情報処理学会論文誌　データベース　Vol. 1　No. 1　79–87 (June 2008)

放送暗号とクラスタリングを用いたデータ共有型 Web アプリケーションにおけるプライバシ保護

川 本 淳 平[†1]　吉 川 正 俊[†1]

Web アプリケーションは，従来デスクトップ上で実装されていたアプリケーションを Web 上で利用可能にしただけでなく，ユーザ同士が簡単にデータ共有を行え，コラボレーションによって新たな知を生み出せる枠組みを提供している．しかし，こうした Web アプリケーションにおいては，サーバにプライバシ情報が収集されてしまうという問題が指摘されている．本研究では，この問題に対処するために放送暗号を利用し，サーバ上ではデータを暗号化したまま，グループ内でデータ共有を可能にする仕組みを提案する．また，Web アプリケーションが提供するアカウントの機能を用いたクラスタリングにより復号コストの削減を行う．

Privacy Protection of Data Shared on Web Applications Using Broadcast Encryption and Clustering

Junpei Kawamoto[†1] and Masatoshi Yoshikawa[†1]

Web applications provide not only services on the Web but also the place in which the users share data easily and create new wisdom by collaboration. At the same time application servers are collecting user data implicitly for render of service. Therefore, users' privacy data might be collected by servers. To overcome this problem, in this paper, we propose a system that makes us share encrypted data in the group using pairing based broadcast encryption. We also discuss retrenchment of decryption costs by data clustering using account service of servers.

†1 京都大学大学院情報学研究科社会情報学専攻
Department of Social Informatics, Graduate School of Informatics, Kyoto University

1. はじめに

従来デスクトップアプリケーションとして提供されていたサービスを，Web を通しブラウザ上で実現する Web アプリケーションが急速に広まっている．こうした Web アプリケーションの特徴として次の 3 つをあげることができる．

(1)　データが Web サーバ上で管理されるため，わざわざ持ち運ぶ必要がない．
(2)　Web に接続できる端末があれば，いつでもどこからでもアプリケーションを利用できる．
(3)　データが 1 カ所で管理されるため，他のユーザとの共有・連携が簡単に行える．

特に，3 番目にあげたユーザ間でのデータ共有および連携の容易さから，Web アプリケーションを利用したコラボレーションが活発に行われ，新たなコンテンツが創出されている．しかし同時に，Web アプリケーションを提供するアプリケーションサーバには，サーバ自身がユーザデータを収集し別の目的に利用することがあるという問題が指摘されている．そのため，Web アプリケーションの利用により，プライバシ情報が漏洩してしまうことが危惧されている．

Web アプリケーションは，その特徴により，人々がいつでも，どこにいてもコラボレーションを行うことを可能とする．すなわち，ユビキタス・コラボレーションを実現するためには欠かせないサービスである．人々が安心して Web アプリケーションを利用するためには，アプリケーションサーバ側が先のプライバシ問題解決に協力し次の機能を提供することが必要である．

- ユーザデータは暗号化してサーバへ送信する．
- サーバでは暗号化されたまま保存し，サーバ自身によるデータの閲覧を不可能にする．
- 暗号化データに対してアプリケーションを実行するために必要な問合せをサポートする．

しかし，現在そのような機能を提供している Web アプリケーションは我々の知る範囲ではない．そのため，現在提供されている Web アプリケーションにおいて，そのアプリケーションが提供する機能のみを利用し，サーバに特別な機能を仮定することなく上記プライバシ保護要件を実現することが本研究の目的である．

プライバシ情報の漏洩を防ぐには，**図 1** に示すようにデータを暗号化して送信するフィルタリング機能をローカル端末上で動作させることが有効である．外部のサービスを用いず，ローカル端末で暗号化することで，最も高いレベルでのプライバシ保護機能を提供することができる．しかし，単純にデータを暗号化して送信した場合，他のユーザとのデータ共

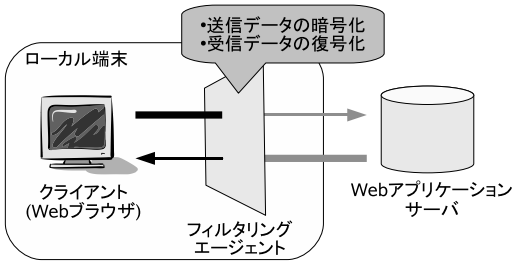


図 1　フィルタリング・エージェントシステム
Fig. 1　Filtering agent system.

有が行えない．そのため，本論文では，放送暗号とクラスタリングを用いた暗号化データ共有機構の構築方法を提案する．

本論文の構成は，2 章で関連研究について，3 章で本研究で利用する概念と提案するフレームワークの概要について述べる．4 章では，既存の手法を我々の目的に応用することを試み，既存手法の問題点について説明する．その後，5 章で我々が提案する，放送暗号とクラスタリングを用いた手法について説明し，6 章においてその効果をシミュレーションによって測定する．最後に 7 章で結論を述べる．

2. 関 連 研 究

リモートデータベース環境において，サービスプロバイダによるデータ閲覧を防ぐための研究としては，主に次の 2 種類が行われている．

1 つ目は，サーバへ保存された暗号化データの検索方法についての研究[1)–3)] である．これらでは，関係データベースに保存される各タプルを暗号化しセキュアな索引を付加することで，暗号化したままの状態で問合せ処理を実現している．同様に，文献 4) は XML データベースに対しても暗号化したまま問合せ処理を実現する方法について論じている．これらの暗号化データに対する問合せ手法は，我々の提案とは独立に機能させることが可能である．本研究では，暗号化データに対するアクセス権限の付加手法のみを論じているが，暗号化データの問合せにはこれらの手法を用いるものとする．

サービスプロバイダによるユーザデータ不正利用防止を目的とする研究の 2 つ目は，暗号化データの共有に必要な，暗号鍵やデータの配送に関する研究[5)–7)] である．これらでは，データ共有を行うグループ情報を公開しユーザ間に階層構造を設定することで，各ユーザが保持しなければならない鍵数の削減を行っている．しかし，データ共有グループ情報を公開してしまうと，アクセス頻度を解析することで，データの重要度やユーザの地位などが推定される可能性がある．我々の手法では，この点も考慮しデータ共有グループを公開せずにアクセス権限情報を付加している．また，ユーザ間に階層構造を設定する手法では，1 度設定した階層構造を変更することは容易ではない．そのため，データ共有グループが動的に生成される場合非効率である．それに対して我々の手法では，ユーザ間に特別な関係を導入しないため，動的に共有グループを生成する場合にも効果的に動作することができる．

3. アカウントの概念と提案フレームワーク

Web アプリケーションでは，いくつかのデータの集まりをユーザアカウントという概念を用いて管理している．ユーザアカウントには，識別子とパスワードによる認証が提供されており，Web アプリケーションを利用するうえでの大まかなデータ集合を構築することができる．本手法では，このユーザアカウントを本来 Web アプリケーションが想定している利用法ではなく，アクセス制御機能を提供するために用いる．すなわち，ユーザアカウントに設定されている識別子とパスワードはクライアント上で動作するユーザエージェントが管理し，ユーザには通知しない．そのため，ユーザはエージェントを介さずアカウントにアクセスすることはできず，提案手法で想定しない操作をアカウントに行うことはできないものとする．Web アプリケーションを利用するためのユーザ認証は，このエージェントに認証機能を用意することで行う．

図 2 は我々の提案するフレームワークの概要を表したものである．ユーザが新しいデータを追加しようとするとき，ユーザエージェントは次のように変換を行う．まず，追加データとアクセス権限を与えるユーザ情報がクライアントマシン上で動作する QueryProcessor に渡される．QueryProcessor は，アクセス権限を与えるユーザに関する情報を BroadcastEncryption Processor に渡す．BroadcastEncryption Processor は，ServiceProvider から各ユーザの公開鍵情報を取得しアクセス権限情報を生成する．次に，QueryProcessor は ServiceProvider からアカウント参照情報を取得し，次章で述べる方法で，データエントリを保存するアカウントを決定する．最後に，QueryProcess は文献 1) などで提案されているセキュアなインデックスとともに暗号化されたデータをサービスプロバイダに送信し保存する．問合せを行う場合，このセキュアインデックスを利用した問合せへと変換しサーバへ問い合わせる．結果に対し，秘密鍵を用いてアクセス権限情報の判定を行い，権限を持つデータのみクライアントへ返す．

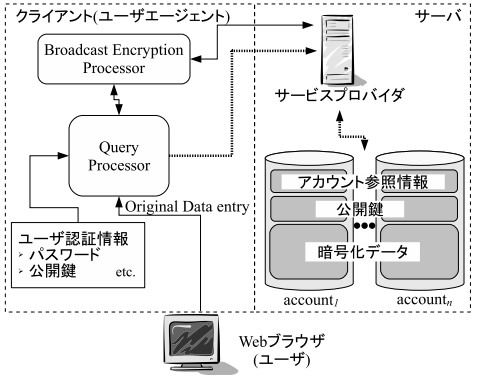


図 2　フレームワーク構造とデータの流れ
Fig. 2　Framework and dataflow.

4. 既存手法の応用

本章では，既存の手法を我々の問題に適用する 2 つのモデル，(1) シングルアカウントモデル，(2) マルチアカウントモデルについて述べ，その特徴と問題点について説明する．以降では，ユーザの総数を n とする．

4.1 シングルアカウントモデル

シングルアカウントモデルでは，**図 3** に示すように，すべてのユーザが 1 つのアカウントに対し読み書きを行う．1 つのアカウントを複数のユーザで共有利用する場合，他人のデータに対する改変および削除が問題となる．しかし，前述のようにアカウントに対する操作はユーザエージェントを通さずには行えない．したがって，不正なアクセスはシステムが取り除き，こうした問題は発生しないものとする．また，アカウントに読み書きするためのパスワードは各ユーザエージェントに前もって知らせておくものとする．

このモデルでアクセス制御を行うためには，データエントリ e の追加時に，次に示す手順で暗号化を行う．以下では，$user_1, user_2, \cdots, user_m$ にアクセス権限を与えるものとし，$user_i$ の公開鍵を pub_i と書く．

(1)　共通鍵 k_e で e を暗号化し $Enc(e, k_e)$ を得る．
(2)　アクセス権限を与えるユーザの公開鍵で共通鍵 k_e を個々に暗号化した，

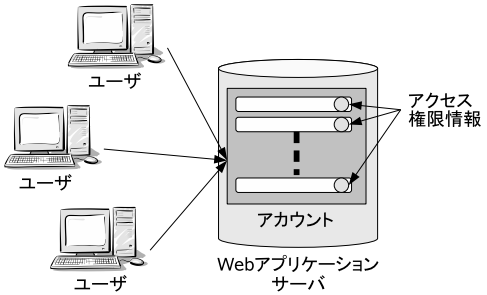


図 3　シングルアカウントモデル
Fig. 3　Single account model.

$$\bigoplus_{j=1}^{m} Enc(k_e, pub_j) \tag{1}$$

を付加する．ただし，演算子 $\oplus$ は文字列の連結を表すものとする．以上の手順により得られた，

$$Enc(e, k_e) \oplus \left(\bigoplus_{j=1}^{m} Enc(k_e, pub_j) \right) \tag{2}$$

をアカウントへ追加する．データエントリを利用する場合，エージェントはユーザの秘密鍵を用い，式 (1) に対して順に復号を試みる．復号化でき共通鍵 k_e を得られればアクセス権限が与えられていると分かり，データエントリ e を手に入れることができる．

この手法では，データエントリごとに細かい権限設定が可能であるが，総エントリ数 N_e に対し $O(n \cdot N_e)$ の膨大なデータが必要になる．また，アクセス権限があるか否かは復号を試みるまで分からない．そのため，多くの不要のデータエントリに対し復号を試みる必要がある．一般に復号にはコストがかかるため，アクセス権限を持たない多くのデータに対し復号を試みることは好ましくない．

4.2 マルチアカウントモデル

マルチアカウントモデルでは**図 4** に示すように，いくつかのアカウントを利用しアカウントごとに読み書きできるユーザ集合 S_i を設定する．この場合も先ほどと同様に 1 つのアカウントを複数人で共有するが，不正操作は行えないものとし，パスワードも各エージェントに前もって知らせるものとする．このモデルでアクセス制御を行う方法は次のようになる．以降ではアカウント集合を A で表し，その総数を N_A で表す．

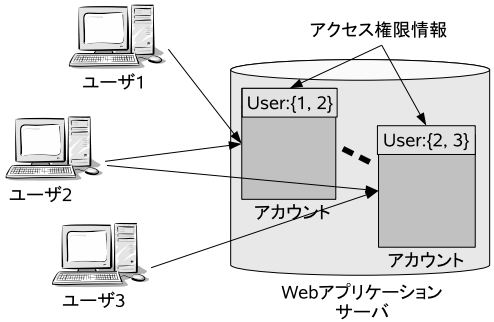


図 4　マルチアカウントモデル
Fig. 4　Multi account model.

4.2.1　データエントリの追加と取得

マルチアカウントモデルを利用する準備として，次の手順でアカウントごとにアクセス権限情報を設定する．

(1)　各アカウント a_i に共通鍵 k_{a_i} を設定する．
(2)　共通鍵 k_{a_i} をアクセス権限を与えるユーザの公開鍵で個々に暗号化したものを a_i へ追加する．

たとえば，$user_1, user_2, \cdots, user_m$ にアカウント a_i へのアクセス権限を与えるとし，4.1 節と同様に $user_i$ の公開鍵を pub_i と書くと，

$$\bigoplus_{j=1}^{m} Enc(k_{a_i}, pub_j) \tag{3}$$

というデータを作成し，アカウントへ追加する．

$user_l$ がアカウント a_i に新たなデータエントリ e を追加する場合，次の手順で行う．

(1)　アカウント a_l から式 (3) で表される権限情報を取得し復号を試みる．
(2)　$user_l$ がアクセスが許可されている場合 k_{a_i} を取得できる．
(3)　追加するデータ e を k_{a_i} で暗号化し $Enc(e, k_{a_i})$ を a_i に追加する．

データエントリの取得も追加と同様に，まずアカウントの共通鍵を得，それを用いて復号化する．逆にアクセスが許可されていない場合，アカウント用の鍵が手に入らないため，不正にアクセスされることはない．

4.2.2　データ数の評価

マルチアカウントモデルでは，サーバへ登録すべきアクセス権限情報は $O(n \cdot N_A)$ となる．$N_A \ll N_e$ を仮定すると，大幅にアクセス権限情報を減らすことができたといえる．しかし，アクセス権限をアカウント単位でしか設定できないため，細かい権限設定を行うとアカウント数が膨大になってしまう．たとえば，3 つのデータエントリ e_1，e_2，e_3 と 2 人のユーザ $user_1$，$user_2$ について，以下のように権限を設定することを考える．

e_1：　$user_1$ のみアクセス可能
e_2：　$user_1$，$user_2$ ともにアクセス可能
e_3：　$user_2$ のみアクセス可能

この場合，3 つのアカウント a_1，a_2，a_3 を作成し，

a_1：　$user_1$ のみアクセス可能
a_2：　$user_2$ のみアクセス可能
a_3：　$user_1$，$user_2$ ともにアクセス可能

と設定する．そして a_1 に $\{e_1\}$，a_2 に $\{e_3\}$ を a_3 に $\{e_2\}$ を登録することになる．このように，マルチアカウントモデルではアクセス権限設定の組合せに応じてアカウントの数が必要となり，最悪 2^n 個のアカウントが必要となる．

アクセス権限の有無を調べるための復号回数は，マルチアカウントモデルでは，アカウントごとにアクセス権限の有無を調べるだけでよく，個々のデータエントリに対し復号を試みる必要はない．したがってアカウント数が少ない状況では，シングルアカウントモデルに比べ復号を試みる回数は少なく効率的である．

5.　放送暗号とクラスタリングの利用

本章では，前章で紹介した 2 種類のアクセス制御機構の問題点について，ペアリングを用いた放送暗号とクラスタリングによる解決策について説明する．

5.1　放送暗号による権限情報固定長化

シングルアカウントモデルの問題点の 1 つは，アクセス権限情報に相当する共通鍵を暗号化したデータを数多く保管しなければならないことである．この問題の解決策として，ペアリングを用いた放送暗号により権限情報の固定長化を行う．

ペアリングとは，ある種の楕円関数上で定義される 2 入力 1 出力の関数 f であり，次のように定義される．

$$f : G \times G \to G_T (G：加法群，G_T：乗法群) \tag{4}$$

また，以下で表される双線形性を持つ[8]．

$$f(P_1+P_2,Q_1)=f(P_1,Q_1)\cdot f(P_2,Q_1) \tag{5}$$
$$f(P_1,Q_1+Q_2)=f(P_1,Q_1)\cdot f(P_1,Q_2) \tag{6}$$

これらにより，次の式を導くことができる．

$$f(\alpha P_1,\beta Q_1)=f(\beta P_1,\alpha Q_1) \tag{7}$$

ただし，P_1，P_2，Q_1，$Q_2 \in G$，α，$\beta \in Z$ である．

このペアリングを公開鍵暗号に利用する研究は，近年さかんに行われている[9),10)]．ペアリングを用いた放送暗号もその 1 つであり，アクセス権限を持つユーザ数にかかわらず，固定長でデータエントリの共通鍵を暗号化することができる[11)]．以下に，その手順を記す．

5.1.1 準　　備

まず，$P \in G$ なる P と，ランダムな整数 α，γ を生成する．次に，

$$P_i=\alpha^i P\ (1\le i\le 2n) \tag{8}$$
$$Q=\gamma P \tag{9}$$

により $P_1,P_2,\cdots,P_{2n}$ および Q を求める．$user_i$ の秘密鍵は $D_i=\gamma P_i$ である．また，全ユーザに P，Q，P_i（$1\le i\le 2n$，$i\ne n+1$）を公開しておく．

5.1.2 暗　号　化

データエントリ e を追加する場合，乱数 τ を生成し，

$$K=f(P_{n+1},P)^{\tau} \tag{10}$$

で計算される鍵 K によりデータエントリ e を暗号化し $Enc(e,K)$ を得る．また，アクセス権限付加のために，

$$C_0=\tau P,\ C_1=\tau\left(Q+\sum_{j\in S}P_{n+1-j}\right) \tag{11}$$

で C_0，C_1 を計算する．ここで，S はアクセス権限を付加するユーザ番号の集合である．これらを用いて，最終的にサーバに記録されるデータは，$Enc(e,K)$ と (C_0,C_1) となる．また，アクセス権限情報として付加される (C_0,C_1) のビット長はユーザ数 n によらず一定となっている．

5.1.3 復　号　化

$user_i$ がデータエントリを復号化する場合，(C_0,C_1) を用いて，

$$K=\frac{f(P_i,C_1)}{f(D_i+\sum_{j\in S,j\ne i}P_{n+1-j+i},C_0)} \tag{12}$$

により共通鍵 K を求め $Enc(e,K)$ を復号化し，e を得ることができる．

安全性に関しては，l−BHDE 問題の困難性を仮定すれば，高い安全性を持っていることが証明できる[11)]．

5.2 クラスタリングによるアクセス数の削減

放送暗号を用いることで，アクセス権限情報に相当する鍵情報列を単一の鍵に置き換えることができる．そのため，権限の有無を調べるために必要な復号回数はつねに 1 回である．しかし，ユーザがアクセス権限を持っているか否かは復号を試みるまで分からず，権限を持っていないデータを復号することによる復号コスト問題は残ったままである．そこで，各アカウントに含まれるデータの権限分布を推定し，ユーザが権限を持たないデータを復号しなくて済むように，データをアカウントに割り当て配置する．

5.2.1 クラスタリングに利用できる情報

本研究の目的は，サーバにデータが閲覧されないようにしながら，ユーザ同士ではデータの共有を行うことである．そのため，繰り返し述べているように，各データにアクセス権限を持つか否かは権限情報を復号するまで分からず，各データが誰と共有しているのかについても復号するまでは分からない．なぜなら，あるデータについて，共有ユーザ情報を平文として付加した場合，どのユーザ同士がデータ共有を行っているのか分かり，ユーザの関係を調べることができてしまうからである．したがって，アカウント割当てには，各データの共有関係に関する情報を利用することはできない．しかし，データを追加するユーザは，自分がアクセス権限を与えるユーザについては知っている．したがって，アカウント割当てはデータを追加するタイミングで行う．

また，各ユーザごとにアクセス権限のあるデータが格納されているアカウントのリストを用意する．このリストをアカウント参照情報と呼ぶことにする．たとえば，$user_1$ のアカウント参照情報がアカウント a_1，a_2 であれば，$user_1$ がアクセス権限を持つデータはこの 2 つのアカウントのみに保存されていることを表し，他のアカウントからデータを取得する必要はない．このようにすることで，各ユーザがアクセス権限の有無を調べる必要のあるデータ数を減らすことができる．また，どのユーザも他のユーザのアカウント参照情報を読み書きすることができるものとする．クラスタリングには，このアカウント参照情報も利用する．

5.2.2 保存先アカウントの決定

アカウント参照情報から各アカウントに含まれるデータの偏りを次のように推定する．まず，あるアカウントを参照しているユーザが n 人いる場合，対象のアカウントには少なくとも，この n 人がアクセス権限を持つデータが保存されている．したがって n が大きい場合，すでに多くのデータがアカウントに保存されていると考えられる．逆に，あるアカウン

トを参照しているユーザが少ない場合，アカウントに保存されているデータ数は少ないと考えられる．つまり，アカウント内データ数がアカウント参照人数に従属すると仮定する．次に，あるユーザが参照するアカウントが増えるとする．このときユーザは，そのアカウントに含まれているデータのすべてにアクセス権限を持たないといえる．したがって，アカウントに含まれているデータ数とそのアカウントを新たに参照するユーザの積が，そのアカウントへデータを追加した場合に発生するコストである．そこで，アカウント a に対してアカウント a をすでに参照しているユーザ数 n_a^+ と，S に含まれるユーザのうち，まだ参照していないユーザ数 n_a^{S-} をとすると．アカウント a のユーザ集合 S におけるコスト $Cost_S(a)$ を以下で定める．

$$Cost_S(a) = (n_a^+)^k \cdot n_a^{S-} \ (k : const) \tag{13}$$

データを追加するアカウントを選択する場合，上記のコストが最も小さなアカウントにデータを保存する．

5.3　提案手法のまとめ

本節では，前節までに紹介した放送暗号とクラスタリングを用いた，提案手法について説明する．

5.3.1　準　　備

まず，ペアリングを用いた放送暗号を利用するための準備を行う．すなわち，式 (8)，(9) により公開情報 P，Q，P_i（$1 \leq i \leq 2n$，$i \neq n+1$）を計算し，全ユーザ $u \in U$ のエージェントに通知する．また，$user_i$ のエージェントに秘密鍵 D_i を通知する．次に，n 個のアカウント a_i（$1 \leq i \leq n$）を取得し，識別子とパスワードを全ユーザのエージェントに通知する．

5.3.2　データエントリの追加

グループ S で共有するデータエントリ e を追加する場合，エージェントは以下のようにデータを変換する．

まず，乱数 τ を生成し，式 (11) によりアクセス権限情報 (C_0, C_1) を求める．次に，本来のデータを暗号化する鍵 K を生成する．しかし，P_{n+1} は公開されていないため，式 (10) ではなく，以下の式で計算する．

$$K = f(P_{n+1}, P)^\tau = f(P_n, P_1)^\tau \tag{14}$$

これが正しいことは，式 (7) により証明できる．

$$\begin{aligned} f(P_{n+1}, P) &= f(\alpha \cdot \alpha^n P, 1 \cdot P) \\ &= f(1 \cdot \alpha^n P, \alpha P) \\ &= f(P_n, P_1) \end{aligned} \tag{15}$$

作成された鍵 K により e を暗号化し，得られた

$$Enc(e, K) \oplus (C_0, C_1) \tag{16}$$

を 5.2.2 項の手法で選ばれたアカウントへ保存する．

5.3.3　データエントリの取得と更新

$user_l \in S$ がデータを取得する場合，参照リストに登録されているアカウントすべてに問合せを行い，目的のデータ (16) を取得する．e に対しアクセス権限が与えられている場合，式 (12) により秘密鍵 D_i を用いて共通鍵 K を得ることができる．鍵 K が得られれば，$Enc(e, K)$ を復号化し e を得ることができる．

データエントリを更新する場合，更新後もアクセス権限を持つユーザ集合が同じであるなら，元々使用されていた共通鍵 K を用いて暗号化しサーバへ送信すればよい．更新によってアクセス権限を持つユーザ集合に変更が生じる場合，まず対象のデータエントリを削除する．その後，新しくアクセス権限を付加するユーザ集合 S に対して，上記データの追加手順を行う．

6.　シミュレーション実験

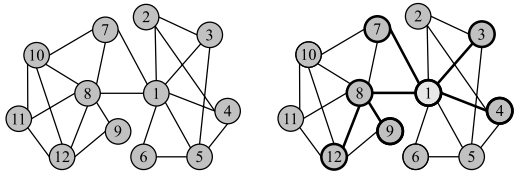
提案手法の有効性を確かめるために，シミュレーションによる評価実験を行った．本章では，まずシミュレーションに使用したユーザ関係のモデル化について説明し，実験方法とその結果について述べる．

6.1　交流関係グラフとデータ共有モデル

本シミュレーションでは，ユーザの交流関係に現実の交流関係を反映させるために，スケールフリー性を持つ交流関係グラフとして設定し，その交流関係をもとにデータの共有を行うものとした．この交流関係グラフでは，各頂点を各ユーザに対応させ枝で結ばれているユーザは交流関係があるものとする．

図 5 (a) は $user_1$ が $user_2, user_3, \cdots, user_8$ と交流関係にある場合を示している．この交流関係グラフを 5 頂点からなる完全グラフから始め，次数 3 の頂点を追加していく BA モデル[12] により作成した．次に，この交流関係グラフを用いて，データ共有を行うグループを次の手順により作成した．

(1)　交流関係グラフ G_n からまだ選ばれていない頂点を 1 つ選ぶ．



(a) 交流関係グラフ　　(b) データ共有グラフ

図 5　交流関係グラフとデータ共有モデル
Fig. 5　Friendship and data-sharing model.

(2)　選ばれた頂点についてグループのリーダになるか否かの判定を行う．頂点の次数に比例する確率でリーダとなる．

(3)　グループに参加した頂点は交流関係のある頂点に対しグループ参加を呼びかける．

ステップ (2) において，次数に比例する確率でリーダを選んでいるのは，多くのユーザと交流関係のあるユーザがグループの中心になることが多いと仮定しているからである．同様に，ステップ (4) において，次数に反比例する確率で，グループへの参加を決めているのは，多くのユーザと交流関係にあるユーザは，他のグループにも参加している可能性が高く，新しいグループへは参加しにくいと仮定しているからである．

図 5 (b) は，ステップ (1) で $user_1$ が選ばれ，リーダになった場合を表している．$user_1$ はステップ (3) で交流関係にある $user_2, user_3, \cdots, user_8$ に対し参加を呼びかける．このうち $user_3$，$user_4$，$user_7$，$user_8$ が参加している．さらに，参加した各ユーザは，交流関係にある他のユーザについても参加を呼びかける．図 5 (b) では，$user_9$，$user_{12}$ が $user_8$ からの参加を承諾した場合について記したものである．

6.2　実験と結果

シミュレーションは，ユーザ総数が 100 人，1,000 人，10,000 人の場合について行った．グループの作成手順は 3 回実行し，各ユーザが最大 3 回リーダになるようにした．次に，各グループに対してグループの構成メンバ数に比例するデータを共有させ，シングルアカウント法，マルチアカウント法および提案手法について，鍵情報列長の平均と，アクセス権限適合率を求めた．また，提案手法に関しては，各ユーザのアカウント参照数とアクセス権限適合率の分布についても求めた．ここで，アクセス権限適合率とは，復号を試みなければならないデータのうち，実際にアクセス権限を持っていたデータの数を表すものである．つまり，$user_i$ のアクセス権限適合率 $r(user_i)$ とは，$user_i$ が復号を試みる必要のあるデータ数を $Check(user_i)$，実際にアクセス権限を持っていたデータ数を $Auth(user_i)$ として次式で定義されるものである．

$$r(user_i) = \frac{Auth(user_i)}{Check(user_i)} \tag{17}$$

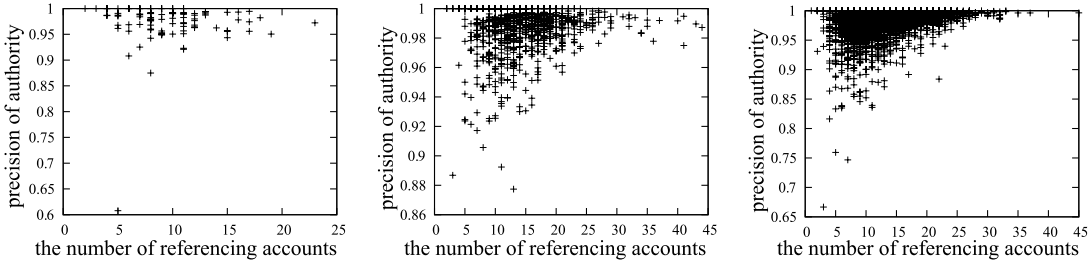
表 1　シミュレーション結果
Table 1　Result of simulation.

ユーザ数	実験番号	グループ数	シングルアカウント法		マルチアカウント法		提案手法	
			鍵長平均	適合率	鍵長平均	適合率	鍵長平均	適合率
100	1	112	19.0328	0.190328	11.4464	-	1	0.981911
100	2	107	16.8371	0.168371	12.0467	-	1	0.991164
100	3	55	13.6349	0.136349	11.4545	-	1	0.790000
1000	1	1034	27.2675	0.0272675	13.9923	-	1	0.987987
1000	2	1083	30.9286	0.0309286	12.5171	-	1	0.984673
1000	3	1056	31.0023	0.0310023	12.9659	-	1	0.984935
10000	1	10563	42.4664	0.00424664	12.2277	-	1	0.988050
10000	2	10432	44.4462	0.00444462	14.1411	-	1	0.989989
10000	3	10642	39.9114	0.00399114	13.5978	-	1	0.988845

表 1 はシミュレーションの結果である．適合率の計算は，各ユーザが取得可能なすべてのデータに対してアクセス権限の有無を調べた，最悪の場合について求めている．実際には，問合せ処理により対象データ数は絞られるため，より大きい値となる．また，マルチアカウント法では，各データに対してアクセス権限の有無を調べる必要がないため適合率は求めなかった．シミュレーションによると，シングルアカウント法では，鍵情報列が総ユーザ数とともに増加し，権限の有無を調べるために数多くの鍵を復号してみなければならないことが分かる．対して，提案手法では，どんな状況であっても 1 回の復号で権限の有無を調べることができている．また，提案手法における適合率は高い値を保っており，各ユーザとも権限の有無を調べたデータのほとんどに対して権限を保持しており，無駄な復号が少ないことが分かる．**図 6** では，横軸にアカウント参照数，縦軸にアクセス権限適合率をとり，各ユーザをこの平面の対応する点にプロットした．アカウントの参照数が多くなれば一般的に通信コストが増える．実験結果では，多くのユーザが少ない参照数を保っており好ましい結果となっている．

7. おわりに

本論文では，Web アプリケーションにおけるプライバシ問題を解決するために，サーバ



(a) 100 ユーザ　(b) 1,000 ユーザ　(c) 10,000 ユーザ

図 6　アクセス権限適合率の分布
Fig. 6　Distribution of users.

へ送信するデータを暗号化するフレームワークの提案を行った．本提案の特徴は，Web アプリケーションの持つコラボレーションの容易さという性質を損なわないように，放送暗号とクラスタリングを用い，サーバ上ではデータを暗号化したままグループ内でのデータ共有を可能している点にある．公開鍵暗号を用いた従来手法の応用では，共有データに関するアクセス権限情報やアカウント数の巨大化や，大きな復号コストが発生する．本提案では放送暗号の仕組みを応用し，少ない情報量でアクセス権限情報を付加し，ユーザの交流関係をもとにしたクラスタリングにより，復号コストを削減している．

我々の提案手法では，1 度アクセス権限情報を付加し暗号化すると，そのデータを共有しているメンバ情報などはアクセス権限を持つユーザが復号してみるまで分からない．このようにして，グループの構成メンバ情報などが漏洩することを防いでいる．しかし，グループメンバの増減に関しては，関連するデータすべての書き換えが必要である．これは，グループ情報を集中管理していない以上避けられない問題である．そのため Web アプリケーションの中には，本提案手法が効果的なものと，あまり効果的でないものがある．Web カレンダーやスケジュール管理ツールなど，1 度登録されたデータが変更されることが少なく，主にデータを追加し権限の更新が少ないアプリケーションに対して本提案手法は効果的である．逆に，頻繁にアクセス権限が更新されるアプリケーションに対しては，更新のたびにコストが生じるため，本提案手法の適用があまり効果的でないといえる．

謝辞　本研究の一部は，京都大学グローバル COE プログラム「知識循環社会のための情報学教育研究拠点」の援助を受けております．ここに記して謝意を表します．

参 考 文 献

1) Hacigümüş, H., Iyer, B., Li, C. and Mehrotra, S.: Executing SQL over encrypted data in the database-service-provider model, *SIGMOD '02: Proc. 2002 ACM SIGMOD international conference on Management of data*, New York, NY, USA, ACM, pp.216–227 (2002).
2) Bouganim, L. and Pucheral, P.: Chip-Secured Data Access: Confidential Data on Untrusted Servers, *VLDB'02: Proc. 28th international conference on Very Large Data Bases* (2002).
3) Hore, B., Mehrotra, S. and Tsudik, G.: A Privacy-Preserving Index for Range Queries, *VLDB'2004: Proc. 13th international conference on Very large data bases*, pp.720–730 (2004).
4) Wang, H. and Lakshmanan, L.V.S.: Efficient secure query evaluation over encrypted XML databases, *VLDB '06: Proc. 32nd international conference on Very large data bases*, VLDB Endowment, pp.127–138 (2006).
5) Damiani, E., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S. and Samarati, P.: Key Management for Multi-User Encrypted Databases, *StorageSS '05: Proc. 2005 ACM workshop on Storage security and survivability*, ACM, pp.74–83 (2005).
6) Damiani, E. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S. and Samarati, P.: Selective Data Encryption in Outsourced Dynamic Environments, *Electron. Notes Theor. Comput. Sci.*, Vol.168, pp.127–142 (2007).
7) Zou, X., Karandikar, Y. and Bertino, E.: A dynamic key management solution to access hierarchy, *Int. J. Netw. Manag.*, Vol.17, No.6, pp.437–450 (2007).

8) Washington, L.C.: *Elliptic Curves: Number Theory and Cryptography*, Discrete Mathematics and Its Applications, Vol.24, Chapman & Hall/CRC (2003).
9) 岡本栄司，岡本　健，金山直樹：ペアリングに関する最近の研究動向，*IEICE Fundamentals Review*, Vol.1, No.1, pp.51–60 (2007).
10) Dutta, R., Barua, R. and Sarkar, P.: Pairing-Based Cryptographic Protocols: A Survey (2004).
11) Boneh, D., Gentry, C. and Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys, *CRYPTO 2005*, *Lect. Notes Comput. Sci.*, Vol.3621, pp.258–275 (2005).
12) Barabási, A.-L. and Albert, R.: Emergence of scaling in random networks, *Science*, Vol.286, No.5439, pp.509–512 (1999).

(平成 19 年 12 月 20 日受付)
(平成 20 年 2 月 19 日採録)

（担当編集委員　有次 正義，DBWeb 2007 推薦論文）

川本　淳平

2007 年京都大学工学部情報学科卒業．現在，同大学大学院情報学研究科修士課程に在学中．プライバシ問題，特に Web におけるプライバシ問題に興味を持つ．日本データベース学会，人工知能学会各学生会員．

吉川　正俊（正会員）

1985 年京都大学大学院工学研究科博士後期課程修了．工学博士．京都産業大学，奈良先端科学技術大学院大学，名古屋大学を経て，2006 年より京都大学大学院情報学研究科教授．この間，南カリフォルニア大学客員研究員，ウォータルー大学客員准教授．The VLDB Journal および Information Systems（Elsevier/Pergamon）の編集委員．XML データベース，多次元空間索引，異種情報源の統合等の研究に従事．電子情報通信学会，ACM，IEEE Computer Society 各会員．