to the maximum entropy possible if each position is equally used in the strings. It is computed as

$$1 + \frac{\sum_{i=0}^{l-1} q_i \log_2(q_i)}{\log_2(l)}, \tag{2}$$

and falls between zero (uniform distribution) and one (set bits only in one position).

Jensen-Shannon distance. Jensen-Shannon distance provides a distance measure between two probability distributions, namely the observed bit frequencies at different positions and an uniform distribution in our case. It is computed as

$$\left(\frac{1}{2l}\sum_{i=0}^{l-1}\log_2\left(\frac{\frac{1}{l}}{\frac{1}{2}(q_i+\frac{1}{l})}\right) + \frac{1}{2}\sum_{i=0}^{l-1} q_i \log_2\left(\frac{q_i}{\frac{1}{2}(q_i+\frac{1}{l})}\right)\right)^{0.5}, \tag{3}$$

and produces a value of zero when the distributions are identical, or one at the other extreme.

Mean distortion ratio. The distortion ratio of a Bloom filter is computed as the ratio of the number of bits set in the filter after transformation, relative to the number before transformation by a probabilistic hardening method. The mean distortion is obtained from the distortion ratios of a given set of Bloom filters and their hardened output.

3 Generating-Sets and Amplification

Let $\mathcal{U}$ be the universe of elements that can be potentially inserted into a Bloom filter. In the case of private record linkage, $\mathcal{U}$ is composed of all possible q-grams that can be created from letters, numbers and select punctuation symbols. Let $G \subseteq \mathcal{U}$ be the set of elements that is inserted into a Bloom filter B of size m. In other words, every element $g \in G$ is hashed using k hash functions $\mathcal{H}_i$, $i = 1...k$, to k positions in the Bloom filter and those positions are set to one. We then refer to G as the *generating-set* of the filter B, denoted as $\mathcal{GS}(B) = G$. When the hash functions choose positions uniformly at random, the size of the generating-set and the number of bits set in the filter (b_{set}) are approximately related as

$$|\mathcal{GS}(B)| \approx -\frac{m}{k}\log\left(1 - \frac{b_{set}}{m}\right). \tag{4}$$

Bit-frequency based measures of privacy for Bloom filters attempt to capture the distribution of set bits in a filter. They serve as an indirect evaluation of the existence or non-existence of frequently occurring bit patterns, which forms the basis for most known attacks against Bloom filters in private record linkage. A common approach in such methods is to identify frequently occurring bit patterns in a set of Bloom filters (resulting from a data set), and map them in full or parts to actual q-grams based on their known common usage frequencies. It is therefore expected that filters that do not carry much information on frequent

bit patterns, in other words, have a more uniform distribution of bits, are likely to be difficult to attack. However, such measures are only useful when comparing hardening approaches [10], and have difficulty in informing us about the difficulty introduced in attacks. If the bit-frequency measures of two hardened Bloom filters are "close," it is not known if they are similarly effective in disrupting a frequent pattern attack. In other words, the sensitivity of a specific measure to the elimination of specific patterns by a hardening method is yet to be analyzed. Some measures can in fact produce values indicating a near-uniform distribution for multiple hardening techniques, thereby making it difficult to assess their effectiveness. In the light of these observations, and that the primary purpose of a Bloom filter in private record linkage is to hide the elements used to create the filter, we seek an alternative measure of privacy that can inform us on the level of uncertainty that a standalone Bloom filter (with or without hardening) introduces in terms of inferring its generating-set.

3.1 Generating-set amplification factor

The objective of an attack on a Bloom filter is to infer its generating-set with best possible accuracy. A typical Bloom filter B has a false positive rate that depends on its size m, the number of hash functions k, and its generating-set size $s = |\mathcal{GS}(B)|$, and approximated as

$$\psi(m, k, s) \approx \left(1 - e^{-ks/m}\right)^k. \tag{5}$$

A false positive element implies that the element is not in the generating-set G, but the bit positions in the filter that would have been set by the element, if present, happens to be already set by a combination of elements in the generating-set. Hence, an element $u \in \mathcal{U}/G$ is a false positive when $\forall i = 1...k, B[\mathcal{H}_i(u)] = 1$. Although the rate of false positives can be controlled by choosing appropriate values for k and m, without knowledge of the generating-set, it is not possible to determine if an element is a true positive or a false positive. Therefore, a perfect attack targeted towards determining the elements used to create a Bloom filter (based on frequent patterns or otherwise) will produce the generating-set of the filter and the false positives. Note that we are not focusing on the attack methodology, but on the output producible by an attack of such nature. Effectively, this set is the largest set of elements that can produce a given Bloom filter B, denoted by $\mathcal{GS}_{max}(B)$, whose size can be approximated as

$$|\mathcal{GS}_{max}(B)| \approx |\mathcal{GS}(B)| + \psi(m, k, |\mathcal{GS}(B)|)|\mathcal{U}|. \tag{6}$$

Therefore, a generating-set may undergo amplification when one tries to infer it from a given Bloom filter. We capture this amplification as a factor, termed the *generating-set amplification factor*, and represented as

$$\mathsf{A}_f(B) = \frac{|\mathcal{GS}_{max}(B)|}{|\mathcal{GS}(B)|}. \tag{7}$$

As an example, a 1024-bit Bloom filter created from 50 unique elements using 25 hash functions and a universe of size 5000 has an amplification factor of 1.016. The amplification factor can be used as an indicator of a hardening technique's effectiveness in obfuscating the generating-set. A value of 1 (or close to 1) would indicate that the hardening technique failed to disrupt the relationship between the generating-set and its Bloom filter representation, effectively retaining statistical correlations between the two, open for exploitation in an attack. As the amplification factor increases, the correlations are expected to disperse since multiple generating-sets may produce the same Bloom filter. We therefore use the amplification factor as a measure of the privacy preserving effectiveness of a hardening technique. Although we are not proposing a new attack in this work, we do not make assumptions on the secrecy of parameters used to create the Bloom filters (size, number of hash functions, and the hash functions themselves) since attacks already exist that do not need knowledge of these parameters [33]. In fact, we work under an oracle-supported model where the attacker can query the hash output of any element in the universe.

Eq. (6) can be used to estimate the amplification factor for a standard Bloom filter, but does not lend itself well for hardening techniques that remap the filter or modify it with randomization. For example, the XOR-folding technique may produce a filter representation that can be obtained from multiple combinations of bit values in specific positions of the original Bloom filter. Randomization methods such as RBS, RBF and RAPPOR generate probabilistic outputs that can be obtained from any Bloom filter. In the following subsections, we provide algorithms to determine the amplification factor for common hardening methods, and an alternative formulation for probabilistic methods.

3.2 Amplification factor in deterministic methods

Besides the standard non-hardened Bloom filter, we consider three deterministic methods to harden Bloom filters, namely BALANCED, XOR-folding and RULE90. For a non-hardened Bloom filter, $\mathcal{GS}_{max}$ can be obtained by checking if each element $u \in \mathcal{U}$ is present in the filter. Unless the filter's false positive rate is high, or the universe size is comparatively quite large, we do not expect a non-hardened Bloom filter to have an amplification factor much larger than 1. In our experimental evaluation, we use 1024-bit filters with 25 hash functions, to effectively have a false positive rate of 0.00005. With 4761 possible bigrams in the universe set, the average amplification factor in the non-hardened method is practically 1! The BALANCED hardening method uses a negative duplicate of a filter and performs a deterministic permutation on those bits. Since the steps of this method are reversible (reverse the permutation and take first half of the output), any element found to be in $\mathcal{GS}_{max}$ for the non-hardening method will also be found for BALANCED, and vice versa. The amplification factor for BALANCED will therefore be equal to that of the non-hardening method.

The XOR-folding and RULE90 methods generate their final output using an XOR operation on bits from different positions in the original Bloom filter. Since an XOR output of zero can be obtained either as $0 \oplus 0$ or $1 \oplus 1$, and an output

of one can be obtained either as $0 \oplus 1$ or $1 \oplus 0$, an element alone may affect the final output, but may not do so when grouped with other elements.

XOR-folding Consider a Bloom filter B with the generating-set G, whose XOR-folded bit string is B_{xor}, with $B_{xor}[i] = B[i] \oplus B[\bar{i}], i = 0...\lceil m/2 \rceil - 1$. Recall that $\bar{i}$ represents the dual location of position i in the filter. A brute force method to find $\mathcal{GS}_{max}$ of B is to take every possible superset of G from $\mathcal{U}$, create a Bloom filter from it, and then test if its XOR-folding equals B_{xor}; the largest superset of G that passes the test will give $\mathcal{GS}_{max}$. The method is exponential in size of the universe (less the size of G). However, we can prune the search using an impossibility condition. If $B_{xor}[i] = 1$, then the bit at position i or $\bar{i}$ must be zero, while the other position is one. If an element sets the zero position, then $B_{xor}[i]$ will become zero. Since a bit's value in B cannot be reverted back to zero once set, irrespective of how many other elements we add, $B_{xor}[i]$ can never become one again. Therefore, any element $u \in \mathcal{U}/G$ that would set a zero value position in B, and that position contributes to an already set bit in B_{xor}, cannot be present in $\mathcal{GS}_{max}$. On the other hand, if $B_{xor}[i] = 0$, then both positions i and $\bar{i}$ have the same value in B. When both positions are one, an element $u \in \mathcal{U}/G$ setting one or both of the positions have no effect on B_{xor}. When both positions are zero, $B_{xor}[i]$ may change to one if u sets only one of the two positions. However, in such a case, there could be another element that sets the other zero value position, thereby reverting $B_{xor}[i]$ back to its original value of zero. Hence, when $B_{xor}[i]$ is zero, elements may be added in groups to B without affecting the XOR-folding output.

We incorporate the above observations into a depth-first-search (DFS) of the set $\mathcal{U}/G$ to identify subsets that can be added to B without changing B_{xor}. As the first step, we remove all elements from $\mathcal{U}/G$ that would create the impossibility condition for any position i in B_{xor} with $B_{xor}[i] = 1$. With a *base set* of G, for each remaining candidate u, we initiate a DFS by creating a temporary Bloom filter B' from $G \cup \{u\}$ and obtain its XOR-folded version B'_{xor}. Due to the handling of the impossibility condition in the first step, mismatches between B_{xor} and B'_{xor} can only happen in positions with $B_{xor}[i] = 0$ and $B'_{xor}[i] = 1$, and not the other way. For every such mismatch position, we identify all elements (not already added) that when added to B' can reset the bit in B'_{xor} at the mismatch position back to zero. Note that while such elements can reset the bit in a mismatch position, they may also create further mismatches in other positions. Hence, the search continues into the next level with the set used to create B' as the base set, and the identified elements as candidates. If no mismatches exist between B_{xor} and B'_{xor} at any point of this recursive procedure, then the set of elements used to create the corresponding B' is noted, and the specific search branch is terminated. Effectively, when all elements from a noted set exist in a Bloom filter, its XOR-folded bit string equals the original folded string B_{xor}. We take the union of all the noted sets as $\mathcal{GS}_{max}$ of B.

RULE90 The approach used to find $\mathcal{GS}_{max}$ for a Bloom filter hardened with RULE90 is similar to that for XOR-folding. The distinction is in how dual locations are defined in the two methods. RULE90 uses the neighboring bits

Table 1. Probability of transformation of bit values. $p_{ij} = \Pr(i \to j)$.

RBS				RBF				RAPPOR			
p_{00}	p_{01}	p_{10}	p_{11}	p_{00}	p_{01}	p_{10}	p_{11}	p_{00}	p_{01}	p_{10}	p_{11}
$1-p$	p	0	1	$1-p$	p	p	$1-p$	$1-\frac{p}{2}$	$\frac{p}{2}$	$\frac{p}{2}$	$1-\frac{p}{2}$

at a position for the XOR operation, and produces an output equal in length to the Bloom filter. While the principles are the same, care should be taken when implementing the algorithm for RULE90. In XOR-folding, a bit position serves as the dual location of exactly one other position, while in RULE90, a bit position i serves as the dual location of two positions, $(i-2) \mod m$ and $(i+2) \mod m$. In XOR-folding, during the impossibility condition check in the first step, we can use a loop variable $i = 0...m-1$ to check if $B[i] = 0$ and $B[\bar{i}] = 1$, and eliminate elements that set the location i. When the loop variable reaches $\bar{i}$, the reverse condition gets checked by symmetry. For RULE90, if $B[i] = 1$ and $B[(i+2) \mod m] = 0$, then the location to avoid setting is $(i+2) \mod m$, otherwise it is i. When the loop variable reaches $(i+2) \mod m$, note that the locations being tested are now $(i+2) \mod m$ and $(i+4) \mod m$.

3.3 Amplification factor in probabilistic methods

Unlike in deterministic methods, probabilistic methods such as RBS, RBF and RAPPOR are parameterized by a probability value p that determines how a Bloom filter is transformed. Due to the probabilistic nature, even a Bloom filter with all bits set can get transformed to any given bit string, albeit with varying probabilities. Therefore, we use an alternative formulation for $\mathcal{GS}_{max}$ in such methods. A probabilistic transformation method parameterized by p effectively gives us the probability of transforming a zero bit to a one, and vice versa. We use p_{ij} to denote the probability of transforming a bit value i to a bit value j. Table 1 lists these probabilities for RBS, RBF and RAPPOR, in terms of the parameter p.

Any element $u \in \mathcal{U}$ when added to a Bloom filter B would set the bits at positions $l_i = \mathcal{H}_i(u), i = 1...k$. Given a transformed version of the filter, B_{trf}, we calculate the probability that the element was present in B as

$$\Pr(u \in B | B_{trf}) = \prod_{i=1}^{k} p_{1B_{trf}[l_i]}. \tag{8}$$

Given $G = \mathcal{GS}(B)$, we consider the probabilities corresponding to all elements $g \in G$ to be significant, no matter how small they are. In other words, any attack that successfully retrieves G using B_{trf} cannot discard these probabilities to be small; otherwise the attack fails. Under such a condition, all elements that have higher probabilities than elements in G will be retained as well. Therefore, we define $\mathcal{GS}_{max}$ as the set of all elements whose probabilities are greater or equal

to the minimum probability of elements in G.

$$\mathcal{GS}_{max}(B|B_{trf}) = \{u \in U | \Pr(u \in B|B_{trf}) \geq \min_{g \in G} \Pr(g \in B|B_{trf})\} \quad (9)$$

It is also possible to define $\mathcal{GS}_{max}$ under a different requirement wherein retrieving a specific fraction f of the elements in G may be considered sufficient under some attack. In such a case, we can change the inequality to be with respect to a quantile value instead of the minimum.

$$\mathcal{GS}_{max}(B|B_{trf}) = \{u \in U | \Pr(u \in B|B_{trf}) > \mathsf{Q}(1-f)\}, \quad (10)$$

where $\mathsf{Q}(q)$ returns a threshold value such that q fraction of values in $\{\Pr(g \in B|B_{trf})|g \in G\}$ are less than or equal to the returned value. This variation can be useful in evaluating if a hardening method can provide amplification when the entire generating-set is not an attack target.

4 Parameter Selection in Probabilistic Methods

Probabilistic methods distort a Bloom filter based on a parameter for bit manipulation, such as the probability parameter p. Higher values of p are likely to provide better privacy, but it comes at the expense of poor linkage quality. Earlier studies have reported on this where p values as small as 0.1 can lead to significant degradation in linkage quality [10]. However, we believe that choice of the parameter is linked with the threshold chosen for similarity acceptance. Nonetheless, it is not known how large of a p value can be accommodated for acceptable linkage quality even with a properly chosen similarity threshold. In this section, we seek to generate some guidance around this issue.

Consider two Bloom filters B and B' of size m, with their transformed versions as B_{trf} and B'_{trf}. For a given number of set bits $b_{set} > 0$ in the two original filters, the probability that the two filters have $b_{common} \in \{0, 1, .., b_{set}\}$ number of common positions where the bits are set can be obtained as

$$\Pr(b_{common}|b_{set}) = \frac{\binom{b_{set}}{b_{common}}\binom{m-b_{set}}{b_{set}-b_{common}}}{\binom{m}{b_{set}}}. \quad (11)$$

When $b_{set} = 0$ or $b_{common} > b_{set}$, this probability is zero. Since the number of set bits cannot be more than m, we normalize this probability as

$$\Pr(b_{common}|b_{set}, m) = \frac{\Pr(b_{common}|b_{set})}{\sum_{b_s=0}^{m}\sum_{b_c=0}^{b_s}\Pr(b_c|b_s)}. \quad (12)$$

If the filters B and B' have $b_{common\text{-}before}$ common bits, then the expected number of set bits in each filter, $b_{set\text{-}before}$, can be obtained as

$$b_{set\text{-}before} = \frac{\sum_{b=0}^{m} b \Pr(b_{common\text{-}before}|b, m)}{\sum_{b_s=0}^{m}\Pr(b_{common\text{-}before}|b_s, m)}. \quad (13)$$

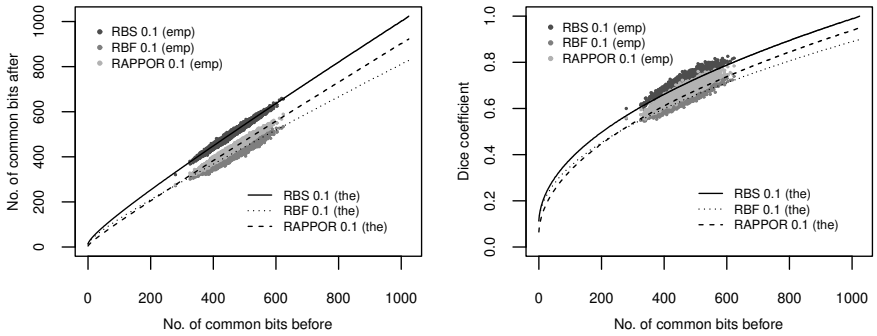

Fig. 1. Comparison of theoretical (the) and empirical (emp) computations for common bits in two Bloom filters (before and after transformation) and Dice coefficient values on the transformed filter. Filter size $m = 1024$ bits.

The denominator is necessary since we are conditioning on the event that the set bits result in exactly $b_{common\text{-}before}$ common bits. During transformation, these set bits remain set with a probability p_{11} and clear bits get set with a probability p_{01}. Therefore, the expected number of set bits in a transformed filter, $b_{set\text{-}after}$, is obtained as

$$b_{set\text{-}after} = p_{11} b_{set\text{-}before} + p_{01}(m - b_{set\text{-}before}). \tag{14}$$

We can then obtain the expected number of common bits between B_{trf} and B'_{trf} as

$$b_{common\text{-}after} = \frac{\sum_{b=0}^{m} b \Pr(b | b_{set\text{-}after}, m)}{\sum_{b_c=0}^{m} \Pr(b_c | b_{set\text{-}after}, m)}. \tag{15}$$

Fig. 1 (left) shows $b_{common\text{-}before}$ and $b_{common\text{-}after}$ computed from the above equations for RBS, RBF and RAPPOR with $p = 0.1$, and $m = 1024$ bits. For accuracy check, we also show 10,000 sample points for each method, obtained from Bloom filter ($m = 1024, k = 25$) encodings of real name and address strings (average of 46 bigrams in a string). The figure (right) also shows the Dice coefficient values computed from the estimated counts of the set and common bits after the transformation. This is effectively $b_{common\text{-}after}/b_{set\text{-}after}$. While the Dice coefficients are slightly underestimated, the trends follow accurately and relative differences across different methods are maintained.

The box-plots in Fig. 2 show the distribution summary of Dice coefficients for non-hardened Bloom filter pairs and ones where different transformations have been performed. Each summary is obtained from Dice coefficient values corresponding to a varying number of common set bits. Non-hardened Bloom filter pairs can have Dice coefficient values as low as zero (no common bits) and as high as one (all set bits are common). The middle 50% (interquartile range) of Dice scores fall in the range of $[0.5002, 0.8661]$, with a median score of

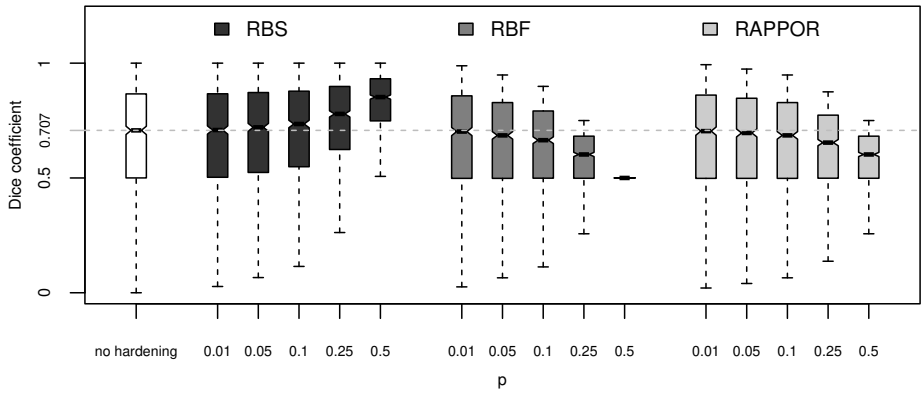

Fig. 2. Distribution (range and quantiles) of theoretically estimated Dice coefficient values.

0.7072. Depending on the hardening method and extent of distortion (choice of the parameter p), the range of scores become less dispersed. For example, with RBF and $p = 0.5$, the Dice scores are expected to become 0.5, irrespective of the number of common bits in the original Bloom filters. In other cases, such as RBS 0.5, RBF 0.25 and RAPPOR 0.5, the scores become significantly less dispersed than in the original Bloom filters. Good dispersion of scores is important to preserve a good linkage quality as it can provide for stronger distinction between matching and non-matching input pairs. We therefore need a quantitative method to assess if the range of Dice scores can become significantly compressed under a hardening method. We use a heuristic based on the interquartile range of Dice scores, wherein a method is deemed unsuitable for record linkage if the entire interquartile range of resulting scores fall above or below the median score in a non-hardened setting. The condition is effectively met if the first quartile value of the scores from a hardening method is greater than the median score in a non-hardened case, or if the third quartile value is smaller than the non-hardened median. Under this heuristic, we expect RBS 0.5, RBF 0.25, RBF 0.5 and RAPPOR 0.5 to be unsuitable for acceptable record linkage. Note that the tests are done on theoretically estimated Dice score distributions since the parametric decision is to be made before Dice scores on an actual data set are calculated. In a real use case, the number of common bits between two Bloom filters will fall in a much narrower range, compressing the distributions further.

5 Empirical Evaluation

In this section, we show an empirical evaluation of different hardening methods and their effectiveness with respect to the generating-set amplification factor measure. We also perform a sample linkage of two data sets and demonstrate

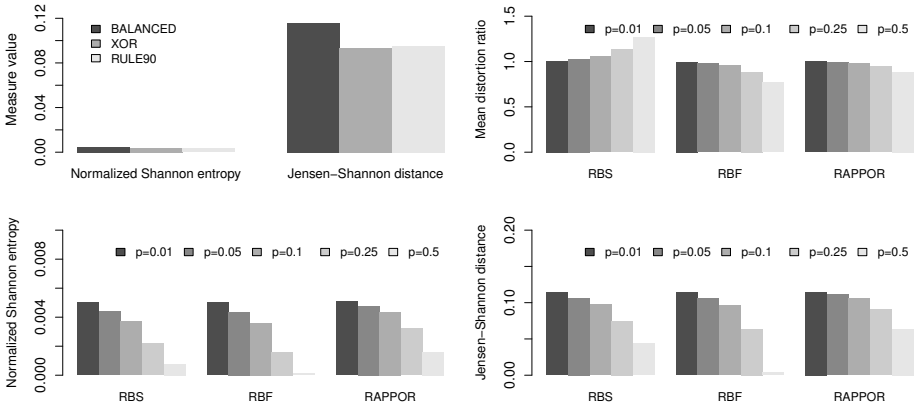

Fig. 3. Bit frequency based measures of privacy.

that linkage quality can be retained for high probabilistic distortions by properly setting the similarity threshold.

5.1 Setup

We use the North Carolina Voter Registration (NCVR) data set to create the data sets for our evaluation (www.ncsbe.gov). NCVR contains more than 7 million records with demographics data on individuals. We combine the attributes related to the name and address of an individual to form a single string representing a record in the data set. We create three data sets for use in the evaluation: (i) a data set of 10,000 records to assess the amplification factor induced by different methods (Sections 5.2 and 5.3), and (ii) two separate data sets, each with 10,000 records, but having an overlap of 1,000 records, to assess linkage quality under different methods (Section 5.4). Records are sampled uniformly at random from the entire NCVR data set.

Each record is converted to uppercase symbols and then split into a set of bigrams (46 bigrams on average in a record). Bloom filters of size $m = 1024$ bits are created from the bigram sets of the records, often referred to as record level filters [22], using $k = 25$ hash functions. Each hash function is instantiated using HMAC-SHA56, but uses a separate randomly generated 32-bit hash key [19]. A hash function's output is mapped to a position in a Bloom filter using the mod m operator on the hash output. Considering all uppercase letters, numbers and punctuations, we have a universe set of $69^2 = 4761$ bigrams.

We explore BALANCED, XOR-folding and RULE90 for deterministic hardening methods, and RBS, RBF and RAPPOR for probabilistic methods, with p values of 0.01, 0.05, 0.1, 0.25 and 0.5. Record linkage is performed by computing the Dice coefficient scores for all pairs of filters after applying a hardening method. For each record in one data set, we find the record in the other data set with the highest Dice coefficient score, and consider them a match if the score

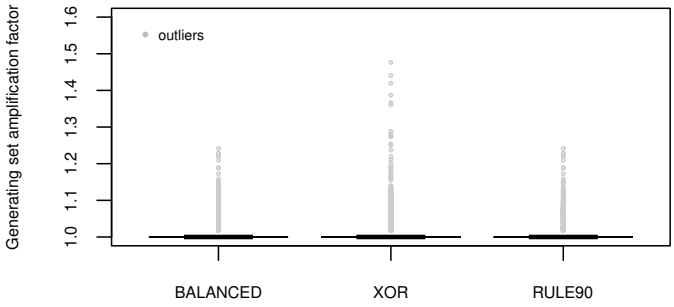

Fig. 4. Distribution of generating-set amplification factor for deterministic methods.

crosses a set threshold. Ties are broken randomly. Typically, blocking approaches are used to reduce the number of score comparisons to be performed, especially for large data sets [9]. We choose a threshold of $\mu + 6\sigma$ for each method, where μ and σ are respectively the mean and standard deviation of the computed Dice scores. While this method of threshold setting may not produce the best linkage in all cases, it gives us the platform to evaluate the linkage quality under an unbiased and uniform methodology. Linkage quality is then evaluated using the measures of precision (fraction of correct matches in identified matches) and recall (fraction of true matches correctly identified).

5.2 Bit frequency measures

Fig. 3 shows the different bit frequency based measures of privacy. For all methods, the normalized Shannon entropy measure is near zero, indicating an "almost" uniform distribution of bits. Even for the non-hardened case, the entropy value is close to zero (0.005). This observation aligns with observations made in other data sets in earlier studies [10]. The Jensen-Shannon distance reduces (indicating closer to uniform) when an obfuscation is applied to the original Bloom filters (e.g. XOR and RULE90) or when the level of distortion is increased in a probabilistic method. The non-hardened Jensen-Shannon distance is similar to the BALANCED method, at 0.116. RBF 0.5 has a near zero Jensen-Shannon distance, which conforms to the observation that Dice scores are uniform as per the theoretical analysis in Section 4. In terms of distortion ratio, the number of bits set is expected to increase in RBS with higher p values. However, the number reduces for RBF and RAPPOR. Both cases lead to more compression of the range of Dice scores; the probability of having a given number of common bits reduces/increases as the number of set bits reduce/increase.

5.3 Generating-set amplification factor

Fig. 4 shows the distribution of amplification factors when using the three deterministic methods. Recall that the amplification factor of a non-hardened case is

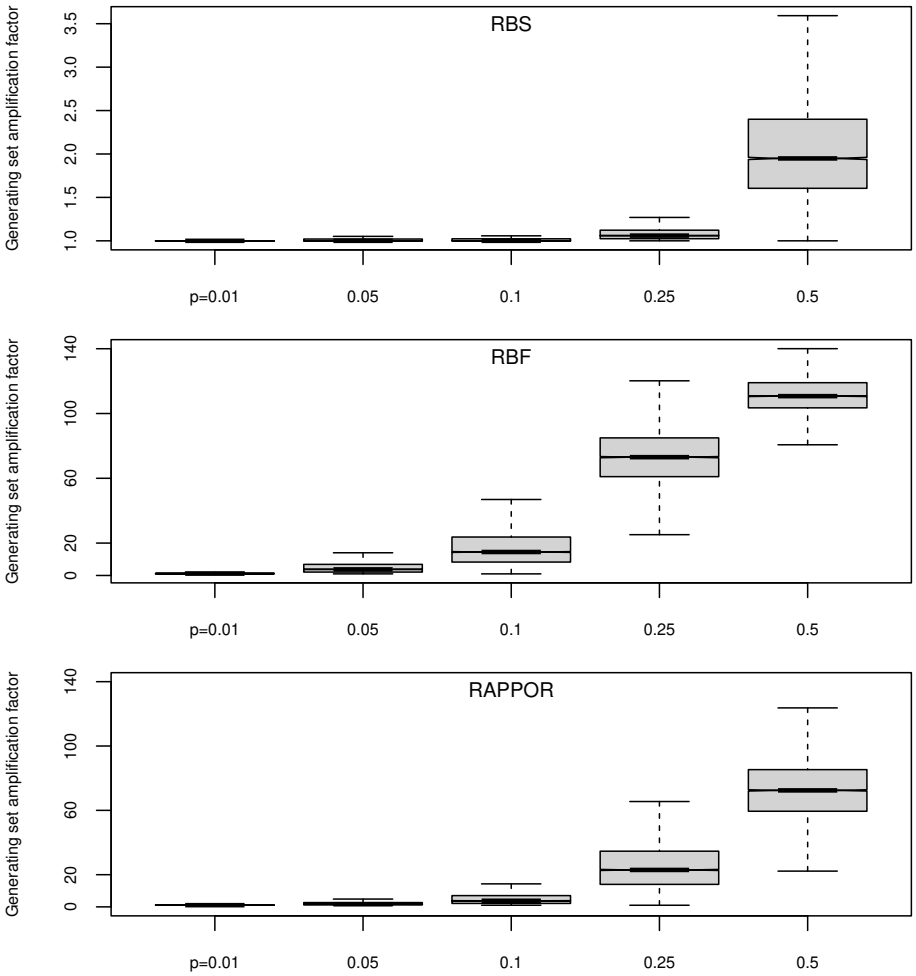

Fig. 5. Distribution of generating-set amplification factor for probabilistic methods. Outliers are not shown.

same as that of the BALANCED method. While some outliers are present in all cases, the amplification factors are close 1 in all three methods, implying that most of the produced bit strings are unique to the generating-set of the filter. While this is not unexpected for a non-hardened method and BALANCED (due to the small false positive rate), XOR-folding and RULE90 also produce unique mappings between generating-sets and output strings.

Fig. 5 shows the distribution of amplification factors for the three probabilistic methods. The amplification factors here are generated based on Eq. (9). The results here are more promising than deterministic methods, especially with higher p values. However, recall from Section 4 that we do not expect RBS 0.5,

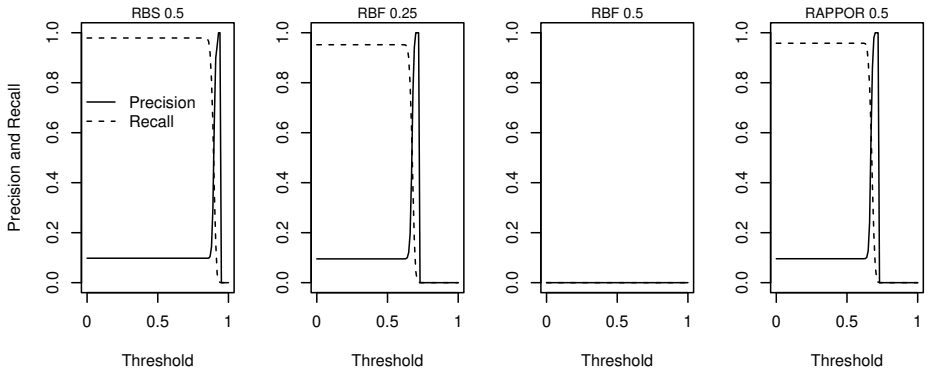

Fig. 6. Precision and recall curves for methods that are not expected to provide reasonable linkage quality per the theoretical analysis (Section 4).

RBF 0.25, RBF 0.5 and RAPPOR 0.5 to be able to produce reasonable linkage quality. With this consideration, the RBS method is unable to provide amplification of the generating-set at $p < 0.5$. RBF with $p = 0.1$ has a mean amplification factor of 17.89, which implies that the bit flipping successfully mixes the probability values of the generating-set elements and approximately 777 elements outside the set (16.3% of the universe). RAPPOR with $p = 0.25$ can provide a slightly higher amplification, but it needs to be able to retain the linkage quality at that level of distortion. Note that, for a given value of p, the probability of flipping a bit in RAPPOR is half that in RBF (Table 1). As such, the amplification we see in RBF 0.1 is similar to as in RAPPOR 0.25. p values lower than 0.1 fail to produce much amplification in all three methods. It is important to point that if Eq. (10) is instead used for determining the amplification factor, say with $f = 0.25$, even RBF 0.1 and RAPPOR 0.25 cannot generate high amplification factors. This is concerning since an attack that can recover 75% of the bigrams in the generating-set could be often sufficient to recreate the entire string.

5.4 Linkage quality

Fig. 6 shows the precision and recall curves for RBS 0.5, RBF 0.25, RBF 0.5, and RAPPOR 0.5, which are scenarios that lead to significant compression of the range of Dice scores as per our determination heuristic (Section 4). The curves are obtained by computing the precision/recall values for varying thresholds between 0 and 1, at 0.01 increments. It is evident that an acceptable balance between precision and recall is not possible in these scenarios, irrespective of the threshold. RBF 0.5 leads to heavy loss of variability in the scores, and hence matched pairings fail to be correct for all strings.

Table 2 lists the precision, recall and F1-score (harmonic mean of precision and recall) for the remaining methods. Recall that the thresholds are set using a computational method ($\mu + 6\sigma$). With the set thresholds, in most cases, the

Table 2. Linkage quality in terms of precision and recall for hardening methods. μ and σ are the mean and standard deviation of the computed Dice scores respectively.

Method	Threshold ($\mu + 6\sigma$)	Precision	Recall	F1-Score
no hardening	0.892	0.958	1.0	0.978
BALANCED	0.892	0.958	1.0	0.978
XOR	0.669	0.840	1.0	0.913
RULE90	0.631	*0.633*	1.0	0.775
RBS 0.01	0.893	0.961	1.0	0.980
RBS 0.05	0.894	0.975	1.0	0.987
RBS 0.1	0.896	0.992	1.0	0.996
RBS 0.25	0.907	0.998	*0.624*	0.768
RBF 0.01	0.884	0.964	1.0	0.982
RBF 0.02	0.854	0.989	1.0	0.994
RBF 0.1	**0.817**	**0.995**	**1.0**	**0.997**
RAPPOR 0.01	0.888	0.961	1.0	0.980
RAPPOR 0.05	0.873	0.976	1.0	0.988
RAPPOR 0.1	**0.853**	**0.980**	**1.0**	**0.990**
RAPPOR 0.25	**0.797**	**0.998**	**0.965**	**0.981**

methods are able to retrieve all the overlapping records, with a precision above 95%. The thresholds are found to adjust based on the distribution of the Dice scores; in RBS, Dice scores increase with more distortion, hence the threshold increases as well, while in RBF and RAPPOR, the scores decrease, leading to lowering of the threshold. RBS 0.25 has a significantly lower recall at the set threshold. Since the precision is high, it is expected that reducing the threshold can produce improvement in the recall without affecting the precision. For example, at a slightly higher threshold of 0.91, RBS 0.25 has a precision of 96% and recall of 96.9%. Similarly, a threshold of 0.7 can increase the precision of RULE90 to 94.3% while maintaining the recall at 100%.

5.5 Discussion

The Shannon entropy measure is not sensitive to changes created in a Bloom filter by the hardening methods. As such, this measure in not suitable for privacy evaluation in record linkage. Comparatively, the Jensen-Shannon distance measure reacts more to changes. Nonetheless, all values we obtained for the measure are typically low, which could lead to a misinterpretation of a method's effectiveness when viewed alone. Distortion ratio shows a change of $\pm 20\%$ at most, even at high p values such as 0.5. In summary, measures based on bit frequencies are generally not very sensitive to differences in the hardening methods. When viewed through the use of amplification factors, we see more prominent changes in the metric's value as more distortion is introduced in a Bloom filter. We also see that typically used small values of p in probabilistic methods are in fact not advisable due to their inability to diffuse the probabilistic relationship between generating-set elements and a transformed Bloom filter. In deterministic methods such as XOR-folding and RULE90, generating-set amplification is almost

non-existent. Both methods use two bit positions in a Bloom filter to derive a resulting bit, which seems to be insufficient. Use of low false positive rates and good pseudo-random hash functions appear to create issues for such methods in mixing the Bloom filter bits. While variants exist to include more bit positions in the operation, they have a detrimental effect on the linkage quality.

Although we show the linkage quality for different methods, note that most of the scenarios do not fare well in terms of the generating-set amplification factor measure. Nonetheless, methods such as RBF and RAPPOR do have few niche parameter settings (e.g. $p = 0.1$) that can provide both quality linkage and high amplification factors. However, getting these methods to perform well in record linkage is dependent on the choice of the set threshold. We showed two cases (RULE90 and RBS 0.25) where a small change in the set threshold significantly improves the obtained precision and recall. In general, linkage quality can be highly sensitive to the threshold when a hardening method compresses the range of similarity scores.

The amplification factor can provide general guidance on the difficulty of obtaining the exact generating-set of a Bloom filter. However, attacks can leverage additional filtering to eliminate elements that cannot sensibly be part of a record. For example, a hardening method may provide a high amplification factor, but most of the false q-grams that it introduces may not be combinable to form record strings. Therefore, it may not be sufficient for a method to introduce a high amplification factor, but the false elements have to be able to introduce uncertainty in terms of record reconstruction.

6 Conclusion and Future Work

A number of Bloom filter hardening methods have appeared in the past two decades, and have also been found to be insufficient for privacy protection. In this work, we explored the issue of quantifying the privacy guarantees of current Bloom filter hardening methods. Using the proposed generating-set amplification factor measure, we found that most methods generate output representations of Bloom filters that are unique to the input set of elements. Few probabilistic methods can offer an amplification of the input set with higher levels of distortion than traditionally performed. However, such methods can significantly compress the range of similarity scores, and require careful setting of the similarity cut-off to be able to produce a good linkage output. We also provided a heuristic to determine if a probabilistic hardening method can provide acceptable linkage quality for a specific level of distortion of the Bloom filters.

We have identified multiple gaps in the current research. Besides the need for hardening methods that can provide better generating-set amplifications, it is important to design guidelines for the choice of similarity thresholds with respect to the methods. Adhoc amplification may also not be useful and effort should be directed at creating hardening methods that can guarantee desired levels of amplification and with sensible false positive elements. Of course, equal effort is needed to ensure that these methods can retain a high linkage quality. Bloom fil-

ter based methods also require more theoretical assessments, primarily because their adoption is outpacing our understanding of their privacy assurances. Finally, more work in needed in the field of privacy measures for Bloom filters in record linkage to reveal different privacy related aspects of a Bloom filter.

References

1. Baker, D.B., Knoppers, B.M., Phillips, M., van Enckevort, D., Kaufmann, P., Lochmuller, H., Taruscio, D.: Privacy-preserving linkage of genomic and clinical data sets. IEEE/ACM Transactions on Computational Biology and Bioinformatics **16**(4), 1342–1348 (2019)
2. Brown, A.P., Randall, S.M., Boyd, J.H., Ferrante, A.M.: Evaluation of approximate comparison methods on Bloom filters for probabilistic linkage. International Journal for Population Data Science **4**(1), 1095 (2019)
3. Christen, P., Ranbaduge, T., Vatsalan, D., Schnell, R.: Precise and fast cryptanalysis for Bloom filter based privacy-preserving record linkage. IEEE Transactions on Knowledge and Data Engineering **31**(11), 2164–2177 (2019)
4. Christen, P., Schnell, R., Vatsalan, D., Ranbaduge, T.: Efficient cryptanalysis of Bloom filters for privacy-preserving record linkage. In: 2017 Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 628–640 (2017)
5. Churches, T., Christen, P.: Blind data linkage using n-gram similarity comparisons. In: 2004 Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 121–126 (2004)
6. Dewri, R., Ong, T., Thurimella, R.: Linking health records for federated query processing. Proceedings on Privacy Enhancing Technologies **2016**(3), 4–23 (2016)
7. Durham, E.A., Kantarcioglu, M., Xue, Y., Toth, C., Kuzu, M., Malin, B.: Composite bloom filters for secure record linkage. IEEE Transactions on Knowledge and Data Engineering **26**(12), 2956–2968 (2014)
8. Erlingsson, U., Pihur, V., Korolova, A.: Rappor: Randomized aggregatable privacy-preserving ordinal response. In: 2014 21st ACM Conference on Computer and Communications Security. pp. 1054–1067 (2014)
9. Franke, M., Sehili, Z., Rahm, E.: Parallel privacy-preserving record linkage using LSH-based blocking. In: 2018 3rd International Conference on Internet of Things, Big Data and Security. pp. 195–203 (2018)
10. Franke, M., Sehili, Z., Rohde, F., Rahm, E.: Evaluation of hardening techniques for privacy-preserving record linkage. In: 2021 24th International Conference on Extending Database Technology. pp. 289–300 (2021)
11. Guesdon, M., Benzenine, E., Gadouche, K., Quantin, C.: Securizing data linkage in french public statistics. BMC Medical Informatics and Decision Making **16**(1), 129 (2016)
12. Hall, R., Fienberg, S.E.: Privacy-preserving record linkage. In: 2010 International Conference on Privacy in Statistical Databases. pp. 269–283 (2010)
13. Karakasidis, A., Verykios, V.: Privacy preserving record linkage using phonetic codes. In: 2009 4th Balkan Conference in Informatics. pp. 101–106 (2009)
14. Kirsch, A., Mitzenmacher, M.: Less hashing, same performance: Building a better Bloom filter. In: 2006 European Symposium on Algorithms. pp. 456–467 (2006)
15. Kroll, M., Steinmetzer, S.: Automated cryptanalysis of bloom filter encryptions of health records. In: German Record Linkage Center, Working Paper Series. No. WP-GRLC-2014-05 (2014)

16. Kuzu, M., Kantarcioglu, M., Durham, E., Malin, B.: A constraint satisfaction cryptanalysis of Bloom filters in private record linkage. In: Privacy Enhancing Technologies. pp. 226–245 (2011)
17. Lazrig, I., Ong, T.C., Ray, I., Ray, I., Jiang, X., Vaidya, J.: Privacy preserving probabilistic record linkage without trusted third party. In: 2018 16th Annual Conference on Privacy, Security and Trust. pp. 1–10 (2018)
18. Mitchell, W., Dewri, R., Thurimella, R., Roschke, M.: A graph traversal attack on Bloom filter-based medical data aggregation. International Journal of Big Data Intelligence **4**(4), 217–226 (2017)
19. Niedermeyer, F., Steinmetzer, S., Kroll, M., Schnell, R.: Cryptanalysis of basic Bloom filters used for privacy preserving record linkage. Journal of Privacy and Confidentiality **6**(2), 59–79 (2014)
20. Schnell, R.: Privacy-preserving record linkage. In: Methodological Developments in Data Linkage, pp. 201–225 (2015)
21. Schnell, R., Bachteler, T.: Privacy-preserving record linkage using Bloom filters. BMC Medical Informatics and Decision Making **9**(1), 41 (2009)
22. Schnell, R., Bachteler, T., Reiher, J.: A novel error-tolerant anonymous linking code. SSRN Electronic Journal (2011). https://doi.org/http://dx.doi.org/10.2139/ssrn.3549247
23. Schnell, R., Borgs, C.: Randomized response and balanced Bloom filters for privacy preserving record linkage. In: 2016 16th International Conference on Data Mining Workshops. pp. 218–224 (2016)
24. Schnell, R., Borgs, C.: XOR-folding for Bloom filter-based encryptions for privacy-preserving record linkage. SSRN Electronic Journal (2016). https://doi.org/http://dx.doi.org/10.2139/ssrn.3527984
25. Schnell, R., Borgs, C.: Hardening encrypted patient names against cryptographic attacks using cellular automata. In: 2018 International Conference on Data Mining Workshops. pp. 518–522 (2018)
26. Smith, D.: Secure pseudonymisation for privacy-preserving probabilistic record linkage. Journal of Information Security and Applications **34**, 271–279 (2017)
27. Stammler, S., Kussel, T., Schoppmann, P., Stampe, F., Tremper, G., Katzenbeisser, S., Hamacher, K., Lablans, M.: Mainzelliste secureepilinker (mainsel): Privacy-preserving record linkage using secure multi-party computation. Bioinformatics **38**(6), 1657–1668 (09 2022)
28. Vatsalan, D., Christen, P., Rahm, E.: Scalable privacy-preserving linking of multiple databases using counting Bloom filters. In: 2016 16th International Conference on Data Mining Workshops. pp. 882–889 (2016)
29. Vatsalan, D., Christen, P., Verykios, V.: A taxonomy of privacy-preserving record linkage techniques. Information Systems **38**(6), 946–969 (2013)
30. Vatsalan, D., Sehili, Z., Christen, P., Rahm, E.: Privacy-preserving record linkage for big data: Current approaches and research challenges, pp. 851–895. Springer International Publishing, Cham (2017)
31. Vidanage, A., Christen, P., Ranbaduge, T., Schnell, R.: A graph matching attack on privacy-preserving record linkage. In: 2020 29th ACM International Conference on Information & Knowledge Management. pp. 1485–1494 (2020)
32. Vidanage, A., Ranbaduge, T., Christen, P., Randall, S.: A privacy attack on multiple dynamic match-key based privacy-preserving record linkage. International Journal of Population Data Science **5**(1) (2020)
33. Vidanage, A., Ranbaduge, T., Christen, P., Schnell, R.: Efficient pattern mining based cryptanalysis for privacy-preserving record linkage. In: 2019 35th International Conference on Data Engineering. pp. 1698–1701 (2019)