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

2003－ＭＢＬ－27　(10)
2003－ＩＴＳ－15　(10)
2003／11／14

匿名データベースを用いた交通事故情報の企業間共有

対馬伸行† 杉野栄二† 村山優子† 宮崎正俊‡

運輸事業者の多くは，自社の交通事故についての統計を持っている．この統計を利用して各事業者は独自の交通安全対策を講じるが，自社で持っている統計情報だけではデータ件数が少なく的確な対策を講じることが難しい．そこで，複数の事業者間で交通事故データを共有して，より多くのデータからより効果的な交通事故の防止対策を講じることが考えられる．しかし，どの事業者がどのデータを提供したかが他の事業者に知られるシステムでは，事業者にとってデータ提供の意欲を削ぐことになる．そこで本稿では，データ提供者と提供されたデータ間の関連を特定することができない匿名データベースにより，事業者間でそれぞれの持つ交通事故データベースの内容を匿名で共有するシステムを提案する．また，匿名データベースの実装に対する匿名性の考察と性能評価についても示す．

Sharing Traffic Accident Information among Companies with an Anonymous Database

Nobuyuki Tsushima† Eiji Sugino† Yuko Murayama† Masatoshi Miyazaki‡

We have developed a traffic accident information database system and an anonymous database system. Using the former, transportation companies collects and manages traffic accident information to improve their transportation services' safety. The problem is that the amount of information a company can collect is limited. Although benefits of data collection depend on the number of cases collected, it is quite difficult for those companies to share the information due to their privacy. As a solution we propose an anonymous database system, which guarantees users' anonymity so that companies offer its information anonymously and share it with the others.

１．はじめに

平成 14 年中に日本国内で約 94 万件の交通事故が発生した．これら交通事故はタクシーや宅配業者などの運輸事業者にとって多大な金銭的そして時間的な損失をもたらす．さらには負傷や死亡といった取り返しのつかない損失も招くこともある．特に需給バランス調整の廃止などの規制緩和により自由競争が始まったタクシー業界では，交通事故による不要な処理コストを削減することが事業の成功のために重要となる．

交通事故を減らすための手段の一つとして，過去の事故についての情報を収集し，それらを分析することで，社員教育や運行計画の改善に役立たせることが挙げられる．事故の情報は，一般に監督官庁に対する事故報告書といった形で各事業者が持っている．これら紙の事故報告書は一覧性や検索性が悪く，有効な事故対策を立案することが困難だった．そ

†岩手県立大学ソフトウエア情報学研究科
Graduate School of Software and Information Science, Iwate Prefectural University
‡有限会社 DAIS
DAIS CO, LTD

こで，電子的な交通事故データベースを作成することによって，事業者が 交通事故情報を分析することを容易にした．

しかし，事業者一社だけで収集できる件数には限度があるため，的確な改善策を立てることは難しかった．データベースの件数を増やすために，運輸事業者グループ内でそれぞれの持つ事故情報を共有することが考えられるが，事故情報には従業員氏名など外部に公開するべきではない項目があるほか，記名での情報提供は事業者にデータ提供への意欲を削ぐことになりうる．

そこで本稿は，匿名性を確保しつつアクセス制御可能な匿名データベースにより，特定の事業者グループ内で交通事故情報を共有するシステムを提案する．本システムにおける匿名性とは，データとその提供者間の関連を特定不能にすることである．また，アクセス制御とは，許可を与えられた事業者グループ外の第三者によるデータベースの利用を阻止することと，データを挿入した事業者にのみそのデータに対する更新と削除を許可することである．

以下 2 章では事業者がそれぞれ自社の事故情報を蓄積する交通事故データベースとその問題点について述べ，その解決策として 3 章では交通事故情報の企業間共有システムの提案を行う．4 章では提案システムの主要コンポーネントである匿名データベースについて述べ，5 章で暗号通信路，6 章で匿名通信路の実装について述べる．7 章で匿名データベースの匿名性についての考察と性能評価について述べ，8 章で全体の要約と今後の課題について述べる．

2．交通事故データベースとその問題点

交通事故に関する情報は警察庁統計局発表の資料[1]などからも得ることができるが，これらは大まかな情報であり，運輸事業者が自身の状況に合わせた事故対策を立てるうえではこれだけでは足りない．従業者の運転適正にあわせた指導などを行うためには，自社の事故情報を活用する必要がある．そこで，従来紙ベースで管理されていた事故情報を電子的に管理，分析できる交通事故データベースを実装した(図 1).

本システムは交通事故報告書の情報の挿入，それら情報の条件検索と検索結果の表形式で一覧表示，そして電子地図☆上に事故発生場所をプロットする機能を提供する．本システムでは各件について，発生場所住所，発生日時，路面状況，交通量，従業員氏名などの項目が定義されており，事業者の必要に応じて項目の定義の変更をすることも可能である．

事故対策を立てるうえでの交通事故データベースの有用性は，収集された事故件数に大きく依存する．しかし，事業者一社だけで収集できる件数には限度がある．自家用乗用車一台あたりの年間走行距離が約 8,000km なのに対し，事業用乗用車一台あたりは約 61,000km である．また，走行 100 万キロあたりの平均事故発生率では，自家用乗用車一台あたり約 1.20 件，事業用乗用車一台あたりでは約 1.44 件である．つまり，一台あたりの事業用乗用車の年間事故発生確率は自家用自動車の約 9.1 倍である．ただし，確率こそ高いものの，86%の法人タクシー事業者の車両保有台数は 50 台以下である[2]．50 台の車両を運行する事業者では，平均して年間約 4.3 件の事故情報しか収集できない．

図 1　交通事故データベース

☆株式会社 昭文社製 Super Mapple Digital Ver.2 及び MappleX を使用

３．交通事故情報の企業間共有システム

各事業者がそれぞれの事故情報を 1 箇所の共有サーバに登録し，それを事業者グループ内で共有することで，閲覧できる事故情報の件数を増やすことができる．このとき，一般のデータベースシステムを共有サーバとして使用した場合，サーバ管理者はどの情報を誰が登録したかをすべて知ることができる．このようなシステムでも，サーバ管理者が意図的にアクセスログを削除にすることで，管理者および第三者にはデータとその提供者の関連を特定することを防止することができる．ただし，サーバ管理者が悪意を持っている場合や管理ミスにより，管理者および第三者が関連を特定できる情報を漏洩してしまう可能性がある．そこで，管理者の負担とユーザの管理者への不信を解消するためにシステムとして匿名性を確保することが必要である．

そこで，交通事故情報の企業間共有システムでは，システムとしてアクセス制御と匿名性を保証する匿名データベースを共有サーバとして使用する(図 1)．また，事故情報には従業員氏名など外部に公開するべきではない項目があるので，登録時に事業者が登録する項目としない項目を指定できるようにする．これらの機能により事業者に対してデータ提供へのモチベーション向上を図る．

本システムにより各事業者は匿名のうちに共有データベース内にあるすべての情報の検索，閲覧と，自身が挿入した情報について更新と削除が可能になる．

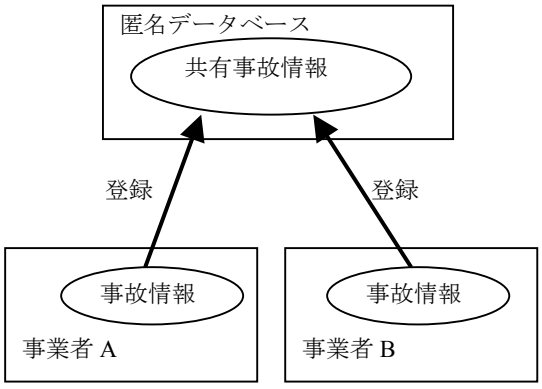

図 2　交通事故情報の匿名共有システム

４．匿名データベース

４．１　関連研究

一般のデータベースシステムの匿名性確保という観点でみると，サーバ管理者はクライアントの IP アドレスやクライアント ID などから，データとその提供者の関連を特定できることが問題になる．これまでに匿名性を確保しながらアクセス制御を可能にするシステムが数多く研究されている．それらの代表格として電子投票が挙げられる．電子投票で提案されている手法は大きく以下の 2 つに分けられる[3].

(1) マルチパーティープロトコルを使用したもの.
(2) 匿名通信路を使用したもの.

前者はゼロ知識証明など，一般には広く用いられていない複雑な暗号を使用したものが多い．そのため，一般に投票のための計算と通信コストが大きい．また，電子投票に特化しているため，各候補者の信任・不信任のいずれかを表す数ビットのみを送信することが前提となっており，データベースのような大量のデータを取り扱うことには向いていない.

後者は，RSA 署名に基づくブラインド署名を使用したものが多い．前者に比べると計算と通信コストが小さく，電子匿名アンケート[4]など電子投票よりも扱うデータが大きいアプリケーションにも応用されている．そのため，本稿における匿名データベースでは後者手法を採用し，大きなデータを高速に処理できることを目指す.

また，電子投票をデータベースシステムの一種と考えると，レコードの挿入のみを提供しているとみなすことができる．しかし，特定グループ内での情報共有では，一般のデータベースシステムのように，クライアントが検索，更新，削除できる機能が必要になる．そこで本システムは匿名性を確保しつつ，それらの機能追加を目指す.

４．２　設計概要

一般のデータベースでは認証とリクエスト処理が同一のサーバで行われるが，本システムではそれぞれ認証サーバとデータサーバに分離されている(図 3)．クライアントは，認証

サーバにて自身の身元を明らかにして，データサーバにアクセスするたびの一種のチケットを要求する．認証サーバはクライアントが権限を与えられた正当な者だと確認できた場合，チケットを発行する．このとき，チケットは第三者に盗まれて悪用されることを防ぐために，暗号通信路を経由して送られる．チケットを入手したクライアントは，今度は自身の身元を匿名通信路により隠して，データサーバに対してチケットを渡す．データサーバはチケットが正しく認証サーバによって発行されたものか検証し，正しければクライアントのリクエストを実行し，不正ならば拒否する．

クライアントのリクエストの種類が検索だった場合，データサーバは検索結果を返し，レコードの挿入ならば，挿入したレコードに対するレシートを発行する．クライアントはこれを秘密に保管する．クライアントのリクエストが更新もしくは削除の場合，データサーバはどのレコードに対して影響があるか調査し，影響のあるレコードに対するレシートの提示をクライアントに要求する．クライアントは自身で保持する対応するレシートをデータサーバに送る．データサーバはレシートを検証して，クライアントが挿入したものだと確認できたレコードについて更新もしくは削除を行う．

クライアントのデータサーバへのすべてのリクエストは送信者匿名性が確保された状態で行われるため，データとその提供者間の特定は不可能になる．また，チケットによりデータサーバの不正なクライアントの利用を防止し，レシートによりレコードを挿入したクライアントのみにそのレコードの更新と削除を許可する．

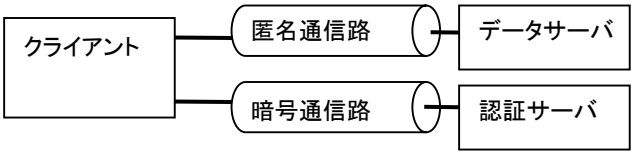

図 3.モジュール構成

４．３　検索

クライアントが検索を実行する場合，まずは SQL で SELECT リクエストを生成し，このハッシュ値 H を MD5 アルゴリズムで算出する．クライアントは秘密の乱数 R を生成し，乱数と認証サーバの公開鍵 e でハッシュ値を暗号化[5]する$(H \cdot R^e \pmod n)$．クライアントはクライアント ID と暗号化したハッシュ値を認証サーバに送る．

認証サーバは受信したクライアント ID に対応するクライアント公開鍵を ACL(Access Control List)から取得し，対になる秘密鍵をクライアントが持っているかをチャレンジ＆レスポンスで検証する．クライアントが正当な場合は，暗号化されたハッシュ値に認証サーバの秘密鍵 d でブラインド署名を施し，クライアントに返送する$((H \cdot R^e)^d \equiv H^d \cdot R^{ed} \equiv H^d \cdot R \pmod n)$．

クライアントは，返送されたものを秘密の乱数 R で復号化し，リクエストに対する署名 H^d を得る$(H^d \cdot R \cdot \equiv H^d \pmod n)$．次にクライアントは平文リクエストとその署名を結合して，4.2 節で述べたチケットにあたるリクエスト証明書を生成する．クライアントはリクエスト証明書をデータサーバに匿名通信路経由で送る．このとき，クライアント ID などのクライアントを特定する情報は送信されず，クライアントの IP アドレスは匿名通信路によりデータサーバから隠蔽される．

データサーバはリクエスト証明書の署名が認証サーバの正当なものか検証し，正しければ平文リクエストを実行して検索結果を返送し，不正ならば実行を拒否する．

認証サーバは，どのクライアントが接続したかはわかるが，リクエストはクライアントの秘密の乱数で暗号化されるために，どんなリクエストなのかは知ることはできない．データサーバは，クライアントのリクエスト内容はわかるが，匿名通信路によりどのクライアントのものかはわからない．これにより，認証サーバとデータサーバが結託した場合でも，リクエストの内容の照合によるクライアントの特定はできない．リクエストの正当性は正しい認証サーバの署名があるかどうかのみで判断する．

４．４　挿入

クライアントがレコードを挿入する場合も，認証サーバからリクエスト署名を受け取り，平文リクエストとともにリクエスト証明書としてデータサーバへ送信する．挿入が成功した場合，データサーバはクライアントに 4.2

節で述べたレシートにあたるレコード所有者証明書を送り返す(図 4)．クライアントはこれを秘密に保持する．レコード所有者証明書は，挿入されたレコードの一意なインデックスとテーブル名とデータベース名を結合したもののハッシュ値に，データサーバの秘密鍵で署名を施したものである．

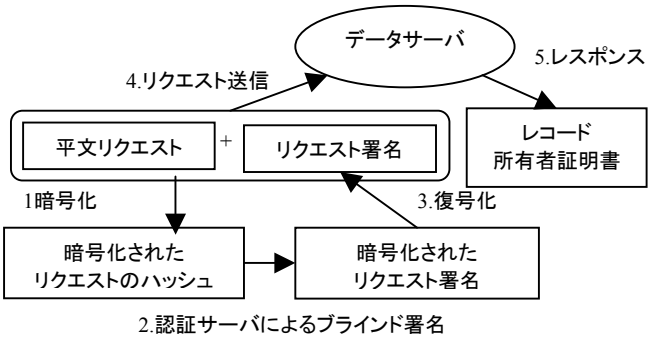

図 4　挿入時のデータの流れ

４．５　更新と削除

クライアントが挿入したデータを更新もしくは削除する場合も，データサーバにリクエスト証明書を送信する．データサーバは更新もしくは削除の影響があるレコードを調査し，それらレコードの ID をクライアントに送信する．クライアントは所持している対応するレコード所有者証明書をデータサーバに送信する．データサーバは受信したレコード所有者証明書を検証し，正当な証明書ならば対応するレコードのみ更新もしくは削除を行う．対応するレコード所有者証明書が送信されなかったレコードとその署名が不正だったレコードに対しては，更新も削除も実行されない．データサーバは，実際に更新もしくは削除されたレコードの ID をクライアントに送り返す．クライアントは実際に削除されたレコードのレコード所有者証明書を削除する．

５．暗号通信路

本システムにおける暗号通信路の役割は，盗聴の防止と認証である．各クライアントおよびサーバ群はそれぞれ 1024bitRSA 暗号の鍵ペアを所持する．これらの鍵を使い，クライアントと認証サーバ間のチャレンジ＆レスポンスによる両者認証と，データストリーム暗号化のための 56bitDES 鍵の共有を行う．今回の実装では暗号ライブラリ AiCrypto[6]利用した．

６．匿名通信路

本システムにおける匿名通信路の役割は，データサーバからクライアントの IP アドレスを隠蔽し，IP アドレスからのクライアントの特定を防ぐことにある．HTTP 用の送信者匿名性を保証する匿名通信路プロトコルとして Crowds[7]があるが，本研究ではこれを一般的な TCP 用に適用した Gunshu を実装した．

Gunshu は匿名化の対象であるイニシエータ，データの最終的な受信者であるレスポンダ，データの転送ノード，転送ノードのリストを管理するマネージャから構成される(図 5)．イニシエータとレスポンダは直接接続しあうことはなく，必ず 1 個以上の転送ノードを経由して通信を行う．

セッション開始時にイニシエータはマネージャから転送ノードのアドレスリストを取得し，その中から転送を依頼するノードをランダムに選択する．イニシエータはこのノードをリストから削除し，かわりにレスポンダのアドレスを挿入したものを次の転送ノードに送信する．リストを受信した転送ノードはリストからランダムに次の接続先ノードを選択し，そのノードをリストから削除したものを，次のノードに送信する．このとき，次の接続先としてレスポンダが選択された場合，経路の確立は完了し，イニシエータとレスポンダはランダムに選択された複数の転送ノードを経由して相互に通信できる．

レスポンダが直接通信する相手は転送ノードであり，イニシエータの IP は入手できない．各転送ノードは一つ前と次のノードの IP アドレスを知ることができるが，前のノードがリクエストを発信したイニシエータなのか，他の転送ノードなのかは判断できない．また，不正な転送ノードにより，イニシエータが意図しないレスポンダに接続されることを防ぐために，サーバのみをチャレンジ＆レスポンスで認証する．さらに，イニシエータとレスポンダ間の通信は，転送ノードによる盗聴を防止するために，レスポンダの公開鍵によって鍵交換された 56bitDES 共有鍵で暗号化される．

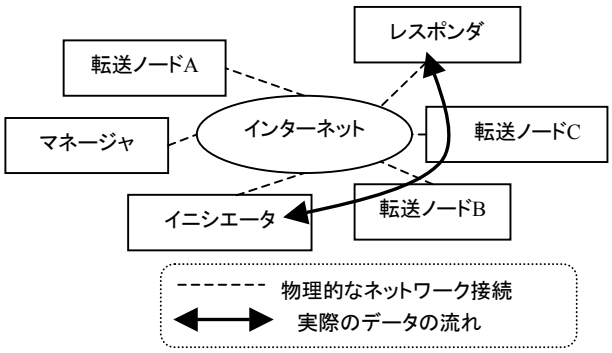

図 5.匿名通信路のノード構成

７．評価と考察

７．１　匿名性の考察

提案システムにおける匿名性とは，データとその提供者間の関連を特定不能にすることである．匿名性に対する脅威は，共有されたデータ解析によるものと，匿名データベースに対する攻撃によるものに大別される．

提案システムでは，従業員名などデータ提供者の特定につながるおそれのある項目は，事業者側で共有しないよう指定できる．しかし，同一地域で営業する事業者間で交通事故情報を共有した場合，事業者が新聞などから得た情報で，データ提供者を推定できる場合がある．事故対策を立てるために，どの項目が必要で，匿名性を確保するためにどの項目が不要なのか今後検討したい．

匿名データベースにおいて，レコードとその提供者間の特定確率はクライアント数に反比例する．しかし，特定の攻撃手法により，レコードとその提供者の関連を高い確率で特定できる場合がある．その一例としてタイミング攻撃[8]が挙げられる．この攻撃手法では，認証サーバとデータサーバが結託し，クライアントが署名を認証サーバに要求した時刻と，データサーバがリクエスト証明書を受信した時刻を比較し，近いものを同一クライアントによるリクエストだと推定する．今後の課題として，匿名データベースに対する脅威の分析とその対策が必要である．

７．２　性能評価

匿名データベースを Microsoft 社の Windows2000 上の C 言語により実装し，その性能評価を行った．実験環境として認証サーバとデータサーバをそれぞれ Intel 社の Celeron 533Mhz を搭載した PC に配置した．クライアントおよび Gunshu 転送ノードとマネージャは同等もしくはそれ以上の速度の CPU を搭載した PC に配置した．すべての PC は 100Mbps のイーサネットで接続されている．

現実装では認証サーバでの署名処理コストが最大のボトルネックになっていると考えられる．認証サーバは毎秒 6.2 回の署名要求で CPU が飽和した．データベースのテーブル定義やレコード件数などの条件によって大きく変わるが，データサーバで複雑な SQL 文を実行したり大量の結果レコードセットがある場合を除けば，データサーバは少なくとも毎秒 6.2 回以上のリクエストを処理できる．つまり，ネットワーク帯域が十分な場合，匿名データベースの処理可能リクエスト数毎秒は，毎秒の認証サーバでのリクエスト証明書発行数と，毎秒のデータサーバでのクエリ実行数のうちどちらか小さい方になる．

暗号通信路および匿名通信路は約 15Mbps で CPU が飽和した．データサーバがクライアントに対して大量の結果レコードセットを送信する場合を除いて，匿名データベースの通信量は小さいため，これら通信路が大きなボトルネックとはならないと考えられる．匿名通信路は転送段数が増えた場合でも，ネットワーク帯域が飽和しない限りスループット減少はほとんど無かったが，遅延は段数に比例して増加した．

８．まとめと今後の課題

運輸事業者がより多くの事故情報からより効果的な事故対策を立てられるように，本稿では事業者間で事故情報を匿名で共有するシステムを提案した．提案システムによって事業者は匿名のうちに，共有された事故情報の検索，自社が提供した事故情報の更新と削除が可能になる．現在，岩手県内のタクシー事業者と交通事故データベースの運用実験を行っている．今後の課題としては，交通事故データベースの運用実験の参加事業者を増やし，提案システムの運用実験を行うことである．

また，提案システムの匿名性を損なういくつかの攻撃手法が発見されたため，その対策とその他安全性についての検討を行う．また，性能上のボトルネックが判明したため，その改善策についても検討する．

参考文献

[1] 警察庁交通局, 平成１４年中の交通事故の発生状況,2003 年 2 月 27 日.

[2] 運輸政策審議会自動車交通部会, タクシーの需給調整規制廃止に向けて必要となる環境整備方策等について(諮問第 16 号), 平成 11 年 4 月 9 日.

[3] Indrajit Ray, Indrakshi Ray, Natarajan Narasimhamurthi, An Anonymous Electronic Voting Protocol for Voting Over The Internet, Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS '01), June 2001.

[4] 横川典子,菊池浩明村井純,電子匿名アンケート機構の設計と実装,マルチメディア通信と分散処理 75-13,1996.

[5] D. Chaum, "Security Without Identification: Transaction Systems to Make Big Brother Obsolete", Communications of the ACM vol. 28 no. 10 pp.1030-1044, October 1985.

[6] 名古屋工業大学 電気情報工学科 磐田研究室, 暗号ライブラリ AiCrypto のページ, http://mars.elcom.nitech.ac.jp/security/

[7] M. K. Reiter and A. D. Rubin, Anonymity loves company: Anonymous Web transactions with Crowds, Comm. of the ACM 42(2) pp. 32-38, 1999.

[8] M. Rennhard, S. Rafaeli, L. Mathy, B. Plattner, and D. Hutchison. An Architecture for an Anonymity Network. In Proceedings of the 6th IEEE Intl. Workshop on Enterprise Security (WET ICE 2001), pp 165-170, June 2001.