情報処理学会論文誌 Vol.53 No.11 2432–2444 (Nov. 2012)

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

竹之内 隆夫[1,2,a)] 川村 隆浩[2] 大須賀 昭彦[2]

受付日 2012年1月27日, 採録日 2012年7月2日

概要：近年，いくつかのサービス事業者はクラウド上でサービスを提供し，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ことが期待される．しかし，パーソナル情報が組み合わさることによって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．またクラウド上のパーソナル情報は，サービス事業者のポリシによって保護されているため，他事業者へすべて開示することによる結合はできない．そこで，各事業者に配備されたエージェントが，パーソナル情報を必要最小限の開示にとどめながら結合し，ユーザが特定されない形に加工する分散匿名化プロトコルが注目されている．しかし既存のプロトコルでは，事業者間でユーザ集合が一致しない場合に，ユーザが自事業者に存在するか否かというユーザ存在が他事業者に漏洩してしまうという問題がある．そこで，本論文ではユーザ存在を隠蔽した分散匿名化プロトコルを行うパーソナル情報保護エージェントを提案する．また，提案するエージェントが実行するプロトコルを特定の条件下で評価し，ユーザ存在を隠蔽しながらも一定の情報の精度を保てることを示す．

キーワード：k-匿名化，プライバシ保護，分散匿名化

Distributed Personal Information Protection Agents for Cloud Data Federation

Takao Takenouchi[1,2,a)] Takahiro Kawamura[2] Akihiko Ohsuga[2]

Received: January 27, 2012, Accepted: July 2, 2012

Abstract: Recently, service providers are starting applications on the cloud platform, which collect vast amount of 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 on the cloud platform because of the security issue. Thus, several researches have investigated distributed anonymization protocol, which is an agent protocol to combine 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 agent protocol of distributed anonymization which hides the presence of individual in each database. Our evaluation results show that the proposed protocol can anonymize them according to the policy of hiding the presence and the anonymity without too much information loss.

Keywords: k-anonymity, privacy, distributed anonymization

[1] 日本電気株式会社情報・ナレッジ研究所
Knowledge Discovery Research Laboratories, NEC Corporation, Kawasaki, Kanagawa 211–8666, Japan
[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, Chofu, Tokyo 182–8585, Japan
[a)] takenouchi@bu.jp.nec.com

1. はじめに

近年，いくつかのサービス事業者はクラウド上でサービスを提供し，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ことが期待される [1], [2]．たとえば，オンデマンドビデオ配信サイト（事業者 A）とローン会社（事業者 B）が連携し，事業者 A が持つユーザの視聴番組および視聴時間帯の情報と，事業者 B が持つ年収情報を結合し，広告代理店（事業者 C）が番組視聴者の傾向分析を行うとする．すると「昼間に視聴するユーザ群」,「夜間に視聴する比較的高収入のユーザ群」および「夜間に視聴する比較的低収入のユーザ群」を見つけられるかもしれない．しかし，もし視聴情報と年収情報を結合しないとしたら，「昼間に視聴するユーザ群」および「夜間に視聴するユーザ群」しか見つけられないだろう．

しかしパーソナル情報を組み合わせると，その組合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．またクラウド上のパーソナル情報は，サービス事業者のポリシによって保護されているため，他事業者へすべて開示することによる結合はできない．

そこで事業者が持つ情報を，必要最小限の開示にとどめながら結合し，新たな情報を生成するマルチエージェント技術が研究されている．特に各事業者に配置されたエージェントが，パーソナル情報を結合し，ユーザが特定されない形式に変換・提供するためのエージェント間のプロトコルとして分散匿名化プロトコルが注目されている [2], [3]．ここでのエージェントとは，他のエージェントと協調し，サービス事業者に代わってパーソナル情報を指定されたポリシに従って加工・結合した結合匿名テーブルを生成するエージェントである（**図 1**）．

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という問題がある．たとえば，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，本論文ではユーザ存在を隠蔽した分散匿名化プロトコルを実行するパーソナル情報保護エージェントを提案する．本論文の構成は以下のとおりである．まず，2 章で既存の分散匿名化プロトコルと，ユーザ存在の隠蔽の課題について説明する．次に，3 章でユーザ存在を推測される可能性を示す指標を提案する．そして，4 章でユーザ存在の隠蔽の課題を解決するエージェントの新たな分散匿名化プロトコルを提案する．続いて 5 章で，提案したエージェントのプロトコルの有効性を評価し，6 章で安全性を評価する．そして 7 章で関連研究を示し，最後に 8 章で本論文の内容をまとめる．

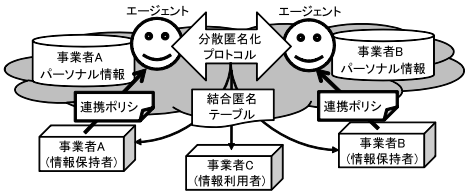

図 1 クラウド上のデータ連携のためのエージェントと分散匿名化プロトコル

Fig. 1 Agents and distributed anonymazation protocol for data federation.

表 1 k-匿名化の実行例

Table 1 k-anonymity.

(a) 匿名化前

郵便	年齢	病状
13053	29	かぜ
14821	36	かぜ
13001	21	HIV
14011	30	ガン

(b) 匿名化後

郵便	年齢	病状
130**	21–29	かぜ
130**	21–29	HIV
14***	30–36	かぜ
14***	30–36	ガン

2. 分散匿名化プロトコルにおける課題

2.1 匿名化

匿名化とは，ユーザが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る．**表 1** (a) では，テーブルのレコードがユーザに，カラムが「属性」に，フィールドの値がユーザの属性の「属性値」にそれぞれ対応する．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）と呼ぶ．また，ユーザが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）と呼ぶ．表 1 (a) の例では，郵便番号と年齢という属性の組合せが準識別子であり，病状という属性がセンシティブ属性と見なすことができる．たとえば，ある病院が表 1 (a) のような全患者の病状を記録したテーブルを保持していたとする．このテーブルには，氏名など直接ユーザを識別できるような属性は含まれていない．しかし，もしユーザ X が事前に「Alice はその病院に通院しており，郵便番号が 14011 で，年齢が 30 である」ことを知っていると，表 1 (a) の 4 番目のレコードが Alice であると知れてしまう．

そこで，準識別子の属性値によってユーザが特定されることを防ぐために，準識別子の属性値を一般化（generalize）して，より抽象的な値にする．このような加工により，準識別子の属性値によって識別されるレコードが少なくとも k 個以上あるテーブルを，k-匿名性 [4] を満たすという．表 1 (b) は 2-匿名性を満たす．

表 3 結合匿名テーブルによるユーザ存在の漏洩の例
Table 3 Example of user presence disclosure problem.

(a) 事業者A(T_A)

userID	年収
user1	300万
user2	400万
user3	550万
user6	600万
user7	650万
user8	700万

(b) 事業者B(T_B)

userID	時刻	番組
user1	16:00	Xドラマ
user2	17:00	Yアニメ
user4	17:30	Xドラマ
user5	16:30	Yアニメ
user6	15:00	Xドラマ
user7	12:00	Yアニメ
user9	14:00	Yアニメ
user10	14:30	Xドラマ

(c)ユーザ存在が漏洩する結合匿名テーブル(T^*)

年収(万)	時刻	番組
500未満	16:00以降	Xドラマ
500未満	16:00以降	Yアニメ
500以上	15:59以前	Xドラマ
500以上	15:59以前	Yアニメ

(d)ユーザ存在が漏洩しにくい結合匿名テーブル(T^*)

年収(万)	時刻	番組
600未満	16:00以降	Xドラマ
600未満	16:00以降	Yアニメ
600以上	15:59以前	Xドラマ
600以上	15:59以前	Yアニメ

表 2 分散匿名化プロトコルの実行例
Table 2 Distributed anonymization protocol.

(a) 事業者A(T_A)

userID	年収
user1	450万
user2	300万
user3	650万
user4	550万

(b) 事業者B(T_B)

userID	時刻	番組
user1	16:15	Yドラマ
user2	17:30	Xアニメ
user3	14:45	Zドラマ
user4	12:00	Xアニメ

(c) 結合匿名テーブル(T)

年収(万)	時刻	番組
500未満	16:00−	Yドラマ
500未満	16:00−	Xアニメ
500以上	−15:59	Zドラマ
500以上	−15:59	Xアニメ

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て匿名化する処理を分散匿名化と呼び，分散匿名化を行うエージェント間のプロトコルとして分散匿名化プロトコルが研究されている [3], [5], [6], [7]．特に事業者 A，B が持つテーブル（T_A，T_B）が異なる属性を持つ場合は，T_A，T_B は共通のユーザ ID を属性に持つ前提となっており，このユーザ ID で T_A，T_B を結合し，結合匿名テーブル（T^*）を生成する（**表 2**）．

分散匿名化プロトコルでは，各事業者に配備されたエージェントが，必要最小限の開示にとど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，テーブルをすべ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．そこで，個別の情報を極力出さずに結合匿名テーブルを生成するために，Top Down アプローチと Multi-Party Computation（MPC）[8], [9] などを組み合わせたプロトコルが用いられる．

Top Down アプローチとは，準識別子の属性値を最も一般化されている状態から徐々に詳細化（specialize）する手法である．ここで詳細化とは，準識別子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．たとえば，年収を「500 万」という分割点で分割すると，「年収 500 万以上」と「年収 500 万未満」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，双方の事業者で共有される．そして k-匿名性が満たされている間，分割を続ける．最後に，分割した双方のテーブル（内部匿名テーブル）を結合して最終的な結合匿名テーブルを生成する．このように何度も詳細化を行うことで，匿名性を保ちながらもデータマイニングなどで有用なデータを生成できる．

Top Down アプローチで分割点を決定するために，分割点決定関数というヒューリスティック関数が用いられる．この関数の計算を行うために個別のテーブルの属性値が必要なため，MPC という暗号プロトコルが用いられる．MPC を用いることで，双方の事業者は自事業者が持つ属性値を相手事業者に秘密にしながら，理論的にはその値を入力とした任意の関数の計算結果を得ることが可能である．しかし，複雑な計算を行う場合には処理量が多くなるため，現実的には大小比較や合計や積集合の計算などの単純な計算に限られる．MPC を用いることで，属性値を相手事業者に隠蔽しながら分割点を決定することができる．

2.3 ユーザ存在情報の漏洩課題

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という前提があった [3], [5], [6], [7]．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にも対応する必要がある．つまり，一部のユーザが片方の事業者のテーブルにだけ存在する場合にも対応する必要がある．しかし，ユーザの集合が一致しない場合に既存のプロトコルを適用すると，「結合匿名テーブルの問題」と「ユーザ ID 通知の問題」というユーザ存在が相互の事業者に漏洩してしまうという 2 つの問題が発生する．

結合匿名テーブルの問題とは，自事業者のテーブルと結合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という問題である．たとえば事業者 A のテーブル T_A が**表 3** (a)，事業者 B のテーブル T_B が表 3 (b) であったとする．そして，匿名化後の結合匿名テーブル T^* が表 3 (c) であったとする．このとき，T^*（表 3 (c)）では年収 500 万未満は 2 名，T_A（表 3 (a)）も年収 500 万未満は 2 名である．このことから事業者 A は，user1，2 の 2 名は確実に T^*（表 3 (c)）に含まれていると推測できる．さらに，T^*（表 3 (c)）に含まれるユーザは事業者 A と事業者 B の双方に存在する共通ユーザであることから，事業者 A は，user1，2 の 2 名が確実に事業者 B にも存在することが推測できる．それに対し結合匿名テーブルが表 3 (d) のように「600 万」で分割されていた場合，事業者 A は，user1，2，3 の 3 名のうちいずれか 2 名が事業者 B に存在することまでしか推測できない．

ユーザ ID 通知の問題とは，プロトコル中のユーザ ID の

通知によって，相手事業者に自事業者のユーザ存在が知られてしまうという問題である．もし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．すると，通知を受け取った事業者は，通知されたユーザ ID のユーザは通知をしてきた事業者に存在することを容易に推測できてしまう．

3. ユーザ存在の推測の可能性に関する指標の提案

本章では「結合匿名テーブルの問題」を解決するための新たな指標を提案する．2 つのテーブルの比較からのユーザ存在の推測の可能性を示す指標として，*δ-presence* [10] がある．この指標を分散匿名化に適用し，事業者におけるユーザ存在の推測の可能性を示す指標として *δ-max-site-presence* を定義する．

まず，*δ-presence* で定義されているユーザ存在の推測の可能性について説明する．あるテーブル T_1 と T_2 が存在し，T_2 は T_1 に存在する一部のユーザのレコード内のデータから構成されたテーブルとする．また，あるテーブル T のレコード数を $|T|$ と表現する．このとき文献 [10] では，テーブル T_1 に存在するユーザのレコード内のデータが T_2 にも存在する可能性を $|T_2|/|T_1|$ と定義している．

そして，この定義を事業者が保持するテーブルと結合匿名テーブルとの比較の場合に適用する．たとえば事業者 A のテーブル T_A が表 3 (a)，結合匿名テーブル T^* が表 3 (d) であった場合，T_A（表 3 (a)）のうち「年収 600 万未満」は user1，2，3 の 3 名分であり，T^*（表 3 (d)）のうち「年収 600 万未満」は 2 名分である．このとき，T^* は T_A の一部のレコード内のデータから抜き出されたテーブルである．よって先ほどの定義より，事業者 A の user1，2，3 が，T^* にも存在する可能性は 2/3 となる．ここで，T^* は事業者 A，B の双方に存在する共通ユーザのテーブルなので，user1，2，3 が事業者 B に存在する可能性は同じく 2/3 となる．

このような事業者におけるユーザ存在の推測の可能性を示した指標を，*δ-max-site-presence* として定義する．

定義 1（*δ-max-site-presence*） T_A，T_B を事業者 A，B が持つテーブル，T^* を結合匿名テーブルとする．ただし，T_A，T_B にユーザ ID 以外の同一の属性はないものとする．そして，T^* のうち事業者 $n \in \{A, B\}$ が持つ属性の属性値の組合せの集合を $\{v_{n,1}, \cdots, v_{n,m_n}\}$ とし，$v_{n,i} \in \{v_{n,1}, \cdots, v_{n,m_n}\}$ とおく．また，$v_{n,i}$ で識別されるテーブル T_n のレコード数を $|T_n[v_{n,i}]|$，$v_{n,i}$ で識別されるテーブル T^* のレコード数を $|T^*[v_{n,i}]|$ と表現する．このとき，以下の式で示されるように，事業者 n の各 $v_{n,i}$ によるユーザ存在の推測の可能性が δ 以下であるとき，T^* は *δ-max-site-presence* を満たすと定義する．

$$\frac{|T^*[v_{n,i}]|}{|T_n[v_{n,i}]|} \leq \delta \quad \forall v_{n,i} \in \{v_{n,1}, \cdots, v_{n,m_n}\} \quad \forall n \in \{A, B\} \tag{1}$$

たとえば表 3 (d) では，T^* のうち事業者 A の属性の属性値の組合せの集合 $\{v_{A,1}, v_{A,2}\}$ は {年収 600 万未満, 年収 600 万以上} である．そのうち，結合匿名テーブル T^*（表 3 (d)）に「年収 600 万未満」に該当するレコードは 2 名分なので $|T^*[v_{A,1}]| = 2$ となり，事業者 A のテーブル T_A（表 3 (a)）に「年収 600 万未満」に該当するレコードは 3 名分なので $|T_A[v_{A,1}]| = 3$ となる．表 3 (d) は 2/3-*max-site-presence* を満たす．

4. ユーザ存在を隠蔽した分散匿名化プロトコルの提案

本章では 2.3 節で説明した問題を解決する，新たなエージェント間の分散匿名化プロトコルを提案する．提案するプロトコルは以下のプライバシ要件を満たしつつ，できるだけ詳細な結合匿名テーブル T^* を出力するように設計した．

要件 1 結合匿名テーブル T^* は k-匿名性と *δ-max-site-presence* を満たすこと

要件 2 プロトコルの通信内容から T^* 以上の詳しい情報が極力漏れないこと

「結合匿名テーブルの問題」の解決のためには要件 1 を満たす必要があり，「ユーザ ID 通知の問題」の解決のためには要件 2 を満たす必要がある．

提案するプロトコルは，既存の文献 [6] の分散匿名化プロトコルと同様に，Top Down アプローチの匿名化アルゴリズムとして広く利用されている Mondrian [11] をベースにする．なお，各事業者は semi-honest であるとする．これは各事業者はプロトコルを介して得られた情報を解析して相手事業者の情報を知ろうとするが，プロトコルを逸脱した攻撃は行わないという信頼モデルである．各事業者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，この信頼モデルは妥当であると考えた．また，センシティブ属性は片方の事業者だけが保持していることを前提とし，以降では事業者 B が保持しているとして説明する．

4.1 ダミーユーザプロトコル

2.3 節の「ユーザ ID 通知の問題」の解決のためにダミーユーザプロトコルを提案する．このプロトコルでは自事業者に存在しないユーザを，ダミーユーザとして存在するかのように扱う．なお，ダミーユーザに対して，存在するユーザを存在ユーザと呼ぶ．ダミーユーザを導入することにより，通知されるユーザ ID がダミーユーザなのか存在ユーザなのかの区別を困難にできる．

ダミーユーザプロトコルは，文献 [6] の分散匿名化プロ

表 4 内部匿名テーブル T_A^*, T_B^* と結合された結合匿名テーブル T^*
Table 4 Local anonymous table T_A^*, T_B^* and result anonymous table T^*.

(a)事業者Aの内部匿名テーブル T_A^*

	GID	userIDs	年収
初期	1	user1–15	200–499
1回目	2	user1–10	200–499
	3	user11–15	200–499
2回目	4	user1–5	200–399
	5	user6–10	400–499
	3	user11–15	200–499

(b)事業者Bの内部匿名テーブル T_B^*

GID	userIDs	開始時刻	userCounts
1	user1–15	17:00–20:59	–
2	user1–10	17:00–18:59	–
3	user11–15	19:00–20:59	–
4	user1–5	17:00–18:59	Xアニメ:1, Yドラマ:1
5	user6–10	17:00–18:59	Xアニメ:1, Yドラマ:1
3	user11–15	19:00–20:59	Xアニメ:1, Yドラマ:1

(c) 最終の結合匿名テーブル T^*

年収	開始時刻	視聴内容
200–399	17:00–18:59	Xアニメ
200–399	17:00–18:59	Yドラマ
400–499	17:00–18:59	Xアニメ
400–499	17:00–18:59	Yドラマ
200–499	19:00–20:59	Xアニメ
200–499	19:00–20:59	Yドラマ

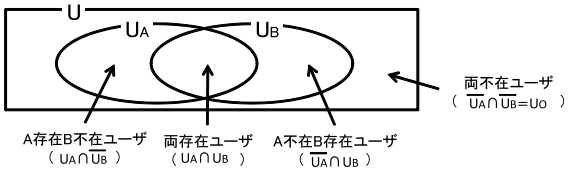

図 2 ダミーユーザと存在ユーザの関係
Fig. 2 Dummy users and existing users.

トコルと同様に分割プロトコルと結合プロトコルで構成される．まず，事業者 A，B が分割プロトコルを実行し，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，事業者 C が結合プロトコルを実行し，事業者 A，B が持つ T_n^* を単純に結合した T^* を取得する．T_n^* の分割と T^* の例を**表 4** に示す．この例では，事業者 A は $T_A(userID,$ 年収 $)$ を，事業者 B は $T_B(userID,$ 視聴開始時刻 , 視聴番組 $)$ を保持している．そして，年収と視聴開始時刻を準識別子，視聴番組をセンシティブ属性として結合匿名テーブル $T^*($ 年収 , 視聴開始時刻 , 視聴番組 $)$ を作成している．

4.1.1 分割プロトコル Step1：ダミーユーザの割当て

分割プロトコルでは最初に，事業者 A と事業者 B が，自事業者のダミーユーザを割り当てる．分割プロトコルでは，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おく．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 としたとき $U = U_A \cup U_B \cup U_O$（$U_O \neq \phi$, $U_A \cap U_B \neq \phi$）となる．このような前提は，たとえば事業者 A，B が Open ID [12] のような同一の認証サーバを利用している場合に成立する．この場合，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 と事業者 B は，事業者 A のダミーユーザを $U - U_A$，事業者 B のダミーユーザを $U - U_B$ と割り当てる．

また，これらのユーザ集合の関係を**図 2** に示す．本論文では，事業者 A，B の両方に存在するユーザを「両存在ユーザ」（つまり「共通ユーザ」），事業者 A に存在するが事業者 B に存在しないユーザを「A 存在 B 不在ユーザ」，逆に事業者 B に存在するが事業者 A に存在しないユーザを「A 不在 B 存在ユーザ」，事業者 A，B の両方に存在しないユーザを「両不在ユーザ」と呼ぶ．この図に示したように，事業者 A のダミーユーザ（$U - U_A$）は A 不在 B 存在ユーザ（$\overline{U_A} \cap U_B$）と両不在ユーザ（$\overline{U_A} \cap \overline{U_B}$）の和集合となり，事業者 B のダミーユーザ（$U - U_B$）は A 存在 B 不在ユーザ（$U_A \cap \overline{U_B}$）と両不在ユーザ（$\overline{U_A} \cap \overline{U_B}$）の和集合となる．

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（表 4 (a), (b) 上）．各事業者は内部匿名テーブルの分割を繰り返すことで匿名化を行う．内部匿名テーブルが持つ属性は $T_A^*(GID, userIDs, QID_A)$, $T_B^*(GID, userIDs, QID_B, userCounts)$ である．ここで GID とは，シーケンシャルに割り当てられる T_n^* の各レコードの識別子である．$userIDs$ とは，T_n^* のレコードに該当するユーザ ID の集合である．QID_A と QID_B とは，事業者 A,B が持つ準識別子である．$userCounts$ とは，$userIDs$ で示されたユーザ集合におけるセンシティブ属性の各属性値の共通ユーザ数であり，分割がすべて完了してから計算される．なお，T_n^* の初期化時は，ダミーユーザの準識別子（QID_A，QID_B）の属性値は，各属性の最小値が割り当てられているとして扱われる．そして T_n^* の $userIDs$ には，ダミーユーザが含まれるように初期化が行われる．

4.1.2 分割プロトコル Step2：分割点の決定と分割処理

続いて，エージェント間で通信を行い，事業者 A の主導により T_A^* と T_B^* を分割していく分割処理を行う（**図 3**）．この分割処理は Mondrian と同様に，分割対象となるユーザ集合を分割後に，分割後のユーザ集合を次の分割対象として再帰的に処理を呼び出す．

まず，ダミーユーザの準識別子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，相手事業者から見て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（U_A, U_B）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 4.2 節で説明する．そして，決定した分割点

function split(U_p:分割対象となるユーザ集合の $userIDs$)
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（MPC を利用）
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（MPC 利用）
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 $userIDs$ を相手の事業者へ送信
9: **else**
10: 相手から分割後の $userIDs$ を受信し，T_n^* を分割
11: **endif**
12: U_{hi}, U_{low} ⇐ 分割後の $userIDs$．split(U_{hi}),split(U_{low}) を再帰呼出し

図 3 分割プロトコルの Step2（分割処理）のアルゴリズム
Fig. 3 Algorithm of Step 2 (split function) in split protocol.

で分割しても k-匿名性と δ-max-$site$-$presence$ を満たせるかを，2 つの MPC を用いて確認する．まず k-匿名性を満たせるかを確認する MPC は，事業者 A，B から入力として分割後のユーザ集合における存在ユーザのユーザ ID の集合を受け取り，その積集合の人数が k 以上であるかを出力する処理である．続いて δ-max-$site$-$presence$ を満たせるかを確認する MPC は，入力は先ほどと同じで，その積集合の人数が「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 × δ」以下であるかを出力する処理である．これらの MPC を用いて k-匿名性と δ-max-$site$-$presence$ を満たせるかを確認する．

そして，指標を満たしている場合のみ T_A^*，T_B^* を分割する．1 回目の分割の様子を表 4 (a)，(b) 中に示す．この分割の分割点は「事業者 B」の「視聴開始時刻」の「19:00」である．この場合，まず分割点の準識別子を持つ事業者 B の T_B^* を分割する（表 4 (b) 中）．そして，分割前の GID と，分割後の GID と $userIDs$ を事業者 A に送信する．事業者 A は，受け取った GID と $userIDs$ に従って T_A^* を分割する（表 4 (a) 中）．最後に，分割後の $userIDs$ に対して再帰的に上記の分割処理を繰り返していく．2 回目の分割の例を表 4 (a)，(b) 下に示す．この例は，「事業者 A」の「年収」を「400 万」で分割した例である．

4.1.3 分割プロトコル Step3：ダミーユーザの削除

すべての分割処理が完了したら MPC を用いてダミーユーザを削除し，$userCounts$ を計算する．この MPC は，T_n^* の各レコードのセンシティブ属性の各属性値 s について，事業者 A からの入力として事業者 A の存在ユーザのユーザ ID の集合，事業者 B からの入力として事業者 B で s を持つ存在ユーザのユーザ ID の集合を受け取り，その積集合の個数を出力する処理である．たとえば表 4 (b) 下の user1-5 のレコードでは，事業者 A の存在ユーザのユーザ ID の集合と「X アニメ」を視聴した事業者 B の存在ユーザのユーザ ID の集合が入力として与えられ，積集合の個数が 1 として出力された場合である．

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 は内部匿名テーブル T_A^*，T_B^* を分割していく．

4.1.4 結合プロトコル

最後に，結合匿名テーブル T^* を取得する事業者 C が，事業者 A，B から T_A^*，T_B^* を取得して結合を行う．まず，事業者 A，B は T^* の $userIDs$ を削除する．続いて，GID から分割の順番を知られないように，事業者 A が主導して GID をランダムに並べ変え再度シーケンシャルな番号を振り直し，GID の振り直し指示を事業者 B に送信する．そして事業者 B は，指示に従って T_B^* の GID を更新する．その後，事業者 C は $userIDs$ が削除され GID が振り直しされた T_A^*，T_B^* を受信し，GID をキーにして結合を行うことで結合匿名テーブル T^* を得る（表 4 (c)）．

4.2 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のための分割点決定関数を提案する．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．そのためには，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．たとえば表 3 (c) では，事業者 A から見ると年収 500 万未満は user1，2，年収 500 万以上は user3，6，7，8 である．このうち事業者 B のダミーユーザは user3，8 であるため，ダミーユーザは偏っている．それに対し表 3 (d) はダミーユーザは偏っていない．

そこで，ダミーユーザのエントロピー（シャノンの平均情報量）を導入する．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．ダミーユーザのエントロピー（Dummy Entropy，DE）を，

以下のように定義する．

$$DE(c,n) = -\sum_{U_i \in \{U_{hi}, U_{low}\}} \frac{|dummy(n,U_i)|}{|U_i|} \cdot \log\left(\frac{|dummy(n,U_i)|}{|U_i|}\right) \quad (2)$$

ここで c は分割点候補であり，分割前のユーザ集合 U_p を上位 U_{hi} と下位 U_{low} へ分割する属性値を意味する．また，$dummy(n,U_i)$ はユーザ集合 $U_i \in \{U_{hi}, U_{low}\}$ から事業者 n のダミーユーザを抜き出したユーザ集合である．このように定義することで，分割後のユーザ集合におけるダミーユーザの偏りが小さくなるときに DE の値が大きくなる．

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．

$$S(c_i) = \alpha\left(\frac{-L(c_i)}{\max_{x_j \in X}(L(x_j))}\right) + (1-\alpha)\frac{1}{2}\sum_{n \in A,B}\left(\frac{DE(c_i,n)}{\max_{x_j \in X}(DE(x_j,n))}\right) \quad (3)$$

$$L(c_i) = \sum_{x_j \in X} |x_j - c_i| \quad (4)$$

ここで α（$0 \le \alpha \le 1$）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，$\alpha = 1$ としたときは c_i が median のときに 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．そして 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おける，ユーザ数に対する事業者 A，B のダミーユーザ数の割合の偏りがほぼなくなるように分割が行われ，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．

4.2.1 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 MPC を用いる．まず，分割点の属性を選ぶ処理で MPC を用いる．この MPC は，事業者 A，B からの入力としてローカルで計算した最大の normalized range を受け取り，どちらが大きいかを出力する処理である．

次に，分割点候補 c_i の DE を計算する際の相手事業者 n のダミーユーザ数（$|dummy(n,U_i)|$）の計算で MPC を用いる．この MPC は，入力として，分割を行う事業者の分割後のユーザ U_i のユーザ ID の集合と，相手事業者 n のダミーユーザのユーザ ID の集合とを受け取り，その積集合の個数であるダミーユーザ数（$|dummy(n,U_i)|$）を出力する処理である．このとき，分割後のユーザ数（$|U_i|$）も一緒に出力する．ただし，この MPC については相手事業者 n にだけ出力する．つまり，たとえば 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ることができるが，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．この MPC により，DE の計算に必要な情報を得ることができるため，DE を事業者内でローカルに計算できる．

最後に，分割点を決定する処理で MPC を用いる．この MPC は，事業者 A，B 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を受け取り，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である．これらの MPC を用いて分割点を決定する．

5. 評価実験

提案プロトコルを実行するエージェントをプロトタイプ実装し，既存手法となるプロトコルと比較することで有効性を評価する．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，文献 [5] の分散匿名化プロトコルの評価と同様に，UCI の Adult データ [13] 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）（$50K 以上 or 未満）を持つ約 3 万レコードのデータである．14 種類の属性を準識別子，年収分類をセンシティブ属性とした．そして，全レコードを約 3 万名の母集団ユーザ（U）としてとらえ，ランダムに並べ変えた上位レコードから事業者 A と事業者 B に存在するユーザ（両存在ユーザ，$U_A \cap U_B$），事業者 A と事業者 B の片方だけに存在するユーザ（（A 存在 B 不在ユーザ，$U_A \cap \overline{U_B}$），（A 不在 B 存在ユーザ，$\overline{U_A} \cap U_B$））），残りを双方に存在しないユーザ（両不在ユーザ，$\overline{U_A} \cap \overline{U_B} = U_O$）とした．つまり，事業者 A の存在ユーザは両存在ユーザと A 存在 B 不在ユーザとなり，これらのユーザのレコードが T_A に格納される．同様に事業者 B の存在ユーザは両存在ユーザと A 不在 B 存在ユーザとなり，これらのユーザのレコードが T_B に格納される．また，ダミーユーザは 4.1.1 項で説明した方法で割り当てられ，ダミー値は各事業者の存在ユーザの準識別子の分布に沿って割り当てられる．なお，「事業者 A と事業者 B の片方だけに存在するユーザ」は，「事業者 A と事業者 B の片方だけに存在する事業者 B と事業者 A のダミーユーザ」でもある．以降ではこのユーザを「片方存在ダミーユーザ」と表記する．実験では，共通ユーザを 1,200 名で固定し，片方存在ダミーユーザを 600〜12,000 名に変化させて評価を行った．評価はそれぞれ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 [11]．

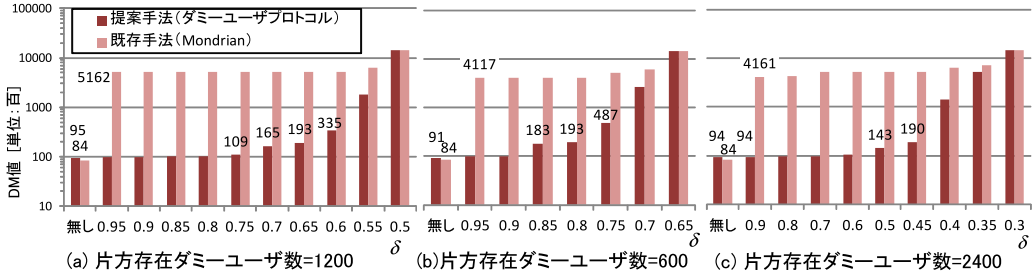

図 4 *δ-max-site-presence* に対する既存手法との比較
Fig. 4 Dummy user protocol vs. Mondrian.

評価指標は，文献 [10] と同じく Discernibility Metric (DM) [14] 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．DM の値は，$qids$ を T^* における準識別子の属性値の集合とおくと，以下の式で計算される[*1]．

$$DM = \sum_{q_i \in qids} |T^*[q_i]|^2 \quad (5)$$

たとえば，1,200 名のデータがきれいに 8 名[*2]ごとに 150 個に分割されている場合は $8^2 \times 150 = 9600$ となる．また，DM 値は人数を 2 乗しているため分割に偏りがあると急激に悪い値となる．

5.1 ダミーユーザプロトコルの有効性評価

最初に，提案手法となるダミーユーザプロトコルの有効性を評価するために，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 との比較を行う．この分散対応 Mondrian は，提案手法と比較するために k-匿名性だけでなく *δ-max-site-presence* も満たしている際に分割を行い，最終結果では共通ユーザだけを出力する分散匿名化プロトコルである．

5.1.1 *δ-max-site-presence* に対する有効性評価

まず，片方存在ダミーユーザを 1,200 名，$k = 2$ として，*δ-max-site-presence* の δ を変化させた際の評価結果を**図 4** (a) に示す．なお，重み α は 0.5 として DE の影響を半分にしている．この結果が示すとおり，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．特に δ が 0.7 付近までは 1〜2 万程度の DM 値であるため良い結果であるといえる．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ることができなかったからである．

5.1.2 ダミーユーザ数に対する有効性評価

次に，片方存在ダミーユーザの人数の割合を変えた場合の有効性を評価する．ダミーユーザプロトコルは，片方存在ダミーユーザによってユーザ存在を隠蔽するという手法であるため，片方存在ダミーユーザの人数が重要となる．そこで，実験に用いる評価データにおける片方存在ダミーユーザ（A 存在 B 不在ユーザ，A 不在 B 存在ユーザ）の人数を 2,400 名とした場合と，600 名とした場合で同様に評価を行った．結果を図 4 (b)，(c) に示す．この結果から分かるように，片方存在ダミーユーザ数が増えると δ が 0.5 付近までは低い DM 値に保つことができるが，片方存在ダミーユーザ数が減ると 0.7 付近までしか低い DM 値を保つことができなくなる．これは，片方存在ダミーユーザが減ることで指定された *δ-max-site-presence* を満たすことができなくなり，分割が終了するためである．なお，片方存在ダミーユーザ数を 12,000 名と大幅に増やして評価を行うと，δ が 0.3 付近でも DM 値は悪化しない．つまり，片方存在ダミーユーザ（A 存在 B 不在ユーザ，A 不在 B 存在ユーザ）が共通ユーザ（両存在ユーザ）に対して多い場合は，δ を小さく設定しても精度の急激な低下はなく，よりユーザ存在を隠蔽したいユースケースにも対応できる．それに対し，片方存在ダミーユーザの人数が小さい場合は，たとえば，共通ユーザの一部を存在しないユーザとして扱い，共通ユーザに対する片方存在ダミーユーザの人数を相対的に大きくするような処理が考えられる．しかし，このような方法では共通ユーザが減るため T^* のレコード数が減り，情報量が低下し，情報の正確性が低下する．よって，片方存在ダミーユーザの人数が少ない場合には，ユーザ存在の漏洩の確率を下げたいユースケースへの対応は困難である．

5.1.3 ダミーユーザ数に応じた重み調整

さらに，分割点決定関数の重み α の最適値を調べるため

[*1] 文献 [14] ではレコード削除（suppression）をした際の DM 値も定義されているが，提案手法ではレコード削除は行わないので無視している．
[*2] データマイニングでは大まかな傾向が分かればよいため，8 名程度の分割であってもマイニング結果に影響が少ないと考えられる．

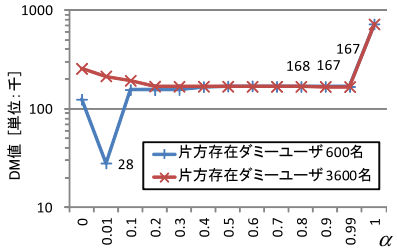

図 5 α を変化させた場合の DM 値
Fig. 5 DM in several α.

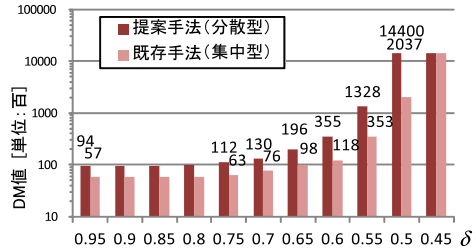

図 6 既存の集中型との比較
Fig. 6 Distributed vs. centralized.

に，α を変化させて評価を行った．**図 5** に片方存在ダミーユーザ数を 600 名と 3,600 名にした場合について，α を変化させた際の DM 値を示す．なお，片方存在ダミーユーザ数に合わせて δ は 0.75 と 0.3 とおいた．この結果から分かるとおり，片方存在ダミーユーザ数が多い場合は α が大きい（DE の影響が小さい）方が DM 値が良くなる．それに対し，片方存在ダミーユーザ数が少ない場合は α が小さい（DE の影響が大きい）方が DM 値が良くなる．これは，片方存在ダミーユーザ数が少ない場合はダミーユーザ数のわずかな偏りによって，*δ-max-site-presence* を満たさなくなりやすいので，DE が重要になるためである．つまり，片方存在ダミーユーザ数に応じて α を設定するとよいことが分かる．

5.2 分散型のダミーユーザプロトコルと集中型との比較

次に，集中型（非分散環境の匿名化）でのユーザ存在の隠蔽手法 [10] である既存手法と比較し，分散型（分散環境の分散匿名化）に対応したダミーユーザプロトコルの有効性を評価する．集中型での既存手法は，あるテーブルと匿名テーブルにおけるユーザ存在を隠蔽する手法であり，提案手法のように事業者 A と事業者 B の双方からみた，ユーザ存在の推測を防ぐというものではない．そこで，公平な評価を行うために事業者 A だけに存在するユーザを 1,200 名，事業者 B にだけ存在するユーザを 0 名としてデータを生成し，評価を行った．

図 6 に，δ を変化させた際の集中型の既存手法 [10] と分散型の提案手法の DM 値を示す．提案手法は δ が 0.6 付近で DM 値が悪化するのに対し，既存手法は低い DM 値を保っている．これは，分散型の提案手法は，分割点決定関数を用いて分割点を探索するアルゴリズムであるのに対し，集中型の既存手法は，分割点候補に対して実際に分割を行った際にユーザ存在を隠蔽可能であるかを何度も確認し，分割点を探索するアルゴリズムのためである．もし分散型でこのような探索を行うと，分割可能かどうかの情報からユーザ存在が知られてしまう．しかし，提案手法でも δ が 0.7 程度であれば既存手法とほぼ同等の DM 値となっている．このことから，提案手法は δ が特に小さい場合を除き，既存手法とほぼ同等の効果が得られると期待できる．また，分散型の提案手法の適用限界（$\delta = 0.7$ 付近）は，集中型の既存手法の適用限界（$\delta = 0.6$ 付近）とほぼ同等であるため，問題ないと考える．

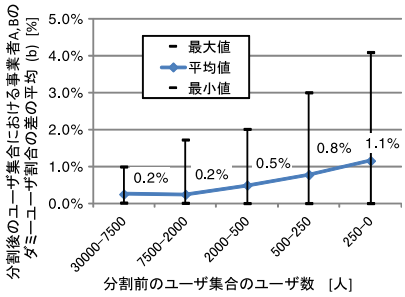

図 7 ダミーユーザの偏り
Fig. 7 Bias of dummy users.

5.3 ダミーユーザの偏りの程度の評価

次に，提案の分割点決定関数によって決定された分割点における，分割後のユーザ集合のダミーユーザの偏りがどの程度であるかを評価した．**図 7** に，分割前のユーザ集合のユーザ数に対する，事業者 A，B のダミーユーザの偏りを示す．このグラフではダミーユーザの偏り b を，以下の式で示したように，分割点 c で分割後の 2 つのユーザ集合（U_{hi}，U_{low}）における，ユーザ数（$|U_{hi}|$，$|U_{low}|$）に対する事業者 $n \in \{A, B\}$ のダミーユーザ数（$|dummy(n, U_{hi})|$，$|dummy(n, U_{low})|$）の割合の差を計算し，その値の事業者 A，B での平均としている．

$$b(c) = \sum_{n\in\{A,B\}} \left| \frac{|dummy(n, U_{hi})|}{|U_{hi}|} - \frac{|dummy(n, U_{low})|}{|U_{low}|} \right| \Big/ 2 \qquad (6)$$

なお，このグラフは，$\delta = 0.7$ とし，その他のパラメータは 5.1.1 項の評価と同じにして，提案手法を 10 回実行し，分割前のユーザ集合のユーザ数を適切な区間で区切って偏り b の平均値，最大値，最小値を計算した結果である．

図 7 に示したように，分割が進み，ユーザ集合が小さくなるに従って，ダミーユーザの偏り b が大きくなる傾向がある．これは，分割対象のユーザ集合が小さいと分割点候補が少なくなってしまうので，偏りが小さくなるような分割点を選べなくなるためである．しかし，ダミーユーザの

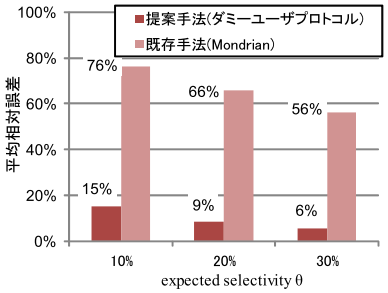

図 8 ユーザ数カウントの平均誤差
Fig. 8 Average relative error.

偏り b の平均は約 1%程度であり，小さいと考えられる．

このように，提案の分割点決定関数によって，分割後のユーザ集合における事業者 A，B のダミーユーザの偏りが小さくなるような分割点が選ばれることが確かめられた．

5.4 ユーザ数を得るクエリにおける相対誤差の評価

次に，結合匿名データに対してデータマイニングを行った場合に，マイニング結果にどの程度の誤差が発生するのかについての評価を行った．この評価では，既存の匿名化の研究 [15] と同様に，データマイニングにおける基本的な集約クエリ（*aggregate query*）とされている，ある条件に合致するユーザ数をカウントするクエリ（“select count(*) from T^* where 条件部”）の結果の相対誤差（*relative error*）を計測する手法で評価を行った．この評価手法は，まずカウントされるユーザ数の割合の期待値（*expected selectivity*）を θ（$0\% < \theta < 100\%$）とおいて，条件部に指定する検索範囲が全体の θ 倍になるようなクエリをランダムに生成する．つまり，T^* に含まれるユーザ数が 1,200 であった場合，$\theta = 10\%$としたクエリで検索されるユーザ数（レコード数）は約 120 となる．そして，このように生成したクエリについて，匿名化前の結合テーブル（T_A と T_B の共通ユーザのレコードを単純に結合したテーブル）に対して得られたユーザ数を act，結合匿名テーブル（T^*）に対して得られたユーザ数を est とし，その相対誤差を $|act - est|/act$ で計算する．なお，est はクエリの条件部に記載された範囲と，汎化された値の重なり度合いに応じて算出する．たとえば T^* に「100～200 万」というレコードが 5 個あり，クエリが「100～120 万」であった場合は，このクエリは「100～200 万」の 20%が重なっているので，$est = 5 \times 20\% = 1$ となる．なお，条件部に利用する属性は 3 つとし，ランダムに選択した．

図 8 に，θ を 10～30%とおいて，ランダムなクエリを 10,000 回生成し，既存手法と提案手法それぞれについて評価を実施した際の相対誤差の平均を示す．なお，この評価では $\delta = 0.7$ とし，その他のパラメータは 5.1.1 項の評価と同じにしている．この結果が示すように，既存手法の相対誤差は約 55～75%と大きいが，提案手法の相対誤差は約 5～15%程度と小さい．この提案手法の相対誤差は，たとえば相関ルールマイニングを行った際に，得られる相関ルールの支持度（support）や確信度（confidence）の相対誤差が約 5～15%程度であることを意味している．つまり，匿名化前の結合テーブルに対して得られた相関ルールの支持度が 10%であった際に，結合匿名テーブル（T^*）に対して得られた相関ルールの支持度に約 15%の相対誤差が入ることになり，約 8～12%の支持度になることを意味している．この程度の誤差であれば，得られる相関ルールに大きな差はないと考える．このように提案手法がマイニング結果に与える影響は小さいと考えられるため，提案手法は十分良い結果であると考える．

5.1～5.4 節の評価結果より，片方だけに存在するユーザ（片方存在ダミーユーザ）の人数がある程度存在し，ユーザ存在の推測がある程度許させるようなユースケースであり δ を小さく設定する必要がない場合は，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．

6. 安全性の評価

本章では，ダミーユーザプロトコルの安全性について評価を行う．ここで安全であるとは，プロトコルの通信内容から，想定されている以上の情報を得ることができないことをいう．まず，事業者 A，B 間の分割プロトコルの通信内容から得られる情報が，プロトコルの実行結果である内部匿名テーブル T_n^*（$n \in A, B$）と，途中計算の結果である 2 つの中間情報だけであることを証明する．次に，これらの情報はプライバシ上の問題が小さいことを示す．

6.1 プロトコルから得られる情報

事業者 A，B 間のプロトコルの通信内容から，事業者 A が事業者 B の秘密の情報を得ることができないことを証明するには，事業者 A が受信する通信内容（事業者 B が送信する通信内容）をシミュレートするシミュレータ S が存在し，S に対する入力として事業者 A が事業者 B との通信内容から得られると想定されている情報と事業者 A が元々持っている情報を与え，S が通信内容をシミュレートできることを示せばよい [6], [8]．なぜなら，S がシミュレートした通信内容には入力として与えられた情報以外の情報がいっさい含まれていないため，シミュレートされた通信内容を受信する事業者 A は，S に入力された情報以外の情報を得ることができないからである．また，提案のプロトコルでは MPC を用いているため，Composition Theorem [8] を用いて証明を行う [6]．Composition Theorem とは，プロトコル F を安全なプロトコルブロック $f_1 \ldots f_n$ で構成できるとしたとき，$f_1 \ldots f_n$ を信頼できる第三者（Trusted Third Party，TTP）を介した通信に置き換えたものが安全であれば F も安全であるという定理である．本証明で

は，まず分割プロトコルのMPCをTTPを介した通信に置き換えたプロトコルにおいて，プロトコルの通信内容をシミュレートできることを証明する．その後，Composition TheoremによってTTPを介した通信をMPCで置き換えても，同様にプロトコルが安全であることを示すことで証明を行う．

定理1 事業者 $n \in \{A, B\}$ は，ダミーユーザプロトコルの分割プロトコルの通信内容から，T_n^* と中間情報1，2以外の情報を知ることができない．

- **中間情報1**：相手事業者の分割点候補における，分割後のユーザ数と分割後の自事業者のダミーユーザ数（ユーザIDは漏洩しない）
- **中間情報2**：自事業者でキャンセルされた分割点における，属性と属性値と満たされなかった指標（k-匿名性 or δ-*max-site-presence*）

証明1 事業者 $n \in \{A, B\}$ が元々持つ T_n と T_n^* と中間情報1，2から，事業者 n が受信する通信内容をシミュレートできることを示せばよい．初めに，T_A と中間情報1，2と T_A^* から，事業者 A が受信する通信内容をシミュレートできることを示す．まずシミュレータは，T_A^* の各レコードの *GID* の値がシーケンシャルに割り当てられていることを利用して，分割を逆順にたどることでどのような分割が行われたかを分析する．この分析では，T_A^* のうち *GID* の値が最も大きい2つのレコードが最後に行われた分割であり，この分割の分割前のレコードは歯抜けになっている *GID* のうち最も大きな値のレコードであると判断する（例：表4(a)下では2が歯抜けになっている最も大きな値であるため，「$2 \Rightarrow 4,5$」という分割が行われたことが分かる）．そして，この2つのレコードを比較することで，分割が行われた事業者と分割後のuserIDを判断できる．また，分割が事業者Aで行われた場合は分割点の属性と属性値も分かる（例：表4(a)下では「事業者A」で「年収」「400万」で分割され，user1〜5とuser6〜10に分割されたことが分かる）．この処理を繰り返すことで，最初の分割までたどることが可能であり，どのような分割が行われたかを分析できる．

次に，分析した分割情報を使って通信内容のシミュレートを開始する．ダミーユーザプロトコルはStep2とStep3で通信を行う．特にStep2では，分割点決定関数の計算と，指標確認と，userIDの通知の際に通信を行う．最初に，Step2の1回目の分割における事業者Aが受信する情報をシミュレートする．Step2の分割点決定関数の計算で行われる通信内容は，分割を行う事業者（事業者A，Bが受信）と，分割点候補における分割点のユーザ数と分割後の自事業者のダミーユーザ数（分割を行わない事業者が受信）と，決定した分割点の属性と属性値（分割を行う事業者だけが受信）である．これらの情報は，分析した分割情報と中間情報1から分かる情報である．シミュレータは，これら情報から事業者Aが受信する情報を抜き出してシミュレートを行う．続いて，分析した分割情報から分割がさらに続くかを判断する．もし分割後のGIDについてさらに分割が続く場合は，Step2の指標確認をOKとしてシミュレートする．そして，この分割が事業者Bで行われていた場合は，Step2の分割後のuserIDの通知をシミュレートする．その後，分割後のGIDについて上記処理を再帰的に繰り返す．

もし分割が続かない場合は，1度指標確認をOKとしてシミュレートした後，中間情報1と2を使って，先ほどと同様に分割点決定関数の計算で行われる通信内容をシミュレートする．ただし先ほどとは違って，分析した分割情報に，分割点の事業者と属性と属性値は含まれていないため，代わりに中間情報2を利用する．そして，その後の指標確認では中間情報2を使って k-匿名性か δ-*max-site-presence* をNGとするシミュレートを行う．このような処理を繰り返すことで，事業者 A が受信する通信内容をシミュレートできる．

続いて，T_B と中間情報1，2と T_B^* から，事業者 B が受信する通信内容をシミュレートできることを示す．Step2については，先ほどと同様にシミュレートできる．Step3の *userCount* については，T_B^* に情報があるためシミュレートできる．

以上のように，TTPを利用した場合の分割プロトコルで，事業者 $n \in \{A, B\}$ が持つ T_n と中間情報1，2と T_n^* から，事業者 n が受信する通信内容をシミュレートできるため，プロトコルの通信内容から T_n^* と中間情報1，2以外の情報の漏洩はない．またComposition Theoremにより，TTPによる計算をMPCに置き換えても分割プロトコルは安全であるといえる．よって，事業者 n は分割プロトコルの通信内容から，T_n^* と中間情報1，2以外の情報を知ることはできない． □

6.2 プロトコルから得られる情報のプライバシ性

T_n^* は事業者 n の相手事業者が持つ属性は含まれない．また，中間情報1，2はユーザIDが含まれないため，どのユーザの情報であるかを知ることはできない．さらに，中間情報1，2と内部匿名テーブル T_n^* は，事業者Cには知られることはなく，事業者A，Bに知られる情報である．事業者A，Bは，実際のビジネスにおいてある程度の契約関係があることが想定されることや，中間情報1，2や内部匿名テーブルの情報からのセンシティブ属性の属性値やユーザ存在の確定はないことから，プライバシ上の問題は低いと考える．

7. 関連研究

分散匿名化は，パーソナル情報の分割の形態の違いにより垂直分割と水平分割に分類される．垂直分割とは，ユー

ザのパーソナル情報が，属性ごとに異なる事業者に保持されている分割形態である．水平分割とは，ユーザのパーソナル情報が，ユーザごとに異なる事業者に保存されている分割形態である．

垂直分割での分散匿名化としては文献 [5], [7] などが存在する．文献 [5] では，本論文と同じ Top Down アプローチと MPC を組み合わせた手法で，複数事業者間での分散匿名化を実現している．それに対し文献 [7] では，Bottom Up アプローチを採用している．これは，それぞれの事業者で個別に内部匿名テーブルを生成した後，結合匿名テーブルの匿名性が保たれることを確認しながら内部匿名テーブルを結合していく手法である．

文献 [6] では，水平分割での分散匿名化で発生するパーソナル情報の保存形式の違いから，情報の保存場所を知られてしまうという問題を，Top Down アプローチで解決している．また，この問題を解決するため *ℓ-site-diversity* という指標を提案しており，本論文では，指標設計や分割点決定関数の設計で，考え方を参考にしている．

また，分散匿名化ではないが公開テーブルと匿名テーブルにおいてユーザ存在の隠蔽を目指した匿名化技術の研究が行われている．文献 [10] では，*δ-presence* というユーザの存在の可能性を示す指標と，その指標を満たすための匿名化アルゴリズムを提案している．しかしこのアルゴリズムは分散匿名化ではないため，事業者間でユーザが異なる場合におけるユーザ存在の隠蔽課題には適用できない．一方，提案している指標は分散匿名化にも適用可能である．*δ-presence* を分散匿名化に適用した指標を *δ-max-site-presence* として本論文で定義している．

8. まとめと今後の課題

本論文では，クラウド上のサービス事業者が保持するパーソナル情報に対して，ユーザ存在を隠蔽した分散匿名化のプロトコルを実行するパーソナル情報保護エージェントを提案した．そして，ユーザ存在が推測される可能性を示した *δ-max-site-presence* という指標を提案し，この指標を満たすことができるダミーユーザプロトコルを提案した．評価の結果，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，分割点決定関数のさらなる改良を行い有効性の向上を図る予定である．また，提案プロトコルでは MPC を用いていたが，MPC を単純に適用すると計算量や通信量が多くなってしまうことが知られている．今後は MPC を，Secure Set Intersection [16] や安全な近傍検索プロトコル [17] などの MPC よりも処理の軽い暗号プロトコルに置き換えて，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

[1] 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).

[2] 佐久間淳，小林重信：プライバシ保護データマイニング，人工知能学会誌，Vol.24, No.2, pp.283–294 (2009).

[3] Fung, B., Wang, K., Fu, A. and Yu, P.: *Privacy-Preserving Data Publishing: Concepts and Techniques*, chapter 11–12, CRC Press (2010).

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

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

[6] Jurczyk, P. and Xiong, L.: Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers, *Proc. DBSec'09*, pp.191–207, Springer (2009).

[7] Jiang, W. and Clifton, C.: Privacy-Preserving Distributed k-Anonymity, *Proc. DBSec'05*, pp.166–177, Springer (2005).

[8] Goldreich, O.: *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press (2004).

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

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

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

[12] OpenID Foundation: *OpenID Authentication 2.0* (2007).

[13] Blake, C.L. and Merz, C.J.: UCI Repository of machine learning databases (1998).

[14] Bayardo, R.J. and Agrawal, R.: Data Privacy through Optimal k-Anonymization, *Proc. ICDE'05*, pp.217–228, IEEE (2005).

[15] Xiao, X. and Tao, Y.: m-Invariance: Towards Privacy Preserving Re-publication of Dynamic Datasets, *Proc. SIGMOD'07*, pp.689–700, ACM (2007).

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

[17] 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).

竹之内 隆夫 （正会員）

2003 年電気通信大学電気通信学部情報工学科卒業．2005 年同大学大学院情報システム学研究科博士前期課程修了．同年日本電気（株）入社．現在，情報・ナレッジ研究所主任．2011 年電気通信大学大学院情報システム学研究科博士後期課程入学．主としてパーソナル情報の利活用におけるプライバシ保護の研究に従事．電子情報通信学会会員．

川村 隆浩 （正会員）

1992 年早稲田大学理工学部電気工学科卒業．1994 年同大学大学院理工学研究科電気工学専攻修士課程修了．同年（株）東芝入社．現在，同社研究開発センター主任研究員．工学博士．2001～2002 年米国カーネギーメロン大学ロボット工学研究所客員研究員．2003 年より電気通信大学大学院情報システム学研究科客員准教授．2007 年より大阪大学大学院工学研究科非常勤講師．主としてマルチエージェントシステム，セマンティック Web の研究・開発に従事．人工知能学会会員．

大須賀 昭彦 （正会員）

1981 年上智大学理工学部数学科卒業．同年（株）東芝入社．同社研究開発センター，ソフトウェア技術センター等に所属．1985～1989 年（財）新世代コンピュータ技術開発機構（ICOT）出向．2007 年より電気通信大学大学院情報システム学研究科教授．工学博士（早稲田大学）．主としてソフトウェアのためのフォーマルメソッド，エージェント技術の研究に従事．1986 年度情報処理学会論文賞受賞．現在，IEEE Computer Society Japan Chapter Chair，人工知能学会理事．電子情報通信学会，人工知能学会，日本ソフトウェア科学会，IEEE CS 各会員．