社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2008－CSEC－42（27）
2008 / 7 / 25

Relational Algebra in Multi-Party Protocol to Enable Secure and Unlimited Databases

Masanori SHIMURA† Tsukasa ENDO† Kunihiko MIYAZAKI†† Hiroshi YOSHIURA†††
†, †††Graduate school of Electro-Communications, University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan
††Hitachi,Ltd.,Systems Development Laboratory
292,Yoshida-cho,Totsuka-ku,Yokohama-shi,Kanagawa,244-0817 Japan
E-mail: †{shimura, endo}@edu.hc.uec.ac.jp, ††kunihiko.miyazaki.zt@hitachi.com, †††yoshiura@hc.uec.ac.jp

Abstract As personal information comes to be in digital and transferred on networks, its leakage is becoming more and more serious social problem. Among various ways of personal information leakage, the leakage from databases is most serious because databases store vast amount of personal information. Methods of making data unreadable even if they have been copied outside are therefore studied actively using secret sharing and cryptography. With these previous methods, however, legal queries are limited, i.e., structural operations over multiple tables (such as JOIN) are impossible. In this paper, we take into account the fact that structural operations of relational databases are modeled by relational algebra. We then propose a method that can execute relational algebra in a multi-party protocol and thus can perform any structural operation over secret-shared databases without restoring plain text data.

Keywords relational database, relational algebra, secret sharing scheme, multi-party protocol,

安全で機能制限のないデータベースを実現するマルチパーティプロトコルを用いた関係代数演算

志村 正法† 遠藤 つかさ† 宮崎 邦彦†† 吉浦 裕†
†電気通信大学大学院 電気通信学研究科 〒182-8585 東京都調布市調布ヶ丘 1-5-1
††株式会社日立製作所 システム開発研究所 〒244-0817 神奈川県横浜市戸塚区吉田町 292 番地

あらまし 個人情報が電子化されネットワーク上で授受されるに従い，その漏洩が社会問題となっている．個人情報の漏洩には様々な形態があるが，なかでもデータベースからの漏洩は大量の個人情報が一度に漏洩するので，極めて甚大な被害をもたらす．データベースからの情報漏洩対策として，秘密分散法及び暗号を用いてデータが漏洩しても読めないようにする方法がある．しかしこれらの従来方法を採用した場合，JOIN 演算など，複数のテーブルにまたがる構造演算が不可能であった．本論文では，関係データベースの構造演算が関係代数によってモデル化されることに着目する．マルチパーティプロトコルを用いて関係代数演算を実現し，秘密分散法によって分散されたデータベース上で，データを一度も復元することなく全ての構造演算を可能とする．

キーワード 関係データベース，関係代数，秘密分散法，マルチパーティプロトコル

1. はじめに

個人情報が電子化されネットワーク上で授受されるに従い，その漏洩が社会問題となっている．個人情報の漏洩には様々な形態があるが，なかでもデータベースからの漏洩は大量の個人情報が一度に漏洩するので，極めて甚大な被害をもたらす．データベースからの情報漏洩対策として，秘密分散法および暗号化を用いてデータが漏洩しても利用できないようにする方法が提案されている．秘密分散法を用いた方式では，データの復元に必要な数のシェアが漏洩しない限りデータは保護される．暗号化を用いた方式では，暗号化データと鍵が同時に漏洩しない限りデータは保護される．しかし上記 2 つの方式を採用した場合，データベースのテーブルに格納された情報を行単位で取り出すような単純検索は可能だが，数値演算や複数のテーブルにまたがる演算(構造演算)は困難であった．検索中にデータを一時的に復元・復号すれば数値演算や構造演算は可能となるが，平文となったデータが漏洩する可能性がある．筆者らの研究の目的はデータが漏洩することを防ぎつつ，任意の演算を可能にすることである．本研究ではその第一ステップとして，データベースサーバ内で一度もデータを復号・復元することなく，全

ての構造演算を可能とする方式を提案する．関係データベースの全ての構造演算は関係代数によって数学的にモデル化されている．提案方式では秘密分散法を用いてデータベースを複数に分散し，その分散データベースサーバ間でマルチパーティプロトコルを用いた関係代数演算を行う．これにより，データベース内でデータを復元せずに構造演算が可能となる．2 章ではデータベースの利用と攻撃について述べる．3 章では従来のデータベース保護方式について分析する．4 章では目標と方針を挙げる．5 章では提案方式について述べる．6 章では関係代数による演算をマルチパーティプロトコルで実現するための論理回路の設計を行う．7 章で提案方式の評価を行い，8 章で結論を述べる．

2. データベースの利用と攻撃

2.1. データベースの利用

データベースは複数のテーブルによって構成される．図 1 の例では，「学生基本情報」「指導教員」「数学の成績」「英語の成績」などのテーブルが存在する．データベースへの問い合わせは以下の 4 種類に大別される．

① 単純取出

指定したテーブル内のデータを行単位で取り出す．例えば，ユーザは ID が 001 の学生情報を要求する．データベースは「学生情報」から ID が 001 に該当する行をユーザに返す．

② 数値演算

1 つのテーブル内の数値データを用いて演算する．ユーザは学生全員の数学の平均点を要求し，データベースは「数学の成績」から平均点を演算し，結果をユーザに返す．

③ 構造演算

1 つあるいは複数のテーブルから新しいテーブルを構成する．ユーザは数学の成績が D である学生とその指導教員に関するデータを要求する．データベースは「数学の成績」で評価が D である学生の ID を参照する．「学生基本情報」からその ID に該当する学生のクラスを参照する．「指導教員」からそのクラスに該当する教員名を取り出す．ID，氏名，評価，教員名を 1 つのテーブルにし，ユーザに返す．

④ 数値演算と構造演算の組合せ

ユーザは学生全員の数学と英語の成績，及び両科目の合計点を要求する．データベースは「数学の成績」と「英語の成績」における ID でデータをまとめ，さらに合計点を計算したデータを合わせて一つのテーブルにし，ユーザに返す．

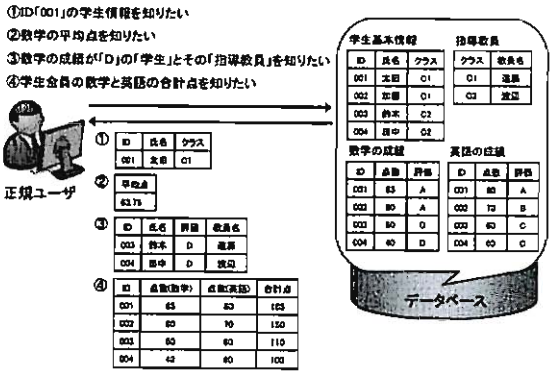

図 1　データベースの利用例

2.2. データベースへの攻撃

データベースへの攻撃は少なくとも以下のようなケースが考えられる(図 2)．

① 他人へのなりすまし

正規ユーザになりすまし，許可されないデータを取得する．認証などの技術的対策やパスワード管理などの人的対策がとられている．

② 外部者の侵入

データベースサーバに侵入し，データを盗む．侵入検知システムやファイヤーウォールなどの対策がとられている．また侵入後に盗難されたデータを利用できないようにするため秘密分散法や暗号化を用いた対策がとられている．

③ 内部者の不正

②と同様にネットワーク経由で攻撃する場合と USB などの外部メモリにデータをコピーする場合がある．②への対策に加えて内部者に対する人的対策がとられている．

④ 正規ユーザの不正

正規のユーザが大量のデータを取得し，外部に漏洩させる．対策として，一度の検索で取得可能なデータ量や検索の回数の制限がある．

本研究では，②外部者の直接侵入，③内部者の不正への対策を考える．従来研究として，次節が提案されている．

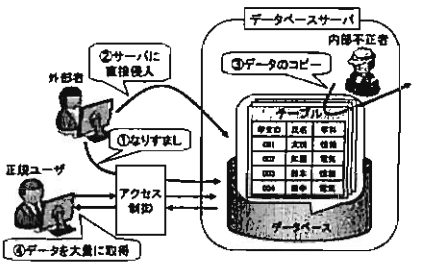

図 2　データベースへの攻撃

3. 先行研究

外部からの侵入者や内部不正者にデータを盗まれても利用できないようにする方式として，秘密分散法を用いたデータベース方式と暗号化データベース方式が提案されている．

(1) 秘密分散法を用いた分散データベース方式

[1]では，各データを(k,n)しきい値秘密分散法を用いてn個のシェアに分散し，異なるn個のデータベースサーバにそれぞれ保存する．検索時，ユーザは復元したい行のIDを各データベースサーバに送信する．ユーザはデータベースサーバから送信されたシェアを集め，元のデータを復元する．シェアを配置するデータベースサーバの数をシェアの数より多くすることによって安全性を高めている．[1]では，2.1における①単純取出が可能であるが，②数値演算，③構造演算は考慮されていない．

[2]においても，(k,n)しきい値秘密分散法を用いた方式が提案されている．[2]ではテーブルをデータベースサーバ内で一時的に復元して検索を行う．全ての演算が可能だが，サーバ内に平文のテーブルが生成されるため，侵入者と内部者の不正に弱い．[2]ではテーブルの復元範囲を小さくするため，テーブルを複数のサブテーブルに分割している．しかし平文のテーブルがサーバ内に生成されることに変わりはなく，検索によっては復元範囲をサブテーブルに限定できない場合がある．また別の方法としてテーブルのシェアをユーザに送り，ユーザがテーブルを復元した後に検索を行う方法がある．しかしユーザに検索対象外の大量の平文データを与えてしまう問題がある．

(2) 暗号化データベース方式

暗号化されたデータに対する問合せ方法が[3]によって提案されている．この方式は行単位でデータを暗号化するため，行の中の個々の要素（図1における点数，教員名など）を参照することができない．そのため2.1の①単純取出しは可能だが，要素間の②数値演算，③構造演算ができない．

なお個々の要素単位で準同型性暗号を用いることで限定された数値演算は可能となるが，構造演算は不可能である．検索時に一時的にテーブルを復号すれば構造演算は可能となるが，[2]と同様に平文のテーブルが生成されるため侵入者，内部不正に弱い．

ここで，演算時にサーバ側でのデータの復元・復号が不要であることを復号不要性と呼ぶことにする．各方式が達成できる機能と復号不要性について表1にまとめた．

表1　各方式のまとめ

		秘密分散法を用いた方式		暗号化方式	目標	提案方式
		[1]	[2]	[3]		
機能	①単純取出	○	○	○	○	○
	②数値演算	×	○	×(注1)	○	検討中
	③構造演算	×	○	×	○	○
復号不要性		○	×	○	○	○

注1:準同型性暗号を用いた場合は限定された数値演算は可能

従来において，①単純取出，②数値演算，③構造演算の機能と復号不要性を同時に達成できる方式はなかった．特に，構造演算と復号不要性を同時に達成する方式はなかった．筆者らの研究の目的は三つの機能と復号不要性を同時に満たすことである．ここではその第一ステップとして，①単純取出，③構造演算と復号不要性を同時に達成する方式を提案する．

4. 目標と方針

4.1. 達成目標

本研究で提案する方式では，以下の3つを同時に達成することを目標とする．

(i) 全ての構造演算を行うことができる

(ii) 復号不要性を満たす

上記を満たす自明な方式として，ユーザ側のコンピュータでテーブルを復号した後に検索を行う方法が考えられる．しかしユーザが検索対象以外の情報まで得られるという問題がある．よって上記(ⅰ)，(ⅱ)に以下の目標を加える．

(iii) ユーザは検索対象以外の情報を得ることができない

4.2. 方針

現在のデータベースは関係データベース，及びその発展形である．関係データベースにおける構造演算は関係代数によって数学的にモデル化されている．上記の目標を達成するため，提案方式は秘密分散法を用いたデータベース方式を基礎とし，以下を方針とする．

1 秘密分散法を用いてデータをシェアに分散し，それぞれを個々のデータベースサーバに格納する

2 個々のデータ・個々のデータベースサーバ管理者をそれぞれ，マルチパーティプロトコルの秘密情報・参加者と考える

3 関係代数を論理回路で表現し，個々のデータベース管理者間のマルチパーティプロトコルによって演算する

上記の演算により，サーバ内でシェアを一度も復号することなく，個々のシェアから関係代数を演算可能にし，構造演算を可能にする．

4.3. 前提技術

4.3.1. 関係データベース

関係データベースは複数のテーブルによって構成されている．テーブルは $A(A_1, A_2,..., A_n)$ で表される．ここで A はテーブル名で，個々の $A_1, A_2,..., A_n$ はテーブル A の属性であり，それぞれ $1,2,...,n$ 列目である．それぞれの属性 $A_1, A_2,..., A_n$ に格納できる属性値の集合を，その属性のドメインと呼ぶ．テーブル A は行の集合とみなされ，$t \in A$ とは t がテーブル A の 1 つの行であることを示す．$t[A_j]$ とは任意の行の j 番目の列の値を示す．

主キーとはテーブル A に対する唯一の識別子である．これはある列または列の組み合わせの中に同じ値を含む 2 つの行が存在しないという性質をもつ列または列の組み合わせである．以下では簡単のため，主キーは 1 列である．

4.3.2. 関係代数

テーブルに対する演算方法を定めた集合演算である．一般的な集合演算により定義される和，差，共通，直積と，関係代数に特有な射影，選択，結合，商の計 8 つの演算子により定義される．和，差，共通は和両立を満たしている場合に演算が可能である．

テーブル $A(A_1, A_2,..., A_n)$ とテーブル $B(B_1, B_2,..., B_m)$ が和両立とは，次の 2 つの条件を満たしていることをいう．

(i) A と B の列数が等しい $(n = m)$

(ii) 各 $j(1 \leqq j \leqq n)$ について A_j と B_j のドメインが等しい

例として，表 2 と表 3 は和両立である．

① 和

A，B の和 $A \cup B$ は，以下のように定義される．

定義：$A \cup B = \{t \mid t \in A \vee t \in B\}$

A の全ての行と B の全ての行で重複なく構成されるテーブルを返す．表 2 と表 3 の和を表 4 に示す．

② 差

A，B の差 $A - B$ は，以下のように定義される．

定義：$A - B = \{t \mid t \in A \wedge \neg(t \in B)\}$

A から，B に属する行を取り除いたテーブルを返す．表 2 と表 3 の差を表 5 に示す．

③ 共通

A，B の共通 $A \cap B$ は，以下のように定義される．

定義：$A \cap B = \{t \mid t \in A \wedge t \in B\}$

A と B の両方に属する行で構成されたテーブルを返す．表 2 と表 3 の共通を表 6 に示す．

④ 直積

A，B の直積 $A \times B$ は以下のように定義される．

定義：$A \times B = \{(s,t) \mid s \in A \wedge t \in B\}$

表2 :科目Aの履修者

学生ID	氏名
001	太田
002	加藤
003	鈴木

表3 科目Bの履修者

学生ID	氏名
001	太田
002	加藤
004	田中

表4: 和

学生ID	氏名
001	太田
002	加藤
003	鈴木
004	田中

表5: 差

学生ID	氏名
003	鈴木

表6 :共通

学生ID	氏名
001	太田
002	加藤

ただし，$s = (a_1, a_2,..., a_m), t = (b_1, b_2,..., b_n)$ とするときに (s,t) を以下のように定義する．

$(s,t) = (a_1, a_2,..., a_m, b_1, b_2,..., b_n)$

⑤ 選択

A において A_i と A_j を θ-比較可能な属性とする．このとき選択 $A[A_i \theta A_j]$ は以下のように定義される．

定義：

(i) $A[A_i \theta A_j] = \{t \mid t \in A \wedge t[A_i] \theta\, t[A_j]\}$

(ii) $A[A_i \theta A_j] = \{t \mid t \in A \wedge t[A_i] \theta\, \alpha\}$　※ α は定数

A において，指定した行のテーブルを返す．

⑥ 射影

A の全属性集合 $\{A_1, A_2,..., A_n\}$ の部分集合を $X = \{A_{i1}, A_{i2},..., A_{ik}\}\ (1 \leqq i_1 < i_2 < ... < i_k \leqq n)$ とすると，射影 $A[X]$ は以下のように定義される．

定義：$A[X] = \{t[X] \mid t \in A\}$

A から指定した列のテーブルを返す．

⑦ 結合

関係 $A(A_1, A_2,..., A_n)$ と関係 $B(B_1, B_2,..., B_m)$ において，A_i と A_j を θ-比較可能とする．このとき結合 $A[A_i \theta B_j]B$ は以下のように定義される．

定義：

$A[A_i \theta B_j]B = \{(t,u) \mid t \in A \wedge u \in B \wedge t[A_i] \theta\, u[B_j]\}$

A と B において関連する属性に注目し，2 つのテーブルを合わせて 1 つのテーブルを返す．表 7 と表 8 の結合を表 9 に示す．

表7 学生

学生ID	氏名	所属学科ID
001	太田	d001
002	加藤	d002
003	鈴木	d002
004	田中	d003

表8 学科

学科ID	学科名	所属学部
d001	機械	工学部
d002	電気	工学部
d003	情報	工学部

表9 表7と表8の結合(学生[所属学科ID] = 学科[学科ID])

学生ID	氏名	所属学科ID	学科ID	学科名	所属学部
001	太田	d001	d001	機械	工学部
002	加藤	d002	d002	電気	工学部
003	鈴木	d002	d002	電気	工学部
004	田中	d003	d003	情報	工学部

⑧　商

n次のテーブル$A(A_1, A_2, ..., A_{n-m}, B_1, B_2, ..., B_m)$と$m$次$(m<n)$のテーブル$B(B_1, B_2, ..., B_m)$に対して，商$A \div B$は以下のように定義される．

定義：

$$A \div B = \{t \mid t \in A[A_1, A_2, ..., A_{n-m}] \wedge (\forall u \in B)((t,u) \in A)\}$$

Bの全ての属性の値を同時に満たすAの行を選び出し，Bの属性を取り除いた列のテーブルを返す．

上記8つの演算子のうち，和，差，直積，選択，射影の演算子は互いに独立である．この5つの演算子を組み合わせることによって共通，結合，商を表現できる．

・共通

$$A \cap B = A - (A - B) \cdots (1)$$

・結合

$$A[A_i \theta B_j]B = (A \times B)[A.A_i \theta B.B_j] \cdots (2)$$

・商

$$A \div B = A[A_1, A_2, ... A_{n-m}] - ((A[A_1, A_2, ..., A_{n-m}] \times B) - A)[A_1, A_2, ... A_{n-m}] \cdots (3)$$

4.3.3. 一般のマルチパーティプロトコル

マルチパーティプロトコルとは，複数の参加者$A_i(i=1,...,n)$がそれぞれ秘密情報$x_i(i=1,\cdots,n)$を持つ状況下で，秘密情報を秘匿したまま，秘密情報の関数値$f(x_1,\cdots,x_n)$を計算する方法である．各参加者$A_i(i=1,...,n)$は秘密情報x_iを以下の関係を満たすn個の情報$x_{i1},...,x_{in}$に分割する．

$$x_i = x_{i1} \veebar \cdots \veebar x_{in}$$

このシェアx_{ij}を他の参加者$A_j(j=1,...,n)$に分配する．参加者はシェアの計算・通信を行い関数値$f(x_1,\cdots,x_n)$のシェアを得る．これらのシェアから関数値$f(x_1,\cdots,x_n)$を復元する．計算途中の情報は全ての参加者が結託しない限り復元できない．論理回路で記述できる関数に対しては全てこの計算が実行できる．

提案方式ではデータベース内のデータを秘密情報と考え，そのシェアを各分散データベースサーバが保持しているとする．ユーザが各分散データベースサーバに検索の関数を送り，その関数に基づいて各分散データベースサーバが計算・通信を行う．各分散データベースサーバは関数値の分散情報をユーザに送り，ユーザはその分散情報を復元し関数値，つまり検索結果を得る．提案方式では検索式を論理回路で記述するために，4.3.2の5つの独立演算子の回路を設計する必要がある．

5. 提案方式

5.1. データの格納

4.3.3の秘密分散法を用いてデータをシェアに分散する．図3においては簡単のためデータを2つのシェアに分散し，2つのデータベースに格納する．

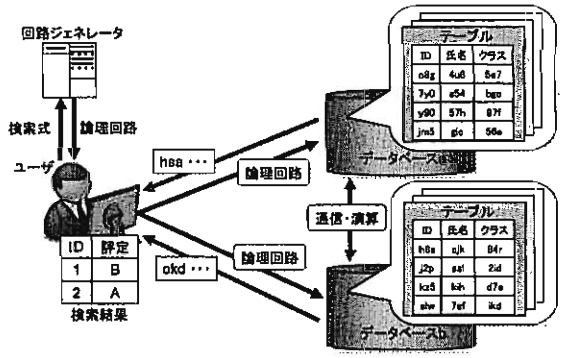

図3　提案方式の概要

5.2. 検索

1　ユーザは関係代数による任意の検索式を発行する

2　この検索式は論理回路に変換され，複数の分散データベースサーバに送られる

3　データベースサーバは論理回路に従ってマルチパーティプロトコルによる関係代数演算を行い，検索結果のシェアを導出する

4　ユーザは検索結果のシェアを取得し，検索結果を復元する

6. マルチパーティプロトコルによる関係代数演算

6.1. 独立演算子の設計

関係代数をマルチパーティプロトコルで実行するために論理回路の設計を行う．和，差，直積，選択，射影の5つの独立演算子について論理回路を設計した．

＜準備＞

- 以下では簡単のため主キーは1列とする
- テーブルAのi行j列目をa_{ij}と表す
- a_{ij}の値は全てhビット固定長であり，a_{ij}のkビット目の値を$a_{ij}^{(k)}(1 \le i \le n_A, 1 \le j \le m, 1 \le k \le h)$と表す
- b_{ij}の値も全てhビット固定長であり，b_{ij}のkビット目の値を$b_{ij}^{(k)}(1 \le i \le n_B, 1 \le j \le m, 1 \le k \le h)$と表す
- 結果はテーブルCに出力する
- c_{ij}の値も全てhビット固定長であり，c_{ij}のkビット目の値を$c_{ij}^{(k)}(1 \le i \le n_A + n_B, 1 \le j \le m, 1 \le k \le h)$と表す
- テーブルTのi行目の全ての値が0のとき，すなわち，$T_{ij}^{(k)}=0(1 \le j \le m, 1 \le k \le h)$のとき，$T$の$i$行目はnullを表す

・ 全ての値が 0 であるような行はデータの格納には用いず，null を表すためにリザーブする

① 和

＜前提条件＞

・ テーブル A を n_A 行 m_A 列，テーブル B を n_B 行 m_B 列とし，主キーはテーブル A において s 列目，テーブル B において t 列目であるとする

・ テーブル A とテーブル B は和両立であるため $m_A = m_B = m$，$s = t$，$A_j = B_j$ が成立する

Step1：テーブル A をテーブル C にコピーする

for($1 \le i \le n_A$){
  for($1 \le j \le m$){
    for($1 \le k \le h$){
      $c_{ij}^{(k)} = a_{ij}^{(k)}$
    }
  }
}

Step2：テーブル B のうち，テーブル A と重複しない行をテーブル C に追加する

for($1 \le i \le n_B$){
$d_i = \{(a_{i1}^{(1)} \veebar b_{i1}^{(1)}) \vee \ldots \vee (a_{i1}^{(h)} \veebar b_{i1}^{(h)})\} \wedge \ldots \wedge \{(a_{in_A}^{(1)} \veebar b_{in_B}^{(1)}) \vee \ldots \vee (a_{in_A}^{(h)} \veebar b_{in_B}^{(h)})\}$
  for($1 \le j \le m$){
    for($1 \le k \le h$){
      $c_{ij}^{(k)} = d_i \wedge b_{ij}^{(k)}$
    }
  }
}

・ テーブル B の i 行がテーブル A のどの行とも異なるときに d_i に 1 を出力し，テーブル C にその行を出力する

・ テーブル B の i 行がテーブル A のいずれかの行と同じであるとき d_i に 0 を出力し，テーブル C に null を出力する

② 差

前提条件は和と同じである．

for($1 \le i \le n_A$){
$d_i = \{(a_{i1}^{(1)} \veebar b_{i1}^{(1)}) \vee \ldots \vee (a_{in}^{(h)} \veebar b_{i1}^{(h)})\} \wedge \ldots \wedge \{(a_{in}^{(1)} \veebar b_{in_B}^{(1)}) \vee \ldots \vee (a_{in}^{(h)} \veebar b_{in_B}^{(h)})\}$
  for($1 \le j \le m$){
    for($1 \le k \le h$){
      $c_{ij}^{(k)} = d_i \wedge a_{ij}^{(k)}$
    }
  }
}

③ 直積

＜前提条件＞

テーブル A は n_A 行 m_A 列，テーブル B は n_B 行 m_B 列とする

for($1 \le i_A \le n_A$){
  for($1 \le i_B \le n_B$){
    for($1 \le j_A \le m_A$){
      for($1 \le k \le h$){
        $c_{(i_A-1)n_B+i_B,\,j_A}^{(k)} = a_{i_A j_A}^{(k)}$
      }
    }
    for($1 \le j_B \le m_B$){
      for($1 \le k \le h$){
        $c_{(i_A-1)n_B+i_B,\,m_A+j_B}^{(k)} = b_{i_B j_B}^{(k)}$
      }
    }
  }
}

④ 選択

本論文では数値演算を扱っていないため，比較演算子 θ が等号の場合について考える．

(i) 列 l_1 と列 l_2 の比較

＜前提条件＞

・ テーブル A を n 行 m 列とする

・ テーブル A において，列 l_1 の値と列 l_2 の値が一致する行を取り出す

for($1 \le i \le n$){
$d_i = \neg\{(a_{il_1}^{(1)} \veebar a_{il_2}^{(1)}) \vee \ldots \vee (a_{il_1}^{(h)} \veebar a_{il_2}^{(h)})\}$
  for($1 \le j \le m$){
    for($1 \le k \le h$){
      $c_{ij}^{(k)} = d_i \wedge a_{ij}^{(k)}$
    }
  }
}

(ii) 列 l と定数 α の比較

＜前提条件＞

・ テーブル A を n 行 m 列とする

・ テーブル A の l 列目の値が，ある値 α に一致する行を取り出す

for($1 \le i \le n$){
$d_i = \neg\{(a_{il}^{(1)} \veebar \alpha^{(1)}) \vee \ldots \vee (a_{il}^{(h)} \veebar \alpha^{(h)})\}$
  for($1 \le j \le m$){
    for($1 \le k \le h$){
      $c_{ij}^{(k)} = d_i \wedge a_{ij}^{(k)}$
    }
  }
}

⑤ 射影

＜前提条件＞

- テーブル A を n 行 m 列とする
- テーブル A から取り出す列の位置情報を $B=\{b_1,b_2,...,b_r\}(r<m)$ で表す(例えば 1，3，4 列目を取り出すとき，$b_1=1,b_2=3,b_3=4$ である)

```
for(1≤i≤n){
    for(1≤j≤r){
        for(1≤k≤h)
            c_ij^(k) = a_ib_j^(k)
        }
    }
}
```

6.2. 組み合わせ演算子

4.3.2 の式(1)～(3)より共通，結合，商は 5 つの独立演算子の組み合わせによって表現できる．よって 6.1 で設計した 5 つの論理回路を組み合わせることによって，共通，結合，商もマルチパーティプロトコルによって演算可能となる．

7. 評価

7.1. 機能

全ての構造演算は関係代数によって表される．8 つの関係代数演算子を論理回路で表すことができたのでマルチパーティプロトコルを用いて関係代数演算を行うことが可能となった．よって秘密分散法によりシェアを保持する分散データベースサーバ間で全ての構造演算が可能となった．

7.2. 安全性

- 演算時にサーバ内で一度もテーブルを復元することがないため，復号不要性を満たしている
- ユーザは検索結果のシェアのみを受け取るため，検索対象以外のデータを得ることができない
- 不正者は，復元に必要な個数のシェアを得るために，全ての分散データベースサーバに侵入する必要がある

8. 結論

関係代数を論理回路で表現し，マルチパーティプロトコルで実行可能とした．その結果，データベースサーバ内で一度もテーブルを復元することなく任意の構造演算を可能にした．今後の課題は以下の通りである．

- 数値演算を可能にする
- 信頼性を向上させるため，(k,n)しきい値秘密分散法を用いて提案方式を実現する
- 計算量と通信量を評価する

文　献

[1] 桜井友二，齊藤泰一，" 情報漏えい防止データベースシステムの提案，" Proc. of SCIS, 1F2-5, 2006.

[2] 守田泰博，宮本俊幸，熊谷貞俊，" 秘密分散共有法を用いた分散データベースシステム，" 信学技報 CST2004-47, PP. 49-54. 2005.

[3] H. Hacigumus, B. Iyer, CLi, and. S. Mehrotra. : " Executing SQL over Encrypted Data in the Databases-Service-Provider Model, " In Proc. of the 2002 ACM SIGMOD International Conference on Management of Data, pp. 216-227, 2002.

[4] C. J. DATE, " データベースシステム概論，" 藤原譲（訳），丸善，東京，1997

[5] 岡本龍明，山本博資，" 現代暗号，" 産業図書，東京，1997