3I2-OS-20-6

プライバシ保護データマイニングのための分散匿名化プロトコルの提案

Proposal of Distributed Anonymization Protocol for Privacy Preserving Data Mining

竹之内 隆夫[*1*2] 川村 隆浩[*2] 大須賀 昭彦[*2]
Takao TAKENOUCHI Takahiro KAWAMURA Akihiko OHSUGA

[*1]日本電気株式会社 情報・ナレッジ研究所
Knowledge Discovery Research Laboratories, NEC Corporation.

[*2]電気通信大学 大学院情報システム学研究科
Graduate School of Information Systems, The University of Electro-Communications.

Recently, service providers collect user's personal information for their business. It is expected that personal information stored by different service providers are federated and combined to make a new service. However, there is a risk that a specific user record can be identified by the combined personal information, and the user's sensitive information is revealed. Also, it is not allowed to disclose all personal information collected by the service provider to other service providers because of the security issue. Thus, several researches have investigated distributed anonymization protocol, which combines the personal information stored by multiple data holders and sanitize them to ensure anonymity policy with the minimum disclosure. However, when a set of the users is not unique between the service providers, there is a problem of revealing a presence of each service provider. This paper introduces a new distributed anonymization protocol which hides the presence of individual.

1. はじめに

近年，いくつかのサービス事業者は，ユーザの年齢等のパーソナル情報を収集して自社のビジネスに利用している．今後これらのパーソナル情報は単一の事業者内での利用にとどまらず，様々な事業者のパーソナル情報と組み合わせて利用され，新たなサービスが創出される事が期待される [佐久間 11].

しかしパーソナル情報を組み合わせると，その組み合わせからのユーザの特定が可能になり，他人に知られたくない情報が特定のユーザに紐付いてしまう恐れがある．またサービス事業者が収集したパーソナル情報は，サービス事業者の情報資産でもあるため，他事業者へ全て開示することは好ましくない．

そこで事業者が持つ情報を，必要最小限の開示に留めながら結合し，新たな情報を生成する分散匿名化プロトコルが注目されている [Mohammed 09, Jurczyk 09]．しかし既存の分散匿名化プロトコルでは，事業者間でユーザ集合が一致しない場合に，ユーザの情報が自事業者に「存在する/しない」というユーザ存在が，他事業者に漏洩してしまう問題がある．例えば，ローン会社にユーザの情報が存在することを知られると，そのユーザは借金をしていると推測される恐れがある．そのため，ユーザ存在はユーザのプライバシに関わる情報といえる．

そこで，著者らはユーザ存在を隠蔽した分散匿名化プロトコルについて研究しており，ユーザ存在の隠蔽についての新たな指標として *δ-max-site-presence* を提案した [竹之内 11]．本論文では，この指標を満たす，新たな分散匿名化プロトコルを提案する．本論文の構成は以下のとおりである．まず，2 章で分散匿名化プロトコルにおけるユーザ存在の隠蔽の課題と，*δ-max-site-presence* について説明する．次に，3 章でユーザ存在の隠蔽の課題を解決する新たな分散匿名化プロトコルを提案する．続いて 4 章で，提案したプロトコルの有効性を評価し，最後に 5 章で本論文の内容をまとめる．

連絡先: 竹之内 隆夫，日本電気株式会社，〒 211-8666 神奈川県川崎市中原区下沼部 1753，takenouchi@bu.jp.nec.com

2. 分散匿名化における課題

2.1 匿名化

匿名化とは，ユーザを特定できないようにパーソナル情報を加工することである．ここでパーソナル情報とは「属性」と「属性値」として表現されるユーザに関する情報であり，あるユーザのパーソナル情報をテーブルのレコードとして表現する．そして，単一の属性ではユーザを特定できないが，複数組み合わせるとユーザを特定できる可能性のある属性の組合せを準識別子 (quasi-identifier,QID) と呼ぶ．また，ユーザを特定された状態で開示されることが望ましくない属性をセンシティブ属性 (sensitive attribute,SA) と呼ぶ．この時，もし攻撃者があるユーザの QID の属性値を知っていたとすると，そのユーザのレコードを特定できてしまい，SA の属性値を知られてしまう．これを防ぐために，QID の属性値を一般化 (generalize) して，より抽象的な値にする方法が知られている．そして，QID の属性値によって識別されるレコードが少なくとも k 個以上ある場合，そのテーブルは k-匿名性を満たすという [Sweeney 02].

2.2 分散環境における匿名化

複数の事業者が保持するテーブルを結合し，結合したテーブルを匿名化する処理を分散匿名化と呼ぶ [Mohammed 09, Jurczyk 09]．本論文では，事業者 A,B が以下のようなテーブル (T_A,T_B) を持ち，匿名化済みのテーブル (T^*) を生成する．

$$T_A(ID, QID_A), T_B(ID, QID_B, SA), T^*(QID_A, QID_B, SA)$$

ここで，ID はユーザ識別子，QID_A, QID_B は事業者 A,B が持つ準識別子であり，異なる属性であるとする．

分散匿名化では，必要最小限の開示に留めながら自事業者のテーブルを結合し匿名化を行う．これは，異なる事業者間で完全な信頼関係を築くのは困難であり，テーブルを全て開示するのは危険であると考えられているためである．但し，本論文では事業者 A,B はある程度の信頼がおける企業であることを前提としているため，各事業者は semi-honest であるとする．

2.3 ユーザ存在情報の漏洩課題と指標

既存の分散匿名化プロトコルでは，事業者間でユーザ集合が一致している前提があった．しかし，今後は様々な事業者間でのパーソナル情報の利用が期待されるため，ユーザ集合が一致しない場合の対応が必要がある．つまり，一部のユーザが片方の事業者にだけ存在する場合にも対応する必要がある．

このように，ユーザ集合が一致しない場合は，事業者 A のテーブル T_A と事業者 B のテーブル T_B を結合し匿名化したテーブル T^* は，事業者 A と B の共通ユーザのレコードだけとなる．すると，自事業者のテーブルと結合後の匿名テーブルの比較によってユーザ存在を推測できてしまう問題が発生する．例えば，事業者 A の T_A に user1〜6 の 6 レコードが存在し，年齢情報として {11,12,13,14,15,16} 才という情報が保持されているとする．また，T^* のレコード数が 4 個であったとする (つまり，user1〜6 の 6 名のうち 4 名は，事業者 B にも存在する)．この時，もし T^* に年齢「11〜12 才」レコードが 2 個，年齢「13〜16 才」レコードが 2 個あるように汎化されていたとすると，事業者 A は，user1,2 の 2 名が「11〜12 才」である事と，T^* には事業者 A と B の共通ユーザのみのレコードである事から，user1,2 は確実に事業者 B にも存在することがわかってしまう．それに対し，T^* が「11〜13 才」レコードが 2 個,「14〜16 才」レコードが 2 個のように汎化されていたとすれば，事業者 A は，user1,2,3 の 3 名のうち，いづれか 2 名が事業者 B に存在することまでしか推測できない．

そこで，T_A や T_B と T^* の比較によりユーザ存在を知られることを防ぐために，既存のユーザ存在の指標である *δ-presence* を分散匿名化用に拡張した *δ-max-site-presence* を提案している [竹之内 11]．この指標では，事業者 $n \in \{A, B\}$ の各属性値の組合せによるユーザ存在の推測の可能性が δ 以下である時，T^* は *δ-max-site-presence* を満たすと定義している．例えば，事業者 A,B から見たユーザ存在の推測の可能性が 2/3 以下であれば，T^* は 2/3-*max-site-presence* を満たすと表現する．

本論文では，これらの課題を解決するために以下の要件を満たしつつ，できるだけ詳細な T^* を出力するような分散匿名化プロトコルを提案する．

要件 1 T^* は *k*-匿名性と *δ-max-site-presence* を満たすこと

要件 2 通信内容から T^* よりも詳しい情報が極力漏れないこと

3. ユーザ存在を隠蔽した分散匿名化プロトコルの提案

提案するプロトコルは，既存の [Jurczyk 09] の分散匿名化プロトコルと同様に，匿名化アルゴリズムとして広く利用されている Mondrian[LeFevre 06] をベースにする．これは，QID の属性値を最も一般化されている状態 (「*」など) から徐々に詳細化 (specialize) する手法である．ここで詳細化とは，QID の属性値で識別されるユーザ集合を，ある境目で分割することである．この分割の境目となる属性値を分割点と呼ぶ．例えば，年齢を「20 才」という分割点で分割すると,「20 才以上」と「20 才未満」に分割することになる．分割後のユーザ集合のユーザ ID は，分割をした事業者 (分割側の事業者) から相手の事業者 (非分割側の事業者) に送信され，共有される．

しかし，ユーザ集合が異なる場合に単純に既存の分散匿名化プロトコルを適用してしまうと，分割後のユーザ ID を相手の事業者に通知する際に，自事業者に存在するユーザ ID だけを通知することになる．つまり，ユーザ ID の通知からユーザ存在を容易に推測されてしまう．そこで，自事業者に存在しないが，あたかも存在するかのように扱うダミーユーザを導入する．これにより，通知されるユーザ ID が存在するユーザなのかどうかの区別を困難にできる．また，ダミーユーザを用いた本プロトコルをダミーユーザプロトコルと呼ぶ．

function split(U_p:ダミー入りユーザ ID 集合)
1: U_p のダミーユーザのダミー値を更新
2: *point* ⇐ 分割点決定関数を用いて分割点を決定
3: *point* で分割した際に *k*-匿名性と *δ-max-site-presence* を満たすか確認
4: **if** 指標を満たせない **then**
5: U_p についての split 処理終了
6: **endif**
7: **if** *point* は自事業者の T_n^* の分割点 **then**
8: T_n^* を *point* で分割し，分割後のユーザ ID を相手の事業者へ送信
9: **else**
10: 相手から分割後ユーザ ID を受信し，T_n^* を分割
11: **endif**
12: U_{hi}, U_{low} ⇐ 分割後の集合．split(U_{hi}),split(U_{low}) を再帰呼出し

図 2: Step2(分割処理) のアルゴリズム

3.1 ダミーユーザプロトコル

ダミーユーザプロトコルは，まず事業者 A,B 間で通信し，各事業者内で内部匿名テーブル $T_n^*(n \in \{A, B\})$ を生成する．その後，結合したパーソナル情報の利用者である事業者 C が，事業者 A,B から T_n^* を取得・結合し，T^* を得る．以降で事業者 A,B 間でどのように T_n^* を生成するかを説明する (図 1(a))．

3.1.1 Step1:ダミーユーザの割当と T_n^* の初期化

最初に事業者 A，B は自事業者のダミーユーザを割り当てる．本プロトコルでは，双方の事業者のユーザを包含する母集団ユーザ集合 U を事前に知っているという前提を置く．ここで U は，事業者 A に存在するユーザ集合を U_A，事業者 B に存在するユーザ集合を U_B，事業者 A,B のどちらにも存在しないユーザ集合を U_O としたき $U = U_A \cup U_B \cup U_O$ ($U_O \neq \phi, U_A \cap U_B \neq \phi$) となる．このような前提は，例えば事業者 A,B が Open ID のような同一の認証サーバを利用している場合に成立し，認証サーバに存在する全ユーザが U となる．そして，事業者 A が持つダミーユーザは $U - U_A$，事業者 B が持つダミーユーザは $U - U_B$ となる．

次に内部匿名テーブル T_n^* を初期化し，最も一般化された状態にする．例えば，U={user1-15} であったとすると，T_A^*={user1-15,*}，T_B^*={user1-15,*} となる．

3.1.2 Step2:分割点の決定と分割処理

続いて，T_n^* を分割していく分割処理を行う (図 2)．まず，事業者 A,B は自事業者のダミーユーザの準識別子 (QID_A, QID_B) の属性値に適切な値を割り当てる．この値をダミー値と呼ぶ．ダミーユーザは，相手事業者からみて存在するユーザ (存在ユーザ) なのかダミーユーザなのか区別がつかないようにする必要があるため，分割対象のユーザにおける存在ユーザの準識別子の属性値の分布に沿ってダミー値を割り当てる．

次に，分割点決定関数を用いて分割点を決定する．この処理の詳細は 3.2 節で説明する．そして，決定した分割点で分割しても *k*-匿名性と *δ-max-site-presence* を満たせるかを確認し，指標を満たしている場合のみ T_A^*，T_B^* を分割する．例えば，事業者 A が持つ QID_A が年齢情報であり,「20 才」で分割を行う場合は，T_A^*={{user1-10,20 才未満 },{user11-15,20 才以上 }} のように 2 レコードに分割される．そして，事業者 A は事業者 B に分割後のユーザ ID を送信する．事業者 B は，受信した内容に従って T_B^* を分割する．この例では，T_B^*={{user1-10,*},{user11-15,*}} となる．

ここで，*k*-匿名性と *δ-max-site-presence* を満たしているかを確認するためには共通ユーザの人数を知る必要があるが，事

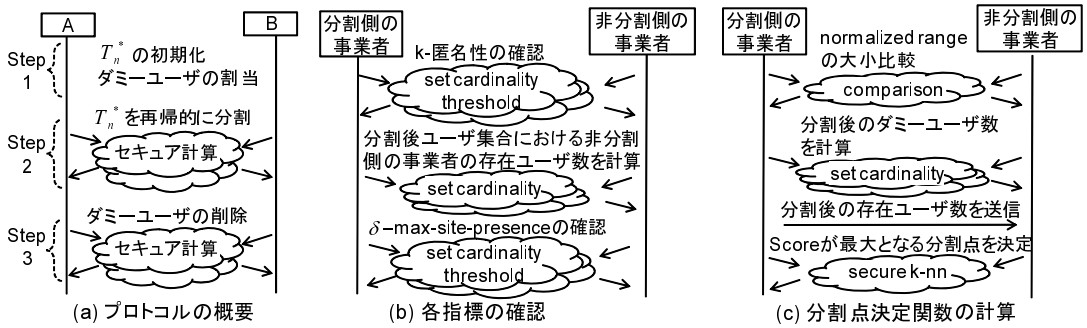

図 1: ダミーユーザプロトコルのシーケンス

業者 A,B 間でユーザ存在を隠す必要がある．そこで，*secure set intersection*[Freedman 04] というセキュア計算 [Lindell 09] のプロトコルを用いる．このプロトコルは，事業者 A,B が持つ集合をお互いに隠蔽しながら，それらの集合の積集合や，積集合の要素数 (*set cardinality*) や，積集合の要素数と指定した値との大小関係 (*cardinality threshold*) を求めることが出来る．事業者 A,B は，存在ユーザのユーザ ID の集合を入力として *secure set intersection* を実行し，積集合の人数が k 以上であるかを求めることで，k-匿名性を満たしているかを確認する (図 1(b))．同様に，積集合の人数が「分割後のユーザ集合における事業者 A,B の存在ユーザ数 $\times$ δ」以下であるかを求める事で，*δ-max-site-presence* を満たしているかを確認する．

3.1.3 Step3:ダミーユーザの削除

全ての分割処理が完了したらダミーユーザを削除し，共通ユーザ数を求める．これは，Secure Set Intersection を用いて，T_n^* の各レコードの SA の各属性値 s について，事業者 A の存在ユーザのユーザ ID の集合，事業者 B で s を持つ存在ユーザのユーザ ID の集合との積集合の個数を求めればよい．

そして，最終的に生成される T_A^* と T_B^* は，例えば T_A^*={{user1-10,20 才未満 },{user11-15,20 才以上 }} と，T_B^*={{user1-10,500 万未満,「s1:2 名，s2:3 名」},{user11-15,500 万以上,「s1:1 名，s2:1 名」}} のようになる．

以上のような Step1〜3 までの分割プロトコルによって，事業者 A,B はお互いにユーザ存在を隠蔽しながら内部匿名化テーブル T_A^*，T_B^* を分割していく．そして，事業者 C が T_A^*，T_B^* を取得して結合することで T^* が生成される．

3.2 ダミーユーザを考慮した分割点決定関数

本節では，ダミーユーザプロトコルで分割点を決定するための分割点決定関数を説明する．従来の Mondrian の分割点決定関数は，各属性の正規化済みの値域 (normalized range) が最大となる属性を選択し，その属性の中央値 (median) を分割点にしている．この従来の分割点決定関数を拡張し，新たに *δ-max-site-presence* も満たしやすい分割点が選ばれるようにする．本手法は，ダミーユーザによってユーザを隠蔽する手法であるため，分割後のユーザ集合にダミーユーザが偏りなく入る分割点が選ばれると良いと考えられる．

そこで，ダミーユーザのエントロピー (Dummy Entropy,DE) を導入する．

$$DE(c,n) = -\sum_{U_i \in U_{hi}, U_{low}} \frac{|du(n,U_i)|}{|U_i|} log(\frac{|du(n,U_i)|}{|U_i|}) \quad (1)$$

ここで c は分割点候補の属性値であり，分割前のユーザ集合 U_p を上位 U_{hi} と下位 U_{low} へ分割することを意味する．また，$du(n,U_i)$ はユーザ集合 U_i のうち事業者 n のダミーユーザの集合である．エントロピーは，事象全体における各事象の発生確率の偏りが小さいほど大きな値になるため，ダミーユーザが分割後のグループ内に偏りなく入る時に DE は大きくなる．

この DE を利用して，ダミーユーザプロトコルの分割点決定の分割点決定関数を定義する．まず，従来の Mondrian と同様に normalized range が最大となる属性を選ぶ．そして，その属性における分割点の候補となる属性値 ($x_i \in X$) を分割点候補 c_i として，以下のように定義したスコア値 S を計算する．

$$S(c_i) = \alpha(\frac{-L(c_i)}{\max_{x_j \in X}(L(x_j))}) + (1-\alpha)\frac{1}{2}\sum_{n \in A,B}(\frac{DE(c_i,n)}{\max_{x_j \in X}(DE(x_j,n))}) \quad (2)$$

$$L(c_i) = \sum_{x_j \in X} |x_j - c_i| \quad (3)$$

ここで $\alpha(0 \leq \alpha \leq 1)$ は，DE の影響を調整するための重みである．また，L は c_i の属性の各属性値 x_i と c_i の距離の和を意味する．median とは L が最小となる点と言い換えることができるため，α=1 とした時は c_i が median の時に S が最大となり，従来の Mondrian と同様に median が分割点に決定される．スコア値 S は，L と事業者 A,B についての DE を正規化して，重み付で足した値となる．S を最大化させる分割点で分割を行うことで，分割後のユーザ集合にダミーユーザが偏りがなく入り，結果的に *δ-max-site-presence* を満たしつつ多くの分割が可能になることが期待される．

3.2.1 分割点決定関数における MPC

提案する分割点決定関数は，属性値やユーザ存在を隠蔽したまま計算する必要があるため，3 つのセキュア計算のプロトコルを用いる (図 1(c))．まず，分割点の属性を選ぶ処理で *secure comparison*[Yao 82] を用いる．これは，事業者 A,B が持つ値を秘密にしながら大小関係を求めるプロトコルである．*secure comparison* を用いて，事業者 A,B がローカルで計算した最大の normalized range を比較し，どちらが大きいかを求め，分割を行う事業者 (分割側の事業者) を決定する．

次に，事業者 A,B で分割点候補 c_i の DE を計算する．しかし，非分割側の事業者では分割後のユーザ集合を知らないので DE をローカルで計算できない．そこで，*secure set intersection* を用いて，非分割側の事業者 n のダミーユーザ数 ($|du(n,U_i)|$) を求める．これは，分割後のユーザ集合 U_i と，非分割側の事業者 n のダミーユーザの積集合の要素数で計算できる．この時，分割後のユーザ数 ($|U_i|$) も一緒に出力する．但し，この

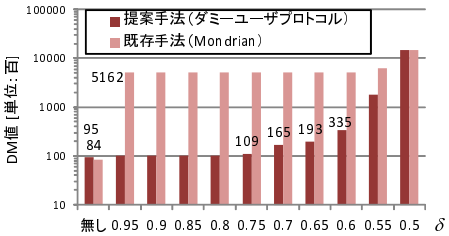

図 3: δ-*max-site-presence* に対する既存手法との比較

MPC については非分割側の事業者 n にだけ出力する．つまり，例えば c_i が事業者 A での分割であった場合，事業者 B は c_i で分割後のダミーユーザ数とユーザ数を知れるが，c_i の分割点の属性値や分割後のユーザ集合を知ることはない．以上により，DE の計算に必要な情報を得ることができたため，DE を事業者内でローカルに計算できる．

最後に，事業者 A,B は *secure k-nearest neighbor*[Zhan 05] というセキュア計算のプロトコルを用いて，分割点を決定する．この処理は，事業者 A,B からの入力として各事業者のローカルで計算した DE と L をそれぞれの最大値で割って正規化した値について，それらを足した $S(c_i)$ が最大となる分割点候補を出力する処理となる．以上のように，属性値やユーザ存在を相手事業者に秘密にしながら分割点を決定することができる．

4. 評価実験

提案プロトコルをプロトタイプ実装し，有効性を評価した．実装は Java 1.6 で行い，仮想的に事業者間で通信を行う構成で動作させた．評価データには，[Mohammed 09] の分散匿名化プロトコルの評価と同様に，UCI Repository の Adult データを 2 事業者に分割したデータを利用した．Adult データは，14 種類の属性と 1 種類の年収分類 (class)($50K 以上 or 未満) を持つ約 3 万レコードの米国の国勢調査を基にしたデータである．14 種類の属性を準識別子，年収分類をセンシティブ属性とした．そして，全レコードを約 3 万人の母集団ユーザ (U) としてとらえ，ランダムに並び変えた上位レコードから共通ユーザ ($U_A \cap U_B$)，事業者 A と事業者 B の片方だけに存在するユーザ ($(U_A - U_A \cap U_B)$, $(U_B - U_A \cap U_B)$)，残りを双方に存在しないユーザ (U_O) とした．実験では，共通ユーザを 1200 人，片方だけに存在するユーザ (つまり，片方だけに存在するダミーユーザ数) を 1200 人，$k = 2$，重み α を 0.5 として DE の影響を半分にして評価を行った．評価はデータ生成を含めて 10 回行い，評価値はその平均とした．なお，Mondrian と同様にカテゴリ値は数値として扱った．比較対象となる手法は，既存手法となる Mondrian を単純に分散環境に対応させた分散対応 Mondrian とした．評価指標は，δ-*presence*[Nergiz 07] と同じく Discernibility Metric(DM) を用いる．DM は匿名化による精度の低下の指標であり，小さいほど良い．

図 3 に，δ-*max-site-presence* の δ を変化させた際の評価結果を示す．この結果が示す通り，δ を指定せずにユーザ存在を隠蔽しない場合は既存手法の方が若干 DM 値が小さくなり，既存手法の方が良い結果となる．それに対し δ を 0.95 以下に指定してユーザ存在を隠蔽する場合は，既存手法は急激に DM 値が大きく (悪く) なるのに対し，提案手法は小さい (良い)DM 値を保っている．これは，ダミーユーザのエントロピー (DE) の追加や分割後のダミー値の更新により，ユーザ存在が隠蔽できるような分割点が選ばれるようになったためである．つまり提案手法により，ユーザ存在を隠蔽しながらも匿名化による情報損失を抑えられることが分かる．また，この評価結果では δ を 0.6 付近にすると DM 値は急激に悪くなる．これは，提案手法であっても適切な分割点を見つけられなかったためである．

以上の評価結果より，提案手法によってユーザ存在を隠蔽しながらも一定程度の情報の精度を保ったままの分散匿名化を実現できることが確認できた．

5. まとめと今後の課題

本論文では，ユーザ存在が推測される可能性を示した δ-*max-site-presence* という指標を満たすための，ユーザ存在を隠蔽した分散匿名化のプロトコルを提案した．評価の結果，特定の条件下であれば，ユーザ存在を隠蔽しながらも一定程度の情報の精度を保ったままの分散匿名化を実現できることが分かった．今後は，実際のデータを用いた評価や，計算量や通信量の評価を行っていく予定である．

参考文献

[Freedman 04] Freedman, M. J., Nissim, K., and Pinkas, B.: Efficient Private Matching and Set Intersection, in *Proc. EUROCRYPT'04*, pp. 1–19 (2004)

[Jurczyk 09] Jurczyk, P. and Xiong, L.: Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers, in *Proc. DBSec'09* (2009)

[LeFevre 06] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R.: Mondrian Multidimensional K-Anonymity, in *Proc. ICDE'06*, p. 25 (2006)

[Lindell 09] Lindell, Y. and Pinkas, B.: Secure Multiparty Computation for Privacy-Preserving Data Mining, *J. of Privacy and Confidentiality*, Vol. 1, pp. 59–98 (2009)

[Mohammed 09] Mohammed, N., Fung, B. C. M., Wang, K., and Hung, P. C. K.: Privacy-Preserving Data Mashup, in *Proc. EDBT'09*, pp. 228–239 (2009)

[Nergiz 07] Nergiz, M. E., Atzori, M., and Clifton, C.: Hiding the Presence of Individuals from Shared Databases, in *Proc. SIGMOD'07*, pp. 665–676 (2007)

[Sweeney 02] Sweeney, L.: k-anonymity: a model for protecting privacy, *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, pp. 557–570 (2002)

[Yao 82] Yao, A. C.: Protocols for Secure Computations, in *Proc. SFCS'82*, pp. 160–164 (1982)

[Zhan 05] Zhan, J. Z., Chang, L., and Matwin, S.: Privacy Preserving K-nearest Neighbor Classification, *Int. J. Network Security*, Vol. 1, No. 1, pp. 46–51 (2005)

[佐久間 11] 佐久間 淳, 高橋 克巳：クラウドストレージにおける個人情報の利活用とプライバシー保護, 情報処理, Vol. 52, No. 6, pp. 706–715 (2011)

[竹之内 11] 竹之内 隆夫, 伊東 直子, 川村 隆浩, 大須賀 昭彦：クラウド上での事業者間データ連携のための分散型パーソナル情報保護エージェント, in *Proc. JAWS2011* (2011)