

# 複数データベース間における機密性を保持した情報共有の一手法

隅 崇佳<sup>†</sup> 村本 俊祐<sup>‡</sup> 上土井 陽子<sup>‡</sup> 若林 真一<sup>‡</sup>

<sup>†</sup>広島市立大学 情報科学部

<sup>‡</sup>広島市立大学大学院 情報科学研究科

〒731-3194 広島市安佐南区大塚東三丁目 4-1

E-mail: <sup>†</sup> t.sumi@lcl.ce.hiroshima-cu.ac.jp, <sup>‡</sup> {yoko, wakaba}@ce.hiroshima-cu.ac.jp

**あらまし** 近年、膨大な情報を扱うようになり様々な情報がデータベース化されている。複数の非公開データベース間で情報を共有する際にデータベースの機密性を確保することが重要になる。例えば複数の競合する企業などが持つデータベースの情報を互いに共有することで多くの情報を得て、さらに新たな知識を得られる。ここで企業は自分の持つ情報に関連する情報が相手にあれば共有したいが、関連する情報は相手によって異なる。また企業は相手が持たない情報は明らかにしたくない。関連する情報のみを共有するにはデータベース全てを公開することはできない。この問題に対処するため R.Agrawal らは複数のデータベース間において機密性を保持しつつ情報共有を行うプロトコルを提案している。本研究ではそのプロトコルで確保されている機密性をさらに向上させるプロトコルを開発する。

**キーワード** データベース, 情報共有, 機密性

## A Method of Secure Information Sharing Across Private Databases

Takayoshi SUMI<sup>†</sup> Shunsuke MURAMOTO<sup>‡</sup> Yoko KAMIDOI<sup>‡</sup> and Shin'ichi WAKABAYASHI<sup>‡</sup>

<sup>†</sup> Faculty of Information Sciences, Hiroshima City University

<sup>‡</sup> Graduate School of Information Sciences, Hiroshima City University

3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima, 731-3194 Japan

E-mail: <sup>†</sup> t.sumi@lcl.ce.hiroshima-cu.ac.jp, <sup>‡</sup> {yoko, wakaba}@ce.hiroshima-cu.ac.jp

**Abstract** In recent years, more attention is focused on sharing information in a distributed system consisting of peers, each of which has a private database. It is important to protect private databases when multiple parties share information on the databases. For example, if two companies which compete each other share information on private databases, they could cooperate in a certain area related with the others. When competing companies share information, they would like to reveal only information related with well-matched data. But it is important to differ from well-matched data with competitor. Also competing companies would not like to share private data. To share the only related information, companies cannot open all data in private databases. To solve this issue, Agrawal proposed a protocol for sharing information across private databases while protecting private data. In this paper, we propose a protocol that executes more secure information sharing than Agrawal's one.

**Keyword** database, information sharing, secret

### 1. はじめに

複数の機関が各々のデータベース上のデータを情報共有する際、それぞれのデータベースのデータが他の機関にすべて明らかになることがこれまでは仮定されていた。当然データベースが大規模になれば、多くの情報が明らかにされることになる。これは必要以上の情報を明らかにしたくない非公開データベースにおいて好ましくない。そこで機密性の観点から質問と無関係な情報が明らかにならないような方法で非公開デ

ータベースを共有する要求が高まってきた。その方法として信頼された第三者機関が情報共有を仲介する方法がある。機関 S と R が情報共有を行う場合に、信頼された第三者が情報共有を行う機関 S と R のデータベースから共有できる情報を決定する。共有できるとした情報を両方の機関に知らせて機関 S と R の情報共有を行う。ここで第三者機関は情報共有を行う機関全てに完全に信頼されていなければならない。しかし第三者を完全に信頼することは難しい。

近年、信頼された第三者を想定せずに、二者間での通信のみで共有する情報を決定する手法が文献[1]で提案されている。文献[1]では情報共有する機関各々のデータベース上のデータを暗号化することで必要以上の情報を明らかにせずとも共有できる情報を決定している。文献[1]のプロトコルでは暗号化関数の特性として *indistinguishability* 特性が仮定されており、この特性が必要以上の情報を明らかにしないために不可欠な特性となっている。

本稿では文献[1]で定義されている暗号化関数に仮定された *indistinguishability* 特性を緩めても同様に機密性を保ち、情報共有を行うことのできるプロトコルを開発する。

## 2. 極小情報共有

### 2.1 セキュリティモデル

二者間の通信のみで共有する情報を決定する場合において情報共有を行う機関として以下のようなセキュリティモデルを仮定する。

#### *honest-but-curious* な機関

プロトコルに参加する機関は忠実にプロトコルに従う。しかし、プロトコル実行中に受け取ったメッセージや行った計算を全て記録してのちに付加情報を得ることを目的として記録を解析するかもしれない。

従来手法[1]においても本稿で提案する手法においてもプロトコルに参加する機関に上記を仮定する。

### 2.2 問題の定式化

本稿で考察する問題を以下に示す。

#### 情報共有(理想)

それぞれ  $D_r$  と  $D_s$  というデータベースを持つ2つの機関  $R$  と  $S$  がある。データベース  $D_r$  と  $D_s$  に関連する質問  $Q$  があたえられると、質問  $Q$  の答えを計算し、互いの機関のどんな付加情報も明らかにせずとも機関  $S$  は  $R$  に応答を返す。

#### 情報共有(極小共有)

それぞれ  $D_r$  と  $D_s$  というデータベースを持つ2つの機関  $R$  と  $S$  がある。データベース  $D_r$  と  $D_s$  に関連する質問  $Q$  とある情報の種類  $I$  が与えられると、機関  $S$  は質問  $Q$  の答えを計算し、 $I$  に含まれる種類の情報を除いては互いの機関でどんな付加情報も明らかにせずとも応答を機関  $R$  に返す。

### 2.3 質問の種類

文献[1]では *intersection*, *equijoin*, *intersection size*, *equijoin size* の4つの質問に焦点を当てている。本稿ではその中の *intersection* に焦点を当てる。

機関  $S$  はデータベーステーブル  $T_s$ ,  $R$  は  $T_r$  を持つ機関とする。両方のテーブルには特定の属性  $A$  が含ま

れている。ここで属性  $A$  は与えられた有限集合  $V$  の値からなる。また  $T_s.A$  を  $T_s$  に存在する属性  $A$  の値の集合、 $T_r.A$  を  $T_r$  に存在する属性  $A$  の値の集合とする。*intersection* プロトコルでは  $T_s.A \cap T_r.A$  を質問の答えとして導出する。

## 3. 従来手法

この節では極小情報共有問題に対する文献[1]のプロトコルを紹介する。はじめに従来手法に用いられている可換性のある暗号化関数の定義とハッシュ関数を用いる理由について述べ、次にそれらを用いる *intersection* プロトコルについて述べる。

### 3.1 可換性のある暗号

可換性のある暗号は  $f(g(v))=g(f(v))$  となるような暗号関数  $f$  や  $g$  の組である。つまり  $v$  を暗号化するために暗号化関数  $f$  と  $g$  の2つを用いることで機関  $R$  が  $S$  の助けなしに値  $v$  を暗号化した値を求めることができないうことを保証している。さらに機関  $R$  と  $S$  の関数を両方用いて値  $v$  を暗号化するとき、暗号化する順序が違って同じ暗号化した値を得ることができる。

#### 定義 1(*indistinguishability*)

$\Omega_k \subseteq \{0,1\}^k$  を  $k$  ビットの有限領域とする。また  $D_1$  と  $D_2$  を  $\Omega_k$  上の分布とする。 $A_k(x)$  を  $x \in \Omega$  が与えられ *true* か *false* を返すアルゴリズムとする。 $D_1$  と  $D_2$  は”計算的に区別がつかない”ということを用いて定義する。以下の式において  $A_k(x)$  は多項式時間内に真か偽を出力するアルゴリズムとし、どんな多項式  $p(k)$  に対しても以下の式は成り立つと定義する。

$$|\Pr[A_k(x)|x \sim D_1] - \Pr[A_k(x)|x \sim D_2]| < 1/p(k)$$

$\Pr[A_k(x)]$  はアルゴリズムが入力に対して真を返す確率であり、 $x \sim D$  は分布  $D$  上のある要素  $x$  ということの意味している。□

#### 定義 2(文献[1]での可換性のある暗号の定義)

可換性のある暗号化関数を  $f$  とし、鍵  $e$  を用いて情報  $v$  を暗号化することを  $f_e(v) \equiv f(e,v)$  と表す。また鍵の領域を  $KeyF$ , 暗号化した値の領域を  $DomF$  とする。 $a \in_r b$  は “ $b$  からランダムに選ばれた  $a$ ” ということの意味する。

従来手法に仮定されている暗号化特性を以下に示す。

1. 可換性：全ての  $e, e' \in KeyF$  に対して  $f_e \circ f_{e'} = f_{e'} \circ f_e$  が成り立つ。
2. 各  $f_e$  に対して暗号化前と後の関係は全単射
3. 鍵  $e$  を用いて暗号化された情報は鍵  $e$  が与えられると多項式時間内に復号が可能
4. *indistinguishability* :  $\langle x, f_e(x), y, f_e(y) \rangle$  の分布は  $\langle x, f_e(x), y, z \rangle$  の分布と区別がつかない。  
( $x, y, z \in_r DomF, e \in_r KeyF$ ) □

特性 4 はある暗号化前後のペアが与えられたとき、次に与えられるペアが同じ鍵で暗号化されたものかどうかを多項式時間内に判定できるアルゴリズムがないということを意味している。特性 4 を可換性のある暗号化関数の特性として用いることで文献[1]では intersection プロトコルにおいてお互いに相手に伝えることを許した情報以外の情報が漏れていないということを証明している。

### 3.2 ハッシュ関数

暗号化関数のみを用い、intersection を実現しようとした場合、似たような情報を暗号化関数で暗号化したものに偏りが出でくる可能性がある。例えば似たような情報 A と A' があり、それぞれを暗号化関数で暗号化したときに似たような暗号になると情報 A' が共通演算により明らかになったときにその A' から似たような情報 A を予想できるかもしれない。これでは無関係な情報を明らかにせず情報共有できたとはいえない。そのためにハッシュ関数を用いて全ての値がランダムに見えるようにする。

また暗号化関数を用いず、ハッシュ関数のみを用い共通の情報を導き出そうとする場合を考える。機関 S と R がそれぞれの機関が持つデータベース上のデータ(データの集合をそれぞれ  $V_s$  と  $V_r$  とする)をハッシュ化する。機関 S が集合  $h(V_s)$  を機関 R に送り、機関 R が送られてきた集合  $h(V_s)$  と自分が持つ集合  $h(V_r)$  を比較する。2つの集合  $h(V_s)$  と  $h(V_r)$  の両方に存在するハッシュ値の集合  $h(V_s \cap V_r)$  から機関 R は集合  $V_s \cap V_r$  を得ることができる。このとき機関 R は R のテーブルにないものをハッシュ化してそれを機関 S から送られたハッシュ値と比較し同じものを見つけることができれば、機関 S の持っている情報を知ることが可能になる。ここで互いの機関が持つデータテーブルに含まれる特定の属性 A は与えられた有限集合 V の値からなる。つまりハッシュ値の取りうる値は有限集合 V の取りうる値の数しかない。このことから自分のテーブルにないものをハッシュ化してそれを相手から送られたハッシュ値と比較し同じものを見つけることができるといえる。よってハッシュ関数のみの暗号化では不十分である。

### 3.3 Intersection プロトコル

文献[1]で紹介されている intersection プロトコルを以下に示す。

1. 機関 S と R はそれぞれが持つ集合  $V_s$ ,  $V_r$  をハッシュ関数  $h$  により変換する。変換した集合をそれぞれ  $X_s=h(V_s)$ ,  $X_r=h(V_r)$  とする。機関 S と R は領域  $KeyF$  からランダムに鍵  $e_s$ ,  $e_r$  を選ぶ。
2. 機関 S と R はハッシュ関数により変換された集合を選んだ鍵で暗号化する。暗号化後の集合を

それぞれ  $Y_s=f_{e_s}(X_s)$ ,  $Y_r=f_{e_r}(X_r)$  とする。

3. 機関 R は集合  $Y_r$  の要素を辞書式順に並べ換えた列を機関 S に送る。
4. (a)機関 S は集合  $Y_s$  の要素を辞書式順に並び換えた列を機関 R に送る。  
(b)機関 S は集合  $Y_r$  に含まれるすべての要素  $y$  を鍵  $e_s$  で暗号化する。暗号化後の集合を  $Z_r=f_{e_s}(Y_r)$  とする。それから機関 R に集合  $Y_r$  に含まれる全ての要素  $y$  に関しペア  $\langle y, f_{e_s}(y) \rangle$  を送り返す。
5. 機関 R はステップ 4 の(a)で機関 S から得た集合  $Y_s$  を鍵  $e_r$  で暗号化し集合  $Z_s=f_{e_r}(Y_s)$  を作成する。また集合  $V_r$  に含まれる要素  $v$  に対してステップ 4(b)で得たペア  $\langle y, f_{e_s}(y) \rangle = \langle f_{e_r}(h(v)), f_{e_s}(f_{e_r}(h(v))) \rangle$  からペア  $\langle v, f_{e_s}(f_{e_r}(h(v))) \rangle$  を得る。
6. 機関 R はステップ 4 で送られてきた暗号化集合  $Z_r$  とステップ 5 で作成した暗号化集合  $Z_s$  から集合  $Z_s \cap Z_r$  を求める。求めた集合  $Z_s \cap Z_r = f_{e_s}(f_{e_r}(h(V_s \cap V_r)))$  に属する暗号文とペアになっている集合  $V_r$  に属する要素の集合が  $V_s \cap V_r$  となる。

## 4. 従来プロトコルの機密性

この節では従来プロトコルにおいて用いられる可換性のある暗号化関数に仮定されている特性 4 がどのようにプロトコルの機密性に関係しているのかについて文献[1]の内容を引用し示す。また、我々は新たな視点から文献[1]のプロトコルにおいて用いる暗号化関数に特性 4 がなければどのような情報が漏れてしまうのかについて 5 節で考察する。

従来プロトコルにおける機密性の証明では機関 S がサイズ  $|V_r|$ 、機関 R がサイズ  $|V_s|$  と集合  $V_s \cap V_r$  を知る以外に他の情報が機関 S と R にもたらされていないことをプロトコルが満たしているかどうかを検討している。この性質は一般にゼロ知識性[3]と呼ばれる性質に近い。ゼロ知識性を満たすプロトコルでは参加者が相手に与えることを許している情報以外の知識が漏れていない。つまり、ゼロ知識性を満たすプロトコルにおいては全ての参加者は通信を行う前と後で自分の持つ知識を使ってできることが変わらないと言える。これは第三者が通信を見たときにある参加者があたかも相手と相互に正しく通信したかのように見えるやりとりを一人の参加者独自で作成するシミュレーションを行うことが可能であると言い換えることができる。

### 4.1 機密性の証明

ゼロ知識性の証明を軸にしてプロトコルの機密性を証明する。プロトコルでは機関 S がサイズ  $|V_r|$ 、機関 R がサイズ  $|V_s|$  と集合  $V_s \cap V_r$  を知ることを許している。よって、これらの知識をお互いにあらかじめ与え

られていると仮定する．その上で機関  $S$  と  $R$  の両方がシミュレート可能であれば，本物のやり取りと偽物のやり取りが区別できない．つまり機関  $S$  と  $R$  に知ることを許した知識以外においてゼロ知識性を満たすと言える．

#### 4.1.1 機関 $S$ によるシミュレーション

機関  $S$  がプロトコルを通して相手から得る情報はプロトコルのステップ 3 での集合  $Y_r$  のみである．本物では

$$Y_r = \{ f_{er}(x_1), f_{er}(x_2), \dots, f_{er}(x_m) \} \quad (m = |V_r|)$$

という情報であるが，シミュレーションにおいて機関  $S$  は領域  $DomF$  からランダムに  $|V_r|$  個の値 ( $z_i$ ) を発生させ，発生させた値を辞書式順に並び換えて集合  $Y_r$  を偽造する．偽造した集合を  $Y'_r$  とする．

$$Y'_r = \{ z_1, z_2, \dots, z_m \}$$

ここで特性 4 は暗号化前後のペアが与えられたとき，次に与えられるペアが同じ鍵で暗号化されたものかどうかを多項式時間内に判定できるアルゴリズムがないということを意味している．特性 4 を文献[1]の複数の補題より拡張したとき，以下の 2 つの  $2 \times m$  タプルは第三者が見ても計算的に区別がつかないと言える．

$$\begin{pmatrix} x_1 & x_2 & \dots & x_m \\ f_{er}(x_1) & f_{er}(x_2) & \dots & f_{er}(x_m) \end{pmatrix}$$

$$\begin{pmatrix} x_1 & x_2 & \dots & x_m \\ z_1 & z_2 & \dots & z_m \end{pmatrix}$$

ここで第三者には暗号化前の値 ( $x_i$ ) がわからないのでアルゴリズムに入力できない．つまり，本物の情報  $f_{er}(x_1), f_{er}(x_2), \dots, f_{er}(x_m)$  と偽物の情報  $z_1, z_2, \dots, z_m$  とを第三者は区別できない．よって，シミュレート可能と言える．

#### 4.1.2 機関 $R$ によるシミュレーション

機関  $R$  がプロトコルを通して相手から得る情報はプロトコルのステップ 4(a) で得る暗号化集合  $f_{es}(X_s)$  とステップ 4(b) で得るペアの集合  $\langle f_{er}(X_r), f_{es}(f_{er}(X_r)) \rangle$  の 2 つである．

まず，集合  $f_{es}(X_s)$  の偽造を考える．機関  $R$  はサイズ  $|V_s|$  と集合  $V_s \cap V_r$  をあらかじめ与えられていると仮定するので， $|V_r - V_s|$  を知ることもできる．さらに機関  $R$  は偽造するための鍵  $e'_s$  をランダムに選び，その鍵を用いて集合  $V_r \cap V_s$  の部分はハッシュ値の集合  $h(V_r \cap V_s)$  を暗号化することで偽造する．集合  $V_r - V_s$  の部分

は機関  $S$  のシミュレーションと同様に領域  $DomF$  からランダムに  $|V_r - V_s|$  個の値を選び偽造する．以上のことから本物では

$$Y_s = \{ f_{es}(x_1), f_{es}(x_2), \dots, f_{es}(x_p) \} \quad (p = |V_r|)$$

という形の集合  $Y_s$  を機関  $R$  は以下のように偽造する．

$$Y'_s = \{ f_{e'_s}(x_1), \dots, f_{e'_s}(x_q), z_{q+1}, \dots, z_p \} \quad (q = |V_r \cap V_s|)$$

暗号化関数が特性 4 を満たすという仮定とそれに関する文献[1]の補題より以下の 2 つのタプルは第三者が見ても区別がつかない．

$$\begin{pmatrix} x_1 & x_2 & \dots & x_q & x_{q+1} & \dots & x_p \\ f_{es}(x_1) & f_{es}(x_2) & \dots & f_{es}(x_q) & f_{es}(x_{q+1}) & \dots & f_{es}(x_p) \end{pmatrix}$$

$$\begin{pmatrix} x_1 & x_2 & \dots & x_q & x_{q+1} & \dots & x_p \\ f_{e'_s}(x_1) & f_{e'_s}(x_2) & \dots & f_{e'_s}(x_q) & z_{q+1} & \dots & z_p \end{pmatrix}$$

第三者は  $x_{q+1} \sim x_p$  の値を知らないが知ったとしても上記の 2 つのタプルは区別できない．よって集合  $Y'_s$  は本物の集合  $Y_s$  と区別がつかないのでシミュレーション可能と言える．

次にペアの集合  $\langle f_{er}(h(V_r)), f_{es}(f_{er}(h(V_r))) \rangle$  を偽造するのは容易である．この情報には機関  $R$  自身のデータベースしか関わっていないので機関  $R$  が集合  $f_{es}(h(V_s))$  を偽造する際に用いた鍵  $f_{e'_s}$  を用いて集合  $f_{er}(h(V_r))$  を暗号化し，集合  $f_{e'_s}(f_{er}(h(V_r)))$  を作成して暗号化前の集合  $f_{er}(h(V_r))$  とペアにすることにより機関  $R$  は自分自身でペアの集合  $\langle f_{er}(h(V_r)), f_{e'_s}(f_{er}(h(V_r))) \rangle$  を作成することができる．

機関  $S$  と  $R$  の両方の機関がシミュレート可能なので機関  $S$  と  $R$  が知ることを許された情報以外の情報が機関  $S$  と  $R$  にもたらされていないと証明できる．

### 5. 特性 4 を満たさない場合における情報漏洩

仮に暗号化関数に特性 4 が仮定されない場合にどんな情報が明らかになってしまうのかを考える．

特性 4 がないと仮定すると，以下のようなタプル

$$\begin{pmatrix} x_1 & x_2 \\ f_{e'_s}(x_1) & z \end{pmatrix}$$

が与えられた場合，ペア  $\langle x_1, f_{e'_s}(x_1) \rangle$  と  $\langle x_2, z \rangle$  が同じ鍵 (ここでは  $e'_s$ ) で暗号化されたものかどうかを多項式時間内に判定できるアルゴリズム  $A$  が存在すると仮定される．暗号化関数に特性 4 がないと仮定したとき，どのように本物とシミュレーションの区別がついてしまうかについて考える．また区別がつくということは何

らかの情報为本物のやり取りにおいて漏れているということになるので漏洩した情報を明らかにする。

まず本物の通信において暗号化関数に仮定されている特性 4 がない場合にどのような情報が漏れてしまうのかを考える。この場合プロトコルを行うことで情報が漏れてしまう原因は機関 S から R に送られる集合 Ys にある。集合 Ys は以下のようなものである。

$$Ys = \{ f_{es}(x_1), f_{es}(x_2), \dots, f_{es}(x_p) \}$$

これを暗号化前のハッシュ値とペアにして  $2 \times p$  タプルにして表すと以下のようなになる。

$$\begin{pmatrix} x_1 & x_2 & \dots & x_q & x_{q+1} & \dots & x_p \\ f_{es}(x_1) & f_{es}(x_2) & \dots & f_{es}(x_q) & f_{es}(x_{q+1}) & \dots & f_{es}(x_p) \end{pmatrix}$$

このタプルにおいて機関 R は  $x_{q+1} \sim x_p$  の値以外を知っている。よって以下のようなタプルを機関 R はアルゴリズム A に入力できる。

$$\begin{pmatrix} x_1 & x_2 & \dots & x_q \\ f_{es}(x_1) & f_{es}(x_2) & \dots & f_{es}(x_q) \end{pmatrix}$$

このタプルをアルゴリズム A に入力すると同じ鍵  $e_s$  を用いて暗号化されているのでアルゴリズム A は真を返す。次に機関 R は intersection でない部分の一つを選び(ここでは  $f_{es}(x_{q+1})$ )それを加えた以下のようなタプルを考える。

$$\begin{pmatrix} x_1 & x_2 & \dots & x_q & x_{q+1} \\ f_{es}(x_1) & f_{es}(x_2) & \dots & f_{es}(x_q) & f_{es}(x_{q+1}) \end{pmatrix}$$

上記のタプルは intersection でない部分のペア  $\langle x_{q+1}, f_{es}(x_{q+1}) \rangle$  を一つ加えたものである。機関 R がこのタプルをアルゴリズム A に入力するには値  $x_{q+1}$  の値を知らなければならない。しかし、値  $x_{q+1}$  の値は有限集合 V に含まれるある値をハッシュ化した値である。よって、有限集合 V から一つ値を選び、ハッシュ化し、それを値  $x_{q+1}$  として上記のタプルをアルゴリズム A に入力する。値  $f_{es}(x_{q+1})$  とペアになっているハッシュ値を選ぶとアルゴリズム A が真を返す。アルゴリズム A が真を返したときの値  $x_{q+1}$  は機関 S のデータベースにしか存在しないデータのハッシュ値である。同様に  $x_{q+1} \sim x_p$  の値を導出できる。つまり特性 4 が仮定されていなければ機関 S しか知らない情報の全てが機関 R に多項式時間内に漏れてしまうことになる。同様に機関 R のプロトコル内での計算を見ることが出来る第三者も  $x_{q+1} \sim x_p$  の値を知ることができる。ここで第三者は本物

の通信において得られる機関 S しか知らない情報をハッシュ化した値  $x_{q+1}, \dots, x_p$  を利用して集合 Y's が本物かどうかを確かめる。第三者は本物のハッシュ値が  $x_{q+1}, \dots, x_p$  ということを知っているの、そのハッシュ値の中から一つ値を選び(ここでは  $x_{q+1}$ ) その暗号化後の値として  $f_{e's}(H_{q+1}), \dots, f_{e's}(H_p)$  とペアにしてアルゴリズム A に入力してみる。本物ならば真を返すような値が  $f_{e's}(H_{q+1}), \dots, f_{e's}(H_p)$  の中に一つ存在する。つまり値  $f_{e's}(H_{q+1}), \dots, f_{e's}(H_p)$  を  $x_{q+1}$  とのペアとして試してもアルゴリズム A が全てに対して偽を返すと第三者はその時の集合 Y's が偽物だと判断できる。この流れを図 1 に示す。

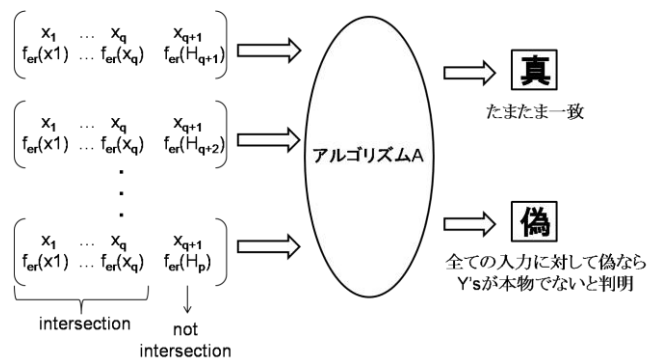


図 1 アルゴリズム A を用いて本物かを区別する例

入力が偽造した集合 Y's であるときアルゴリズム A が真を返した場合はたまたま本物のハッシュ値  $x_{q+1}$  とペアになる  $f_{e's}(H_i) (i=q+1 \sim p)$  を選んでいただけである。残りのハッシュ値  $x_{q+1}, \dots, x_p$  についても同様のことが言える。偽造の際に本物から得たハッシュ値  $x_{q+1}, \dots, x_p$  の全てに対応する  $f_{e's}(H_{q+2}), \dots, f_{e's}(H_p)$  を作成することは極めて困難である。よって本物と偽物の区別がつかない、シミュレーションが不可能と言える。

このことから特性 4 を仮定しなければ本物と偽物の区別がつかない、シミュレーションはできない。また本物の通信においては機関 R に S の intersection 部分でないハッシュ値が漏れることになる。

以上のことから特性 4 が機密性を確保する上で重要になることがわかる。

また、特性 4 が暗号化関数に仮定されていないことはある暗号化前後のペアが与えられたとき、次に与えられるペアが同じ鍵で暗号化されたものかどうかを多項式時間内に判定できるアルゴリズム A があるということの意味していた。しかし、見方を変えると入力としてある暗号化前後のペア  $\langle x_1, f_{e_s}(x_1) \rangle$  と別の暗号化後の値  $f_{es}(x_2)$  とその暗号化前の値が含まれている値の集合  $h(V)$  が与えられたとき、暗号化後の値  $f_{es}(x_2)$  を復号した値  $x_2$  を出力するアルゴリズム B があると言い換えられることがこの節の議論を通してわかった。

アルゴリズム B を用いて情報を得る例を図 2 に示す。

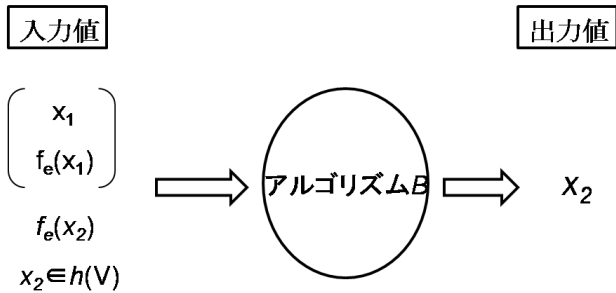


図 2 アルゴリズム B の動作例

このことを踏まえ特性 4 を緩めることができるようなプロトコルを次の節で提案する。

## 6. 提案プロトコル

この節では従来プロトコルの暗号化関数に仮定されていた特性 4 を緩めても機密性を保った情報共有を可能とするプロトコルを提案する。

### 6.1 提案 Intersection プロトコル

暗号化関数に特性 4 を仮定しない場合、暗号化前後の関係からアルゴリズム A に入力したときハッシュ値が明らかになってしまっていたので、従来のプロトコルのステップ 4 において機関 S が暗号化前と後をペアの集合  $\langle Y_r, f_{e_s}(Y_r) \rangle$  にして機関 R に送るという通信を行わずに情報共有するプロトコルを考えた。

我々が提案するプロトコルを以下に示す。

1. 機関 S と R はそれぞれが持つ集合  $V_s, V_r$  をハッシュ関数  $h$  より変換する。変換後の集合をそれぞれ  $X_s=h(V_s), X_r=h(V_r)$  とする。機関 S と R は領域  $KeyF$  からランダムに鍵  $e_s, e_r$  を選ぶ。
2. 機関 S と R はハッシュ化された集合を選んだ鍵で暗号化する。暗号化後の集合をそれぞれ  $Y_s=f_{e_s}(X_s), Y_r=f_{e_r}(X_r)$  とする。
3. 機関 S は集合  $Y_s$  の要素を辞書式順に並び換えた列を機関 R に送る。
4. 機関 R は集合  $Y_r$  の要素を辞書式順に並び換えた列を機関 S に送る。
5. 機関 S は鍵  $e_s$  を用いて機関 R から送られてきた集合  $Y_r$  を暗号化する。集合  $Y_r$  を暗号化したものを集合  $Z_r=f_{e_s}(Y_r)$  とする。
6. 機関 R は S から送られてきた  $Y_s$  を鍵  $e_r$  を用いて暗号化し、辞書式順に並べて機関 S に送り返す。集合  $Y_s$  を暗号化したものを集合  $Z_s=f_{e_r}(Y_s)$  とする。
7. 機関 S は R から送り返された集合  $Z_s$  とステップ 5 で作成した集合  $Z_r$  を比較し、集合  $Z_s \cap Z_r$  を求める。
8. 機関 S はステップ 7 で求めた集合  $Z_s \cap Z_r$  を鍵  $e_s$

で復号する。復号すると集合  $W_s=f_{e_r}^{-1}(Z_s \cap Z_r)=f_{e_r}(h(V_s \cap V_r))$  を得られる。復号した集合  $f_{e_r}(h(V_s \cap V_r))$  を機関 R に送る。

9. 機関 R は S から送られた集合  $W_s$  を鍵  $e_r$  で復号し、集合  $h(V_s \cap V_r)$  を求め集合  $V_s \cap V_r$  を得る。

### 6.2 提案プロトコルの機密性を確保する暗号化特性

提案プロトコルの機密性を証明するには 4 節で述べたように、お互いがプロトコルを通して得られる知識をあらかじめ与えられてシミュレーションできることを証明できればよい。

提案プロトコルでは機関 S がサイズ  $|V_r|$  と  $|V_s \cap V_r|$  を、機関 R がサイズ  $|V_s|$  と集合  $V_s \cap V_r$  を知ることを許している。よって、これらの知識をお互いにあらかじめ与えられていると仮定する。その上で機関 S と R の両方が各自で二者間のやり取りをシミュレート可能であればよい。ここで従来プロトコルでは暗号化前後のペアが分かっていたが、提案プロトコルでは相手機関に暗号化後の値のみしかわからないようになっているため、提案プロトコルに用いる暗号化関数は特性 4 の代わりに以下のような特性を暗号化関数に仮定する。

[encryption-indistinguishability 特性]

$\langle x_1, f_e(x_1), x_2, f_e(x_2) \rangle$  の分布は  $\langle x_1, z_1, x_2, z_2 \rangle$  の分布と計算的に区別がつかない。

$$(x_1, x_2, z_1, z_2 \in {}_r DomF, e \in {}_r KeyF)$$

この特性は入力としてある暗号化後の値の組  $(f_e(x_1), f_e(x_2))$  とそれらの暗号化前の値が属している集合  $U (x_1, x_2 \in U)$  が与えられたとき、暗号化後の値の組  $(f_e(x_1), f_e(x_2))$  を何らかの鍵で復号した値の組  $(x_1, x_2)$  の集合を出力する多項式時間アルゴリズム C はないということを意味している。また以降本稿では encryption-indistinguishability 特性を E-I 特性と略記する。

この特性を暗号化関数に仮定しなければ図 3 のように復号した値の組  $(x_1, x_2)$  を導くことが可能になる。

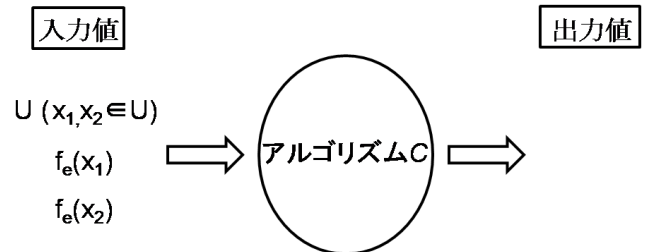


図 3 アルゴリズム C の動作例

暗号化関数にこの特性を仮定して提案プロトコルにおける機関 S と R の通信を機関 S と R が独立にシミュレ

ートすることが可能になるか考察する。

### 6.3 提案プロトコルの機密性

提案プロトコルの機密性を証明するには4節で述べたように、お互いがプロトコルを通して得られる知識をあらかじめ与えられてシミュレーションできることを証明できればよい。

提案プロトコルでは機関 S が得られる情報はサイズ  $|V_r|$  と  $|V_s \cap V_r|$  を、機関 R が得られる情報はサイズ  $|V_s|$  と集合  $V_s \cap V_r$  を知ることを許している。よって、これらの知識をお互いにあらかじめ与えられていると仮定する。その上で機関 S と R の両方がシミュレート可能であればよい。ここで提案するプロトコルに用いる暗号化関数は *indistinguishability* 特性をもたないと仮定する。

#### 6.3.1 E-I 特性を仮定した場合における機関 S によるシミュレーション

機関 S が通信において機関 R から得る情報はプロトコルのステップ 4 で受け取る集合  $Y_r$  とステップ 6 で受け取る集合  $Z_s$  の 2 つなのでそれらの偽造を考える。また、本物と偽造のやりとりを比べて多項式時間内に区別がつくか、つかないかでシミュレーションが可能かどうかを判断している。

まず、本物の通信において集合  $Y_r$  から機関 R の情報が引き出せるか検討する。機関 R から受け取る集合  $Y_r$  は以下のようなになる。

$$Y_r = \{f_{er}(x_1) \ f_{er}(x_2) \ \dots \ f_{er}(x_n) \ f_{er}(x_{n+1}) \ \dots \ f_{er}(x_m)\}$$

ここで、提案プロトコルでは相手機関に暗号化後の値と対応する暗号化前の値がわからないような通信に変えたため、暗号化後の値  $f_{er}(x_i)$  に対応するハッシュ値  $(x_i)$  が全てわからなくなっている。しかし、プロトコルのステップ 7 で機関 S が得る集合  $Z_s \cap Z_r$  から *intersection* の暗号化した値  $f_{er}(x_1), \dots, f_{er}(x_n)$  とそうでない部分の暗号化した値  $f_{er}(x_{n+1}), \dots, f_{er}(x_m)$  とを区別することができる。また、機関 S は *intersection* 部分である  $f_{er}(x_1), \dots, f_{er}(x_m)$  に対応するハッシュ値  $x_1, \dots, x_m$  の元の値は機関 S のデータベースに含まれていることが分かる。同様に *intersection* 部分でないハッシュ値  $x_{n+1}$  は有限集合  $V$  から機関 S のデータベースにある要素を除いた  $|V - V_s|$  個の要素をハッシュ化した値の中に含まれていることも分かる。つまり、*intersection* の暗号化した値と *intersection* 部分でない暗号化した値を一つずつ選び以下のようなタプル

$$\left( \begin{array}{cc} y & y' \\ f_{er}(x_i) & f_{er}(x_{n+1}) \end{array} \right) \quad (y \in h(V_s), y' \in h(V - V_s))$$

を考える。このとき、新たに E-I 特性を暗号化関数に仮定しているので暗号化後の入力  $f_{er}(x_i), f_{er}(x_{n+1})$  と集合  $h(V)$  に対して出力として対応するハッシュ値の組  $(y, y')$  を導き出すようなアルゴリズム  $C$  が存在しないと仮定される。よって、本物の集合  $Y_r$  を機関 S が解析しても情報は得られない。これは通信を見ている第三者も同様のことが言える。

次に機関 S による集合  $Y_r$  の偽造について考える。機関 S はサイズ  $|V_r|$  と  $|V_s \cap V_r|$  という情報は与えられているので、集合  $Y_r$  に含まれる *intersection* の個数  $|V_s \cap V_r|$  と *intersection* でない部分の個数  $|V_r - V_s|$  を知ることができる。よって、*intersection* 部分の  $|V_s \cap V_r|$  個の情報を自分のデータベースから鍵  $e'_r$  を用いて暗号化する。残りの *intersection* でない部分の  $|V_r - V_s|$  個については有限集合  $V$  に含まれる値をランダムに選び、その値をハッシュ化する。そのハッシュ値  $(H_i)$  を鍵  $e'_r$  を用いて暗号化し、以下のように集合  $Y_r$  を偽造したとする。偽造した集合を  $Y'_r$  とする。

$$Y'_r = \{f_{e'_r}(x_1), \dots, f_{e'_r}(x_n), f_{e'_r}(H_{n+1}), \dots, f_{e'_r}(H_m)\}$$

この集合  $Y'_r$  に対して本物で行ったことと同様にして第三者は以下のようなタプル

$$\left( \begin{array}{cc} y & y' \\ f_{e'_r}(x_i) & f_{e'_r}(H_{n+1}) \end{array} \right)$$

を考える。第三者はこのタプルにおいて暗号化後の値  $f_{e'_r}(x_i)$  と  $f_{e'_r}(H_{n+1})$  とその暗号化前の値が  $h(V)$  に属していることを知っているがそれらに対応するハッシュ値は知らない。さらに、暗号化後の値  $f_{e'_r}(x_i)$  と  $f_{e'_r}(H_{n+1})$  から本物と区別がつくような情報を得ようとしても、E-I 特性が暗号化関数に仮定されているため何も情報を得ることができない。よって、本物と比べても  $Y'_r$  が偽物かどうか区別がつかない。

次に本物の通信においての機関 R から送られてくる集合  $Z_s$  について考える。集合  $Z_s$  は機関 S のデータベースを機関 S と R の鍵  $e_s, e_r$  を用いて暗号化したもので、シミュレーションでは機関 S が S 自身の鍵  $e_s$  と機関 R の鍵を偽造するためランダムに選んだ鍵  $e'_r$  の 2 つの鍵を用いて暗号化することで偽造できる。偽造した集合を  $Z'_s$  とし、 $Z_s$  と比べても暗号化に用いた鍵が違うのだけなので第三者は集合  $Z'_s$  と  $Z_s$  とを区別することができない。

以上より機関 S は独自にシミュレーションを行うことが可能と言える。

### 6.3.2 E-I 特性を仮定した場合における機関 R によるシミュレーション

機関 R が通信において機関 S から得る情報はプロトコルのステップ 3 で受け取る集合  $Y_s$  とステップ 8 で受け取る集合  $f_{er}(h(V_s \cap V_r))$  の 2 つである。

まず、本物の通信において集合  $Y_s$  から機関 S の情報が引き出せるか考える。本物の通信で機関 R が S から送られる集合  $Y_s$  は以下のとおりである。

$$Y_s = \{f_{es}(x_1) \ f_{er}(x_s) \ \cdots \ f_{er}(x_q) \ f_{er}(x_{q+1}) \ \cdots \ f_{er}(x_p)\}$$

提案プロトコルでは相手機関に暗号化後の値と対応する暗号化前の値がわからないようになっているため、機関 R は暗号化した値  $f_{es}(x_i)$  に対応するハッシュ値  $x_i$  がわからない。その上 intersection の暗号化した値とそうでない部分の暗号化した値とを区別することができない。ここで  $|Y_r|(|Y_r|-1)/2$  の組合せの中から intersection 部分とそうでない部分を選び出せた場合の暗号化後の組  $(f_{es}(x_i), f_{es}(x_{q+1}))$  において以下のタプル

$$\begin{pmatrix} y & y' \\ f_{es}(x_i) & f_{es}(x_{q+1}) \end{pmatrix} \quad (y \in h(V_r), y' \in h(V - V_r))$$

を考える。このタプルにおいて機関 R がプロトコルを通して知る情報は暗号化後の値  $f_{es}(x_i)$  と  $f_{es}(x_{q+1})$  のみである。ここで、機関 S のシミュレーションと同様に新たな特性を暗号化関数に仮定しているので暗号化後の値  $f_{es}(x_i)$  と  $f_{es}(x_{q+1})$  とそれらの暗号化前の値が属している集合  $h(V)$  から対応するハッシュ値の組  $(y, y')$  を導き出すようなアルゴリズム C は存在しない。よって、機関 R は集合  $Y_r$  を解析しても知識を得ることができない。

次に集合  $f_{er}(h(V_s \cap V_r))$  の偽造を考える。この集合は intersection である集合  $V_s \cap V_r$  を機関 R 自身の鍵  $e_r$  で暗号化した集合である。プロトコルを通して機関 R が集合  $V_s \cap V_r$  を知ることを許しているため、その知識を使って偽造できる。偽造した集合と本物の通信で得る集合  $f_{er}(h(V_s \cap V_r))$  は同じものなので第三者は区別することができない。

以上より機関 R は独自にシミュレーションを行うことが可能と言える。

### 6.4 機密性に関する考察

提案プロトコルにおいては暗号化関数に特性 4 を仮定していない代わりに E-I 特性を仮定した。それにより、機関 S と R が通信で得る情報を全て偽造することが可能で、第三者に本物と偽物の区別が多項式時間内ではつかないのでゼロ知識性を満たす。よって、提案プロトコルの機密性は確保されていると言える。

次に、機密性の向上に関して説明する。従来プロトコルでは 5 節で述べたアルゴリズム B が存在すると機関 S のみが持つ情報のハッシュ値が機関 R に漏れていた。このとき、シミュレーションを不可能にするような情報を得るために、アルゴリズム B は入力として暗号化前後のペア  $\langle x_1, f_e(x_1) \rangle$  と暗号化後の値  $f_e(x_2)$  が必要としていた。次に、提案プロトコルにおいて E-I 特性を暗号化関数に仮定しない場合、6.2 節で述べたアルゴリズム C からシミュレーションを不可能にするような情報を得るために、入力として暗号化後の値  $f_e(x_1)$  と別の暗号化後の値  $f_e(x_2)$  が必要になった。二つのアルゴリズム B と C が必要とする入力には暗号化前後の値のペアを必要とするか暗号化後の値のみを必要とするかの違いがある。また、二つのアルゴリズムはシミュレーションが行えなくなるような同じ情報を入力する。よって、出力される情報の価値が同じなので入力する情報が少ないアルゴリズム C は B より強力なアルゴリズムと言える。よって、より強力なアルゴリズム C がなければ機密性を崩すことのできない提案プロトコルのほうが従来プロトコルと比べて機密性が高いと言える。

### 7. おわりに

本稿では機密性を確保しつつ情報共有を行うために提案されている従来手法を紹介した。その従来プロトコルの機密性を確保するために暗号化関数に仮定されている特性が仮定されていない場合にどのような情報が漏れてしまうのかを考察することで、機密性の向上につながる E-I 特性を暗号化関数に仮定し、情報共有を行うプロトコルを提案した。その上で機密性の証明を行った結果、機密性を確保しつつ情報共有を行えることを示した。また文献[4]では情報共有を行う際に相手機関が敵対者(情報を得るために不正を行う者)の場合についても考えられている。よって、敵対者の場合に対しても安全に情報共有ができるよう提案プロトコルを拡張することが今後の課題として挙げられる。

### 文 献

- [1] R. Agrawal, A. Evfinievski and R. Srikant, "Information sharing across private databases," Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD 2003), pp.86-97, 2003.
- [2] D. Boneh, "The decision Diffie-Hellman problem," Proceedings of the 3<sup>rd</sup> International Algorithmic Number Theory Symposium, volume 1423 of Lecture Notes in Computer Sciences, pp.48-63, 1998.
- [3] B. Schneier, "Applied Cryptography," Second Edition, John Wiley & Sons, 1996.
- [4] N. Zhang, W. Zhao, "Distributed privacy preserving information sharing" Proceedings of the 31<sup>st</sup> VLDB Conference, pp.889-900, 2005.