公募シンポジウム | 第40回医療情報学連合大会（第21回日本医療情報学会学術大会） | 公募シンポジウム
公募シンポジウム2
安全安心なデータ流通プラットホームとセキュリティ基盤技術
2020年11月19日(木) 14:00 〜 16:00 F会場 (イベントホール・特設会場1)

[2-F-2-03] ブルームフィルタを利用した階層データの近似マッチング

*三本 知明[1]、清本 晋作[1]、宮地 充子[2] （1. 株式会社KDDI研究所, 2. 大阪大学）
*Tomoaki Mimoto[1], Shinsaku Kiyomoto[1], Atsuko Miyaji[2] （1. 株式会社KDDI研究所, 2. 大阪大学）
キーワード：Privacy-preserving record linkage, Bloom filter, Algorithm

ビッグデータの利活用に向けた取り組みが進む中、プライバシ保護の動きも活発になっている。プライバシ保護技術として、これまで匿名化や秘密計算によるデータ分析などが提案されてきており、本研究ではその中でもブルームフィルタを用いたデータマッチングに着目する。ブルームフィルタを用いることで非常に高速な処理が可能となるため、大規模なデータであってもより安全にデータが利活用できるようになる。本研究ではブルームフィルタ上で階層データを近似マッチングするためのデータコード化アルゴリズムを提案する。

©一般社団法人 日本医療情報学会

ブルームフィルタ上の近似マッチングに向けたコード化アルゴリズム

三本知明[*1,2]、清本晋作[*1]、
宮地充子[*2]
*1 KDDI 総合研究所、*2 国際電気通信基礎技術研究所
*3 大阪大学

A Coding Algorithm for Approximate Record Linkage on Bloom Filter

Tomoaki Mimoto[*1,2] , Shinsaku Kiyomoto[*1] , Atsuko Miyaji[*3]
*1 KDDI Research, Inc.,
*2 Advanced Telecommunications Research Institute International,
*3 Osaka University

Privacy-preserving record linkage (PPRL) is one of privacy-preserving technique and it enables to link real-world data held by different parties without revealing any sensitive information about these data. There are some techniques to achieve the aim such as secure multi-party computation (SMC), but it is still difficult to realize the aim for huge data in realistic time. PPRL using bloom filter can be processed at high speed and this technique has attracted attention in recent years. In this paper, we propose a coding algorithm for privacy-preserving and approximate record linkage algorithm based on bloom filter. Our proposal enables to change words to bit strings that maintain the distance semantically and enables to apply to a multi-party PPRL algorithm. We propose a greedy algorithm but the results of experiments show the algorithm is effective. Furthermore, the problem we are going to solve comes down to an optimization problem and we formalize the problem.

Keywords: Privacy-preserving record linkage, Bloom filter, Algorithm

1. はじめに

近年様々な分野において、個人データの利活用に向けた取り組みが進められている。それと同時にプライバシ保護にも注目が集まっており、法改正やプライバシ保護に向けたシステム設計などが行われている。ビッグデータを利活用するには企業間のデータ共有は不可欠であるが、その多くは個人データを含むため、データの匿名化や個人のデータ利用に際して再同意の取得が必要となり、大きな課題となっている。医療分野においても各医療機関が保有する患者の医療データの共有および解析が期待されており、それに向けて[1]など様々な活動が行われている。データ利活用に向けた大きな方向性として、データを暗号化した状態で秘密計算等を用いて解析を行う方法や、データを個人が識別できない状態に匿名化した後に、データを共有する方法、またそれらを組み合わせた方法が検討されており、本稿では匿名化したデータのうち必要な情報だけを安全に共有する Privacy-preserving record linkage (PPRL)[2]の一種を提案する。

本稿では従来研究[3]のマルチパーティによるブルームフィルタを用いた近似マッチングアルゴリズムにカテゴリデータを適用可能にするコード化アルゴリズムを提案する。データベースのマッチングを行う際、実際のデータは必ずしも正しく整形されているわけではないため、完全なマッチングではなく、近似マッチングが求められている。[3]ではブルームフィルタを用いてその課題を解決しているが、意味上で距離が近いデータの近似マッチングを行うことができないという課題があった。そこで、我々は近い意味を持つデータを[3]のアルゴリズムに適用可能にするためのデータ整形を行うアルゴリズムを提案する。

具体的な利用シーンとして、複数の病院が患者情報を共有するケースを想定する。ある病院は患者の年齢、性別、生活習慣などと特定の疾病との関係を調べたいとする。しかし、その病院が保有する患者データは限られているため、他の医療機関の患者データを利用したいという状況が考えられる。各医療機関は情報漏えいの対策の一環として一定レベルの匿名化を行ってデータを保管しているが、我々が提案する手法を用いることで各医療機関が保有するデータをそれぞれのブルームフィルタに格納し、秘密裏に近い意味を持つデータの近似マッチングを行い、例えば目的とする疾病に近い疾病を持つ患者データのみを安全に共有することが可能となる。安全なデータ共有を行うことで、単純にデータを共有する場合と比較して患者からのデータ利用に関する同意取得が取りやすくなるというメリットもある。近似マッチングは他にも歯列情報などにも適用できる。2011 年の東日本大震災など、大規模な自然災害などが発生した場合、歯列情報をもとにした身元確認を行うことがあるが、医療機関が持つ歯列情報と実際の被災者の歯列情報は完全に一致しているわけではなく、近似マッチングによる個人推定が求められる。このようにデータを安全に近似マッチングさせる技術により、データの共有を柔軟かつ円滑に推進することが可能である。

2. 関連研究

2.1 ブルームフィルタ

個人データを保護しつつ、データをリンクさせる手法の一つとして、ブルームフィルタを用いる手法がある。ブルームフィルタは長さmのビット配列とk個のハッシュ関数から構成される。各ハッシュ関数は入力に対し、一様にランダムなビット位置を返す。[3]では文字列の近似マッチングを提案している。まずブルームフィルタの各配列には 0 を入力する。次に入力文字列Sに対してn-gram を用いて$Q_S = \{q_1, \dots, q_z\}$の文字列に分割する。各文字列はハッシュ関数を用いて、その返り値

に基づいて 1 をセットする。その後各文字列の出力結果に対して or 演算を行ったものをブルームフィルタにセットする。

より正確には、文字Σから構成される文字列$S \in \Sigma^*$を定義したとき、$f: \Sigma^n \mapsto \{0,1\}^m$を$k$個のハッシュ関数を含む$n$-gram エンコーディング関数、$g: \Sigma^* \mapsto \{0,1\}^m$を以下を満たすエンコーディング関数と表すことができる。

$$g(S) = \bigvee_{i=1}^{z} f(q_i)$$

例えば$S = jcmi, n = 2, m = 10, k = 2$の場合、$Q_{jcmi} = \{_j, jc, cm, mi, i_\}$となる。次に$f(_j) = 0001001000, f(jc) = 1001000000, f(cm) = 0000001100, f(mi) = 0100000100, f(i_) = 1000000001$を得たとすると、$g(jcmi) = 1101001101$となり、エンコーディングが完了する。

2.2 ブルームフィルタを用いた近似マッチング

[3]や[4],[5]などでブルームフィルタを用いたマッチングアルゴリズムが提案されている。これらは完全に一致するかどうかを判定するアルゴリズムであるが、[3]ではマルチパーティによる文字列の近似マッチングを可能としている。文字列の近似には Dice 係数を用いる。Dice 係数とはq個の集合$s_1, \dots, s_q$に対して、以下のように求められる。

$$Dice(s_1, \dots, s_q) = \frac{q \cdot |s_1 \cap \dots \cap s_q|}{|s_1| + \dots + |s_q|}$$

これは集合の平均要素数に対する共通の要素数の割合を表しており、集合が完全に一致すれば1、一致しなければ0というように$0 \le Dice(\cdot) \le 1$で集合の類似度を表す。

[3]では複数のブルームフィルタ上のデータを分割し、マルチパーティで Dice 係数を評価することで、ブルームフィルタ上で表すデータの類似度を評価する手法を提案している。マルチパーティで必要な処理は単純な or 演算と and 演算のみであるため、秘密計算を用いて評価を行うなどすれば各ブルームフィルタ上のデータの保護も可能である。

2.3 攻撃手法

ブルームフィルタを用いた PPRL では、高速に処理が可能であるという利点がある一方で、ハッシュ関数を用いることでデータが一意に決定するため、頻度分析攻撃などいくつかの攻撃が考えられる。[6]などでは文字列を n-gram で分割してハッシュをかけるため、大量のデータを集めることで、出現しやすい文字列とブルームフィルタ上のデータを推測する攻撃を提案している。他にも[7],[8]など数多くの攻撃が提案されている。

しかしその一方で、攻撃対策の検討も進められており、例えば[9]では構成したブルームフィルタに 0 と 1 を反転させたブルームフィルタを連結することで、0 と 1 の割合が等しいブルームフィルタを再構成することでブルームフィルタの偏りをなくす対策や、[10]ではブルームフィルタを分割し、XOR など追加の処理を行うことでブルームフィルタそのものへの攻撃を防ぐ手法が提案されている。また、前述の通り、ブルームフィルタを用いる手法は非常に高速な処理が可能であるため、大規模なデータセットに対する PPRL に非常に向いており、スケーラビリティを考慮した PPRL の提案として[11],[12]などがある。

3. 提案手法

3.1 概要

[3]ではブルームフィルタ上で近似マッチングが可能であるが、3.の方法では文字列の近似しか行うことができない。しかし、現実には文字列が異なっていても意味上で近いデータが存在する。例えば「野球」と「サッカー」は同じ「球技」というカテゴリに分類されるため、意味上では近いデータであると考えることができるが、文字列としては全く異なるため既存手法では、意味上での近さを評価することができないという課題がある。そこで本稿ではカテゴリ化されたデータのブルームフィルタ上での近似マッチングを可能とする文字列のコード化アルゴリズムを提案する。

3.2 距離指標

文字の意味上の近さを定量評価するために、距離指標が必要となる。ここでは階層木構造を持つデータに対する距離を定義する。[13]などでは、このような階層木構造を持つデータの距離を階層を元に定義しているが、本稿では各データ間の距離が必要となる。

そこで、まず階層木構造の各ノード$n_i \in N$の情報量$I(n_i)$を定義する。n_iがリーフLに含まれるとき、情報量を$I(n_i) = |L|$、n_iがルートのとき、情報量を$I(n_i) = 0$と定義する。さらに、中間ノード$n_i = W$は該当ノード配下のリーフの集合をL_Wとすると、$I(n_i) = |L| - |L_W|$と定義する。このとき、ノードa, bの距離は、a, bの共通ノードで最も下位に位置するノードを$\Lambda(a, b)$とすると、以下で定義する。

$$Dis(a, b) = \max\big(I(a), I(b)\big) - I\big(\Lambda(a, b)\big)$$

例えば、図1のノード 1,10 の場合、$I(n_1) = 7, I(n_{10}) = 4$であり、$I\big(\Lambda(n_1, n_{10})\big) = I(n_{11}) = 0$より$Dis(n_1, n_{10}) = 7$となる。

3.3 コード化アルゴリズム

ここでは、具体的なコード化アルゴリズムを提案する。具体的には、各データを上記距離指標に応じたハミング距離（以下では距離とする）を持つようなビット列に変更する。したがって、最終的にはデータiは$\boldsymbol{x^i} \in \{0,1\}^*$で表されることとなる。意味上の距離が近いデータはコード化することでハミング距離が近くなり、n-gram で分割したデータセットをブルームフィルタに格納すると、その結果も近くなるため、近似マッチングを行うことが可能となる。まず、表記を整理する。提案アルゴリズムでは階層木構造を持つデータを対象とする。階層木の高さはリーフを高さ 1 としてHとする。また、高さ 2 のノードおよびその配下のリーフを含めたサブツリーをブロックb_iとし、b_iの親ノードをp_iとする。さらに、b_iに含まれるリーフの数を$n_i = 2k_i$あるいは$n_i = 2k_i - 1$とする。

以下ではコード化アルゴリズムの一例を示す。

1. まず各ブロックについて長さ$k_i \cdot (n_i + 1)$のビット列を$n_i + 1$個用意し、各ビット列の距離が$2k_i$となるように各ビットに 0 と 1 を入力する。ここで各ブロックのビット列はk_i個のビットセットから構成され、各ビットセットにはたかだか 1 個だけ 1 が存在するものとする。
2. 各ブロックのビット列を連結する。このとき連結するビット列はp_iに該当するビット列とする。また、b_iに含まれないノードについてはすべてのp_iに該当するビット列を連結する。
3. 高さが$2 \le h \le H$のノードに該当するビット列について、自分の下位ノードに含まれるリーフが表すビット列のうち、リーフのみ 1 を含むビットセットを一つ選び、すべて 1 に変換する。

4. 各ビット列にルートが持つ枝の数だけビット列（全て 0）を追加する。ルート以外の各ノードについて、ルートまで遡る際に通過する枝に対応するビット列を 1 に変換する。ルートについては追加した全てのビット列を 1 に変換する。
5. 全ノード間のハミング距離の総和$H'(\boldsymbol{x}^i, \boldsymbol{x}^j)$と実際の距離$H(\boldsymbol{x}^i, \boldsymbol{x}^j)$の差の総和$\sum_{i,j}\left|H(\boldsymbol{x}^i, \boldsymbol{x}^j) - H'(\boldsymbol{x}^i, \boldsymbol{x}^j)\right|$、この値が減少する限り 4.を繰り返す。
6. 最後に、各ビット列に対して同一の permutation 関数を用いてビット列の並び替えを行う。

1.ではリーフおよび高さ 2 のノードの距離を可能な限り適切な値になるように調節し、3.ではルートおよび高さ$H-1$のノードの距離を調節する。最後に 4.および 5.では中間ノードの距離を調整しており、最終的に全体の距離が実際の距離に近づくようにコード化できている。

Example. ここでは具体的なアルゴリズムの適用例を示す。図 1 のような構造を持つデータを想定する。

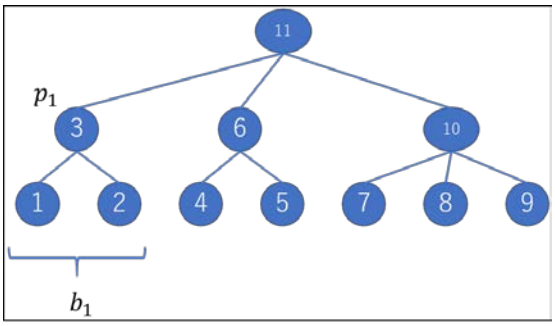

図 1:データの階層構造

1. ブロックb_iごとに、$2k_i$の距離を持つビット列を生成する。例えばb_1の場合、001、010、100 を生成し、b_3の場合、00010001、00100010、01000100、10001000 を生成する。
2. 1.で生成したビット列を連結する。例えば 1 が表すノードは 001||100||10001000、10 が表すノードは 100||100||10001000 となる。また、この時点では 3、6、11 も 10 と同じビット列となる。
3. ビット列の変換を行う。例えば 10 が表すノードは配下のリーフである 7,8,9 のみ 1 を持つビットセットの一つを全て 1 に変換する。したがって、100||100||10001111 となる。また、11 が表すノードは 111||111||10001111 となる。
4. 各ビット列に全てが 0 である 3 ビット連結する。その後、追加したビット列の一部を 1 に変換する。例えば 10 が表すノードは 100||100||10001111||001 となる。また、ルートは 111||111||10001111||111 となる。
5. $\sum_{i,j}\left|H(\boldsymbol{x}^i, \boldsymbol{x}^j) - H'(\boldsymbol{x}^i, \boldsymbol{x}^j)\right|$を計算し、結果が小さくなる限り 4.の処理を繰り返す。図 1 の例では、再度 4.の処理を行うと結果が増加するため、処理は一度で終了する。

3.4 最適化問題としてのコード化

本稿で提案したアルゴリズムは greedy な手法であるため、必ずしも最適なコード化とは言えない。3.までリーフ周辺およびルート周辺のノードの距離を調整しており、4.および 5.の処理では中間ノードの距離を修正している。しかし、各ノード間の距離は従属関係にあるため、4.および 5.の処理によって 3.までで調整した距離にズレが生じてしまう。そのためデータ構造が複雑な場合、目的とする距離と実際の距離とに大きな乖離が生じる可能性がある。したがって、より適切なコード化を行うために、将来的には解決する課題を最適化問題として検討する必要がある。

ノード数をnとすると、ノードiはmビットのベクトル$\boldsymbol{x}^i = \{0,1\}^m$で表すことができる。各ノード間は目的とする距離が定められており、ノードi,jの目的とする距離を$d_{i,j}$とする。このとき、各ノード間には以下の方程式が成り立つ。

$$x_1^1 \oplus x_1^2 + \cdots x_m^1 \oplus x_m^2 = d_{1,2}$$
$$\vdots$$
$$x_1^1 \oplus x_1^n + \cdots x_m^1 \oplus x_m^n = d_{1,n}$$
$$\vdots$$
$$x_1^{n-1} \oplus x_1^n + \cdots x_m^{n-1} \oplus x_m^n = d_{n-1,n}$$

また、実際の距離を$d'_{i,j}$とすると、$\sum_{i,j} d_{i,j} - d'_{i,j}$を最小化する問題を解くようなアルゴリズムを検討すればよい。

4. 評価実験

本章では、提案したアルゴリズムの有用性を評価する。本稿では図1の木構造（木構造1）および高さ3の完全二分木について評価を行う。まず、図2は木構造1の各ノードの距離を 4.2 章で定義した距離指標に基づいて計算した結果を表す。ノード間の距離が近いほどセルの色を濃く、距離が離れるにつれてセルの色をを薄くマッピングしている。

	1	2	3	4	5	6	7	8	9	10	11
1	0	2	2	7	7	7	7	7	7	7	7
2		0	2	7	7	7	7	7	7	7	7
3			0	7	7	5	7	7	7	5	5
4				0	2	2	7	7	7	7	7
5					0	2	7	7	7	7	7
6						0	4	4	4	5	5
7							0	3	3	3	7
8								0	3	3	7
9									0	3	7
10										0	4
11											0

図2:木構造 1 における実際の距離

次に図3，4，5は実際に提案アルゴリズムを各ノードに適用し、近似マッチングした結果を表す。ブルームフィルタの配列長は$m = 2^{16} = 65536$と固定し、n-gram の$n = 5,7,10$として近似マッチングを実施した。実験結果からはコード化したデータは 0,1 のみで表現されるため、nが小さいと距離が遠いビット列であってもブルームフィルタに格納される値は近くなり、ブルームフィルタ上では近いデータと見なされることが分かる。しかし、nを大きくすることで実際の距離関係をブルームフィルタ上で再現できている。特に図5は$n = 10$とした場合の結果であり、図2の実際の距離とブルームフィルタ上の距離がおおよそ一致していることが分かる。このように、実際の利用時には各パラメータを適切に与える必要がある。

	1	2	3	4	5	6	7	8	9	10	11
1	1	0.81481	0.6875	0.48485	0.625	0.45714	0.51613	0.46667	0.48276	0.4	0.17647
2		1	0.68966	0.46667	0.48276	0.4375	0.42857	0.44444	0.46154	0.375	0.12903
3			1	0.45714	0.52941	0.54054	0.42424	0.4375	0.45161	0.54054	0.44444
4				1	0.85714	0.78947	0.64706	0.66667	0.625	0.52632	0.21622
5					1	0.75676	0.72727	0.6875	0.64516	0.54054	0.22222
6						1	0.55556	0.57143	0.58824	0.6	0.41026
7							1	0.90323	0.86667	0.77778	0.28571
8								1	0.96552	0.74286	0.23529
9									1	0.76471	0.24242
10										1	0.41026
11											1

図3:木構造 1 におけるブルームフィルタ上の距離（$n = 5$）

	1	2	3	4	5	6	7	8	9	10	11
1	1	0.7027	0.61905	0.25	0.5	0.25532	0.34783	0.31818	0.34146	0.29787	0.08333
2		1	0.64865	0.27907	0.32558	0.28571	0.29268	0.30769	0.33333	0.28571	0.09302
3			1	0.29167	0.375	0.29787	0.30435	0.31818	0.34146	0.29787	0.33333
4				1	0.74074	0.71698	0.61538	0.56	0.51064	0.45283	0.11111
5					1	0.67925	0.65385	0.68	0.59574	0.5283	0.11111
6						1	0.47059	0.4898	0.52174	0.46154	0.22642
7							1	0.79167	0.75556	0.70588	0.15385
8								1	0.83721	0.69388	0.16
9									1	0.73913	0.17021
10										1	0.16352
11											1

図4：木構造 1 におけるブルームフィルタ上の距離（$n = 7$）

	1	2	3	4	5	6	7	8	9	10	11
1	1	0.65306	0.59259	0.10345	0.31034	0.10345	0.10345	0.17241	0.14286	0.17241	0.06897
2		1	0.61224	0.11321	0.15094	0.11321	0.11321	0.18868	0.15686	0.18868	0.07547
3			1	0.13793	0.2069	0.13793	0.13793	0.2069	0.17857	0.2069	0.27586
4				1	0.64516	0.64516	0.29032	0.22581	0.33333	0.22581	0.09677
5					1	0.6129	0.22581	0.32258	0.36667	0.25806	0.09677
6						1	0.25806	0.25806	0.36667	0.25806	0.09677
7							1	0.6129	0.56667	0.54839	0.12903
8								1	0.63333	0.5914	0.12903
9									1	0.6	0.13333
10										1	0.12903
11											1

図5：木構造 1 におけるブルームフィルタ上の距離（$n = 10$）

また、図6は高さ$H = 3$の完全二分木の各ノード間の実際の距離を表し、図7は。$n = 10$としたときのブルームフィルタ上の各ノードの距離を表す。この結果からも$n = 10$程度で高い精度で一致していることが分かる。

0	2	2	4	4	4	8	8	8	8	8	8	4	8	8
	0	2	4	4	4	8	8	8	8	8	8	4	8	8
		0	4	4	2	8	8	6	8	8	6	2	8	6
			0	2	2	8	8	8	8	8	8	4	8	8
				0	2	8	8	8	8	8	8	4	8	8
					0	8	8	6	8	8	6	2	8	6
						0	2	2	4	4	4	8	4	8
							0	2	4	4	4	8	4	8
								0	4	4	2	6	2	6
									0	2	2	8	4	8
										0	2	8	4	8
											0	6	2	6
												0	2	4
													0	4
														0

図6：二分木構造における実際の距離

1	0.56	0.56	0.493333	0.44	0.4	0	0	0	0	0	0.04	0.4	0	0
	1	0.52	0.4	0.52	0.4	0	0	0	0	0	0.04	0.4	0	0
		1	0.44	0.48	0.52	0.04	0.04	0.04	0.04	0.04	0.08	0.64	0.04	0.181818
			1	0.56	0.56	0.12	0.12	0.12	0.12	0.12	0.12	0.48	0.12	0.045455
					0.52	0.12	0.12	0.12	0.12	0.12	0.12	0.44	0.12	0.045455
					1	0.16	0.16	0.16	0.16	0.16	0.16	0.56	0.16	0.045455
						1	0.56	0.56	0.4	0.44	0.4	0.04	0.4	0.090909
							1	0.52	0.4	0.44	0.4	0.04	0.4	0.090909
								1	0.44	0.48	0.44	0.04	0.56	0.090909
									1	0.56	0.56	0.04	0.48	0.090909
										1	0.52	0.04	0.44	0.090909
											1	0.04	0.56	0.090909
												1	0.04	0.318182
													1	0.090909
														1

図7：二分木構造におけるコード化データの距離($n = 10$)

5. まとめ

本稿ではブルームフィルタ上で階層構造を持つデータの近似マッチングが可能となるデータのコード化アルゴリズムを提案した。実験結果からは実際の各データ間の距離とブルームフィルタに格納されたデータ間の距離の関係を維持できており、従来研究の近似マッチングアルゴリズムを用いることで、安全に階層構造を持つデータの近似マッチングを行うことが可能となる。しかし、提案手法は最適な手法とは言えず、階層構造以外の構造を持つデータに対して、また複雑な階層構造を持つデータに対しては正確に行うことができない可能性がある。したがって、今後はデータのコード化を最適化問題として扱い、より適切なコード化アルゴリズムの提案を目指す。

参考文献

1)https://www.jst.go.jp/kisoken/crest/project/45/14532150.html

2) R. Hall, S. Fienberg, Privacy-preserving record linkage, in PSD (2010), pp. 269-283.

3) S. M. Randall, A.M. Ferrante, J.H. Boyd, J.B. Semmens, Privacy-preserving record linkageon large real world datasets, inElsevier JBI(2014) volume 50, pp. 205-212.

4) D. Karapiperis, A. Gkoulalas-Divanis, and V. S. Verykios, “FEDERAL: A framework for distance-aware privacy-preserving record linkage,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 2, pp. 292--304, Feb. 2017.

5) D. Vatsalan, Z. Sehili, P. Christen, and E. Rahm, “Privacy-preserving record linkage for Big Data,” in Handbook of Big Data Technologies. Berlin, Germany: Springer, 2017, pp. 851--895.

6) Kuzu, M., Kantarcioglu, M., Durham, E. and Malin, B. (2011). A constraint satisfaction cryptanalysis of bloom filters in private record linkage. In Privacy Enhancing Technologies. Springer, 226-245.

7) M. Kroll and S. Steinmetzer, “Who is 1011011111...1110110010? Automated cryptanalysis of Bloom filter encryptions of databases with several personal identifiers,” in Proc. Int. Joint Conf. Biomed. Eng. Syst. Technol., 2015, pp. 341--356.

8) F. Niedermeyer, S. Steinmetzer, M. Kroll, and R. Schnell, “Cryptanalysis of basic Bloom filters used for privacy preserving record linkage,” J. Privacy Confidentiality, vol. 6, no. 2, pp. 59?79, 2014.

9) R. Schnell and C. Borgs, “Randomized response and balanced Bloom filters for privacy preserving record linkage,” in Proc. IEEE 16th Int. Conf. Data Mining Workshops, 2016, pp. 218--224.

10) R. Schnell and C. Borgs, “XOR-folding for Bloom filter-based encryptions for privacy-preserving record linkage,” German Record Linkage Center, Nuremberg, Germany, Rep. no. WPGRLC-2016-03, 2016.

11) T. Ranbaduge, D. Vatsalan, and P. Christen, “Hashing-based distributed multi-party blocking for privacy-preserving record linkage,” in PAKDD, Springer LNAI, Auckland, 2016.

12) T. Ranbaduge, P. Christen, and D. Vatsalan, “Clustering-based scalable indexing for multi-party privacy-preserving record linkage,” in PAKDD, 2015.

13) J. -W. Byun, A. Kamra, E. Bertino, and N. Li, ″Efficient k-anonymization using clustering techniques.″ in Proc. of the 12th Int. Conf Database Systems for Advanced Applications, (DASFAA’07).