M_035

# Distributed Privacy Oblivious Polynomial Evaluation

小瀬木 浩昭† 平原 耕一† 大矢 健太† 折笠 大典* 武田 正之‡
Hiroaki Ozeki Kouichi Hirahara Kenta Ohya Daisuke Orikasa Masayuki Takeda

## 1. はじめに

情報漏洩事件の多発，個人情報保護法の施行などを受け，近年，情報保護への関心が急速に高まっている．

情報の電子化とインターネットの普及に伴い，より個人に特化した効果的なマーケティング戦略を立てる観点から，個人情報を含んだデータベースや，その活用技術であるデータマイニングの重要性が高まっている．その一方で，個人情報の漏洩事件は後を絶たない．問題の本質は，電子化・ネットワーク共有されることにより情報の有用性はより高まるが，同時に，流出によるリスクも増大することにある．

最近，情報セキュリティ技術における秘密関数計算プロトコルを適用し，個人の属性情報を暗号化したまま解析することで，各種の統計情報や属性間の相関関係などの有益な知識を獲得しようとする，Privacy-Preserving Data Mining（プライバシーを保護したデータマイニング）と呼ばれる研究が注目を集めている．その手法は大きく次の3つに分かれる．**(1)** 個人情報の変更や衛生化（sanitization）によって，一般的なデータとして解析する試み．**(2)** 秘密分散やセキュア関数計算（Secure Multiparty Computation[1]）によって，個人情報を秘匿したまま計算する試み．**(3)** データに意図的なランダムノイズを乗せて，個人情報を意味のないものにゆがませてから，統計的な手法を用いて真のデータの分布を復元する試み．**(1)** は最も簡単な手法であるが，個人情報そのものを解析に利用したい要求に答えられない問題がある（例えば，性別で分類したいのに個人情報として衛生化・除去されてしまっている場合など）．**(3)** は，実用的な手法のひとつであるが，ランダムノイズを乗せるために比較的規模の大きなデータベースの存在を前提とする問題，統計的な手法に頼るために正確な値が得られない問題，データの種類により分布が異なり，用いる統計的な手法も異なることから，ランダムノイズの乗せ方を決定するにおいて，元のデータの傾向や性質がある程度分かっていないと決定できないという矛盾の問題がある．**(2)** は，正しい値を得られる手法であり，さらに大きく2つに分類できる．**(2-1)** Secure Function Evaluation などの，予め総当りで計算結果をすべて計算し，計算結果から Oblivious Transfer[2]を用いて，必要な結果だけを取得する手法．**(2-2)** Oblivious Polynomial Evaluation[2],[21]やそれに類似した，本来求めたい真の値に，それと区別できないダミーの値を何個か混ぜて，計算結果から Oblivious Transfer を用いて，必要な真の値の計算結果だけを得る手法．**(2-1)** は理論的には適用範囲の広い秘密関数計算が可能であるが，総当りで結果を出すために効率が極めて悪く，単純な計算の利用に留まるのに対し，**(2-2)** は計算量，通信量，通信回数全てにおいて効率が良く，実用化が期待されているが，個々のプロトコルが，ある特定の適用範囲の小さい問題に特化することで，ある特定の前提状況下で特定の問題に対してだけ有効であるという課題があり，現在その適用範囲の拡充が望まれている．**(2-2)** の手法の中でも，1999 年に Naor.らによって考案された，Oblivious Polynomial Evaluation（紛失多項式評価，以下 OPE）[2],[21]は，その汎用性と効率の良さから現在注目されている．Privacy-Preserving Data Mining については，[4]にその動向がまとめられている．なお，**(1)**～**(3)**の手法は，相互に補完的な手法であり，実際の問題解決においては，その状況に応じて通常複数のアプローチを組み合わせて要求を実現する．特に OPE を用いた例が[4]の 3.2 節で紹介されている．

これまで我々は，Secure Multiparty Computation[1]の要素技術の中でも特に，Oblivious Polynomial Evaluation(OPE)及び Oblivious Transfer(OT)の拡張に関して研究を行ってきた（[5],[6],[7],[8],[9],[10]）．本研究の目的は，効率が良く比較的汎用性の高いといわれている，暗号プロトコルの要素技術である OPE と OT を拡張し，従来の Secure Multiparty Computation では扱い難い問題に適用可能にすることにある．それは直接的には**(2)**で述べた Secure Multiparty Computation の分野の発展に貢献し，間接的には最近急激に重要性が増している Privacy-Preserving Data Mining などのプライバシー重視のデータ活用の研究において，多種多様な要求を安全かつ効率的に実現するためのツールとして活用されることで，その適用範囲の拡充や効率性の向上などの効果をもたらす．

本稿では，1章で研究背景を述べ本研究分野の重要性を明らかにする．2章で，これまで2者間に限定され，秘密関数計算に使用できる多項式が1変数に限定されていた従来の OPE を，初めて，多者間においてそれぞれが入力値である秘密を持つとき，お互いの入力値の秘密を保持したまま，秘密関数計算の結果を得ることが可能な，分散秘密 OPE へと拡張するとともに，その計算例を示す．3章で安全性についての議論を行い，4章で分散秘密 OPE が有用性を持つ具体例として，分散データマイニングの構成例を示す．5章で関連研究を述べ本拡張の新規性と有用性を裏付け，6章で本稿をまとめる．

## 2. 分散秘密 OPE

本章では，多者間においてそれぞれが秘密を持つとき，それぞれの秘密を多変数多項式に代入し結果を得ることが可能な，分散秘密 OPE (OPE on Distributed Privacy, DPOPE)の構成を示す．

### 2.1. 定義

DPOPE は次のような機能を実現する Multi-Party（多者間）プロトコルである．**(1)** ノード $A_1$ は $n$ 変数多項式 $P(y_1,\cdots,y_n)$ を持っている．そして各ノード $A_i(i=1,\cdots,n)$ はそれぞれ秘密情報 $\alpha_i$ を持っている．プロトコルの終了後，ノード $A_n$ は $P(\alpha_1,\cdots,\alpha_n)$ の値を取得する．**(2)** 各ノード $A_2,\cdots,A_n$ は，多項式 $P(y_1,\cdots,y_n)$ に関してまったく分からない．**(3)** 各ノード $A_1,\cdots,A_{n-1}$ は，$P(\alpha_1,\cdots,\alpha_n)$ に関してまったく分からない．**(4)** 各ノード $A_i(i=1,\cdots,n)$ は，それぞれが持つ秘密情報 $\alpha_i$ 以外の秘密情報 $\alpha_j(i\neq j)$ に関してまったくわからない．

### 2.2. 前提条件

$A_1$～$A_n$ はそれぞれが互いに独立し，結託することはないものとする．

### 2.3. プロトコル

以下，プロトコルの流れについて示す．

**(Step 1)** まず $A_1$ 以外の各ノード $A_i(i=2,\cdots,n)$ は公開鍵暗号方式による公開鍵 $E_i$ と秘密鍵 $D_i$ をそれぞれ用意する．

**(Step 2)** $A_1$ は各ノード $A_i(i=2,\cdots,n)$ の公開鍵 $E_i$ を手に入れる．

**(Step 3)** $A_1$ は各ノード $A_i(i=2,\cdots,n)$ に対して，それぞれランダムな値 $\beta_i$ を定め，$P_i(\beta_i)=0$ を満たす 1 変数多項式 $P_i(x_i)$ を生成する．また(Step 2)で手に入れたそれぞれの公開鍵 $E_i$ を使い，$\beta_i$ を暗号化した $E_i(\beta_i)$ を生成する．

**(Step 4)** $A_1$ は暗号化した $\{E_i(\beta_i)\,|\,(i=2,\cdots,n)\}$ を $A_2$ に送信し，$A_2$ から $A_3$，$A_3$ から $A_4$，…，$A_{n-1}$ から $A_n$ と同報していく．

**(Step 5)** 各ノード $A_i(i=2,\cdots,n)$ は受け取った $E_i(\beta_i)$ を秘密鍵 $D_i$ で復号し，$\beta_i$ を手に入れる．そして手に入れた $\beta_i$ を使って，秘密情報 $\alpha_i$ を隠すために，$S_i(\beta_i)=\alpha_i$ を満たす1変数多項式 $S_i(x)$ を定義する．

**(Step 6)** 次に $A_1$ は秘密多項式 $P(y_1,\cdots,y_n)$ を隠すために，各ノードごとにランダムな 1 変数多項式

†東京理科大学大学院 理工学研究科 情報科学専攻，
Graduate School of Science and Technology,
Tokyo University of Science
‡東京理科大学 理工学部 情報科学科，
Dept. of Information Sciences, Tokyo University of Science
*日立製作所 RAID システム事業部，Hitachi, Ltd.

$z_i = p_i(y_i)\ (i=2,\cdots,n)$ を生成し，それぞれの公開鍵 $E_i$ で暗号化し，$\{E_i(p_i(y_i)) \mid (i=2,\cdots,n)\}$ を (Step 4)と同様に同報する．ただし，$p_i : y_i \to z_i$ は単射とする．

**(Step 7)** $A_1$ は(Step 6)で生成した $z_i = p_i(y_i)\ (i=2,\cdots,n)$ を使って，プロトコル実行中に秘密多項式 $P$ が他のノードに推測されないように，

$$P(y_1,\cdots,y_n) = P'(y_1, p_2(y_2),\cdots,p_n(y_n))$$
$$= P'(y_1, z_2,\cdots,z_n)$$

と秘密多項式 $P$ を $P'$ へ置換する．さらに，

$$Q(\beta_2,\cdots,\beta_n,y_1,z_2,\cdots,z_n) = P'(y_1,z_2,\cdots,z_n)$$

を満たす $2n-1$ 変数関数

$$Q(x_2,\cdots,x_n,y_1,z_2,\cdots,z_n)$$
$$= P_2(x_2)+\cdots+P_n(x_n)+P'(y_1,z_2,\cdots,z_n)$$

を生成し，$y_1$ に自分の持つ秘密情報 $\alpha_1$ を代入する．

最終的に $A_n$ が，

$$R(\alpha_1,\beta_2,\cdots,\beta_{n-1},x_n)$$
$$= Q(\beta_2,\cdots,\beta_{n-1},x_n,$$
$$\alpha_1, p_2(\alpha_2),\cdots,p_{n-1}(\alpha_{n-1}),p_n(S_n(x_n)))$$

を知ることができれば，

$$R(\alpha_1,\beta_2,\cdots,\beta_n)$$
$$= Q(\beta_2,\cdots,\beta_n,$$
$$\alpha_1, p_2(\alpha_2),\cdots,p_{n-1}(\alpha_{n-1}),p_n(S_n(\beta_n)))$$
$$= P_2(\beta_2)+\cdots+P_n(\beta_n)$$
$$+P'(\alpha_1,p_2(\alpha_2),\cdots,p_{n-1}(\alpha_{n-1}),p_n(S_n(\beta_n)))$$
$$= P(\alpha_1,\cdots,\alpha_{n-1},S_n(\beta_n))$$
$$= P(\alpha_1,\cdots,\alpha_{n-1},\alpha_n)$$

より，目的の値を得ることができる．

**(Step 8)** $A_1$ が秘密の多項式 $P$ を持ち，$A_2$ が秘密情報 $\alpha_2$ を持つという条件の下で，2 者間における OPE プロトコルを $A_1$ と $A_2$ において実行する．ただし，$A_2$ の入力値は $(a_j, p_2(S(a_j)))\ (j=1,\cdots,m)$ ($a_j$ は乱数)とダミーデータであり，また，多項式 $Q$ における $x_2, z_2$ 以外の変数 $x_3,\cdots,x_n,z_3,\cdots,z_n$ に関しては，ここでは計算せずに変数として文字のまま扱う．よって，$A_2$ は $Q(\beta_2,x_3,\cdots,x_n,\alpha_1,p_2(S_2(\beta_2)),z_3,\cdots,z_n)$ を得る．

**(Step 9)** 同様に，$A_2$ が(Step 8)で得た多項式を，$A_2$ が持つ秘密多項式として，$A_3$ が秘密情報 $\alpha_3$ を持つという条件の下で 2 者間における OPE プロトコルを実行する．

一般的に記述すると，$A_i\ (i=1,\cdots,n-1)$ は秘密多項式

$$Q(\beta_2,\cdots,\beta_i,x_{i+1},\cdots,x_n,$$
$$\alpha_1,p_2(S_2(\beta_2)),\cdots,p_i(S_i(\beta_i)),z_{i+1},\cdots,z_n)$$

を持ち，$A_{i+1}$ は秘密情報 $\alpha_{i+1}$ を持っている．この条件の下で，2 者間 OPE プロトコルを実行し，$A_{i+1}$ は

$$Q(\beta_2,\cdots,\beta_{i+1},x_{i+2},\cdots,x_n,$$
$$\alpha_1,p_2(S_2(\beta_2)),\cdots,p_{i+1}(S_{i+1}(\beta_{i+1})),z_{i+2},\cdots,z_n)$$

を得る．この処理を $A_{n-1}$ と $A_n$ 間で完了するまで行う．

**(Step 10)**(Step 9)の処理が $A_{n-1}$ と $A_n$ の間で完了すると，最終的に $A_n$ は目的の値 $P(\alpha_1,\cdots,\alpha_n)$ を得ることができる．

以上が DPOPE のプロトコルである．

### 2.4. 計算例

今，$A_1,A_2,A_3,A_4$ の 4 人の party が DPOPE を使って秘密計算をする．ここで，$A_1$ は秘密多項式

$$P(y_1,y_2,y_3,y_4) = y_1^2+2y_1y_4+y_2y_3-3y_2+2y_3^2+5y_4+1$$

を持っている．また各 party はそれぞれ秘密情報

$$\alpha_1=3,\alpha_2=-1,\alpha_3=2,\alpha_4=-2$$

を持つ．ノード $A_1$ は自分の情報を $A_2,A_3,A_4$ に知られることなく，

$$P(\alpha_1,\alpha_2,\alpha_3,\alpha_4) = P(3,-1,2,-2)$$

の値を取得したい．この要求を DPOPE プロトコルを使用して実現する．

**(Step 1)** $A_2,A_3,A_4$ は公開鍵 $E_2,E_3,E_4$ と秘密鍵 $D_2,D_3,D_4$ をそれぞれ用意する．

**(Step 2)** $A_1$ は $E_2,E_3,E_4$ を手に入れる．

**(Step 3)** $A_1$ は $A_2,A_3,A_4$ に対して，ランダムな値

$$\beta_2=1,\beta_3=2,\beta_4=-1$$

を定め，それぞれに対して 1 変数多項式

$$P_2(x_2) = x_2^2-17x_2+16\ (s.t.\ P_2(\beta_2)=P_2(1)=0)$$
$$P_3(x_3) = 3x_3^2-x_3-14\ (s.t.\ P_3(\beta_3)=P_3(2)=0)$$
$$P_4(x_4) = 5x_4^2-2x_4-3\ (s.t.\ P_4(\beta_4)=P_4(-1)=0)$$

を生成する．また，ランダムな値 $\beta_2,\beta_3,\beta_4$ に関しては，それぞれを暗号化する．

**(Step 4)** $A_1$ は暗号化した $E_2(\beta_2),E_3(\beta_3),E_4(\beta_4)$ を $A_2$ から各 party に同報する．

**(Step 5)** $A_2,A_3,A_4$ は受け取った $\beta_2,\beta_3,\beta_4$ を使って，1 変数多項式

$$S_2(x_2) = x_2^2-3x_2+1\ (s.t.\ S_2(\beta_2)=\alpha_2 \Rightarrow S_2(1)=-1)$$
$$S_3(x_3) = -3x_3+8\ (s.t.\ S_3(\beta_3)=\alpha_3 \Rightarrow S_3(2)=2)$$
$$S_4(x_4) = -2x_4^2+x_4+1\ (s.t.\ S_4(\beta_4)=\alpha_4 \Rightarrow S_4(-1)=-2)$$

を生成する．

**(Step 6)** $A_1$ は $A_2,A_3,A_4$ に対して，多項式 $P$ を置換する写像

$$z_2 = f_2(y_2) = (y_2-1)/3 \Leftrightarrow y_2 = f_2^{-1}(z_2) = 3z_2+1$$
$$z_3 = f_3(y_3) = -y_3+3 \Leftrightarrow y_3 = f_3^{-1}(z_3) = -z_3+3$$
$$z_4 = f_4(y_4) = (y_4+7)/5 \Leftrightarrow y_4 = f_4^{-1}(z_4) = 5z_4+7$$

を生成する．

**(Step 7)** $A_1$ は(Step 6)で生成した写像 $f_2,f_3,f_4$ を使って，

$$P(y_1,y_2,y_3,y_4) = P'(y_1,f_2(y_2),f_3(y_3),f_4(y_4))$$
$$= P'(y_1,z_2,z_3,z_4)$$
$$= y_1^2+2y_1(5z_4-7)+(3z_2+1)(-z_3+3)-3(3z_2+1)$$
$$+2(-z_3+3)^2+5(5z_4-7)+1$$
$$= y_1^2+10y_1z_4-14y_1-3z_2z_3+2z_2^2-13z_3+25z_4-16$$

へ置換する．

また，$A_1$ は多項式 $P$ と(Step 3)で生成した多項式 $P_2(x_2),P_3(x_3),P_4(x_4)$ の合成関数

$$
\begin{aligned}
&Q(x_2,x_3,x_4,y_1,z_2,z_3,z_4)\\
&=P_2(x_2)+P_3(x_3)+P_4(x_4)+P'(y_1,z_2,z_3,z_4)\\
&=(x_2^2-17x_2+16)+(3x_3^2+x_3-14)+(5x_4^2+2x_4-3)\\
&\quad+(y_1^2+10y_1z_4-14y_1-3z_2z_3+2z_3^2-13z_3+25z_4-16)\\
&=x_2^2-17x_2+3x_3^2+x_3+5x_4^2+2x_4+y_1^2+10y_1z_4\\
&\quad-14y_1-3z_2z_3+2z_3^2-13z_3+25z_4-17
\end{aligned}
$$

を生成し，$y_1$ に $\alpha_1=3$ を代入する．

$$
\begin{aligned}
&Q(x_2,x_3,x_4,\alpha_1,z_2,z_3,z_4)\\
&=x_2^2-17x_2+3x_3^2+x_3+5x_4^2+2x_4+9+30z_4-42\\
&\quad-3z_2z_3+2z_3^2-13z_3+25z_4-17)\\
&=x_2^2-17x_2+3x_3^2+x_3+5x_4^2+2x_4-3z_2z_3+2z_3^2-13z_3\\
&\quad+55z_4-50
\end{aligned}
$$

**(Step 8)** $A_1$ が秘密多項式 $P(y_1,y_2,y_3,y_4)$ を持ち，$A_2$ が秘密情報 $\alpha_2=-1$ を持つという条件の下で，2 者間 OPE プロトコルを実行する．プロトコルの実行後 $A_2$ は

$$
\begin{aligned}
&Q(\beta_2,x_3,x_4,\alpha_1,f_2(S_2(\beta_2)),z_3,z_4)\\
&=Q(\beta_2,x_3,x_4,\alpha_1,f_2(\alpha_2),z_3,z_4)\\
&=3x_3^2+x_3+5x_4^2+2x_4+2z_3^2-11z_3+55z_4-66
\end{aligned}
$$

を得る．

**(Step 9)** 同様に $A_2,A_3$ 間で 2 者間 OPE を実行すると，$A_3$ は

$$
\begin{aligned}
&Q(\beta_2,\beta_3,x_4,\alpha_1,f_2(\alpha_2),f_3(S_3(\beta_3)),z_4)\\
&=Q(\beta_2,\beta_3,x_4,\alpha_1,f_2(\alpha_2),f_3(\alpha_3),z_4)\\
&=5x_4^2+2x_4+55z_4-61
\end{aligned}
$$

を得る．

**(Step 10)** 最後に $A_3,A_4$ 間で 2 者間 OPE を実行すると，$A_4$ は

$$
\begin{aligned}
&Q(\beta_2,\beta_3,\beta_4,\alpha_1,f_2(\alpha_2),f_3(\alpha_3),f_4(S_4(\beta_4)))\\
&=Q(\beta_2,\beta_3,\beta_4,\alpha_1,f_2(\alpha_2),f_3(\alpha_3),f_4(\alpha_4))\\
&=P_2(\beta_2)+P_3(\beta_3)+P_4(\beta_4)+P'(\alpha_1,f_2(\alpha_2),f_3(\alpha_3),f_4(\alpha_4))\\
&=P(\alpha_1,\alpha_2,\alpha_3,\alpha_4)\\
&=-3
\end{aligned}
$$

を得る．

## 3. 分散秘密 OPE の安全性についての議論

従来 OPE の安全性については，[21]において，安全となるパラメータの取り方について議論されている．ここでは分散関数 OPE に拡張した際に新たに検討する余地のある部分についてだけ議論する．

### 3.1. 秘匿性

DPOPE の安全性について検討する．各ノード $A_i(i=2,\cdots,n)$ が得る情報は，

$$
\begin{aligned}
&Q(\beta_2,\cdots,\beta_i,x_{i+1},\cdots,x_n,\alpha_1,\cdots,\alpha_i,z_{i+1},\cdots,z_n)\\
&=P_{i+1}(x_{i+1})+\cdots+P_n(x_n)\\
&\quad+P'(\alpha_1,p_2(\alpha_2),\cdots,p_i(\alpha_i),z_{i+1},\cdots,z_n)
\end{aligned}
$$

の多項式であり，この情報から以下の項目の秘匿性を満たすことが可能であるか考察する．ここでは，$1\le h<i<j\le n$ とする．

**多項式 $P$ の秘匿性**：$z_j=p_j(y_j)$ により多項式 $P$ は $P'$ へ置換されているが，$A_i$ はその関係を知らないため，$P'(\alpha_1,p_2(\alpha_2),\cdots,p_i(\alpha_i),z_{i+1},\cdots,z_n)$ から元の多項式 $P$ を求めることはできない．**秘密情報 $\alpha_h$ の秘匿性**：$\alpha_h$ は，すでに多項式 $P$ によって処理された値のため，多項式 $P$ を知ることができなければ求めることができない．**途中結果の秘匿性**：$z_j=p_j(y_j)$ により多項式 $P$ は置換されているので，$P$ の計算途中の結果を知ることができない．さらに，$p_j(x_j)$ によって定数項がランダム化されているので，計算途中の正確な値を求めることができない．

### 3.2. ノード同士の悪意ある結託について

一般に Secure Multiparty Computation においては，いくつかの構成ノードの悪意ある結託により秘密が漏えいする場合が考え得るが，結託に対する耐性の強度と，プロトコル自体の，計算量，通信量，通信回数などの効率性とはトレードオフの関係にあり，本稿では各ノード同士は結託をしないという前提で議論を進めた．悪意ある結託に対する耐性や防止の手法は別の技術で解決する方法もあるが，効率性を保ちつつ結託対策を可能とする最適な手法の選定は今後の課題としたい．

## 4. 分散データマイニングの構成

OPE プロトコルを用いた，プライバシーを保護したデータマイニングについては，すでに Lindel と Pinkas らによって提案されている[11],[12]．ただしこの方式では，2つのグループを作成しそれぞれが代表を選び，最終的に代表2者の間で従来の 2 者間 OPE プロトコルを使用するため，計算負荷が秘密計算を行う代表の2者に集中するという問題があった．そこで，DPOPE を用いた，多者間におけるプライバシー保護データマイニングの構成例を示す．この構成例では，それぞれの構成ノードが等分に DPOPE を行うため，計算負荷が分散されるという利点がある．なお，構成例では，データマイニング手法としては[11],[12]と同様に，決定木を使った手法で相関ルールを導き出す．

### 4.1. 決定木学習アルゴリズム(ID3)

決定木学習アルゴリズムである ID3[13],[14]では，対象データを分類する属性の識別力を評価する方法として情報理論に基づいた手法を用いている．学習データを $T$，$t$ 個の値から成る目的属性を $C=\{c_1,\cdots,c_t\}$ とする．$|T(c_k)|$ を学習データ $T$ において $c_k$ を含むデータ数とする．すなわち，$|T|=\sum_{k=1}^{t}|T(c_k)|$ が成り立つ．ここで学習データ $T$ における目的属性 $C$ によって得られるエントロピー(平均情報量)は，$H_C(T)=\sum_{k=1}^{t}\{-\frac{|T(c_k)|}{|T|}\log\frac{|T(c_k)|}{|T|}\}$ で表される．

また，識別に用いる条件属性を $A=\{a_1,\cdots,a_m\}$ とすると，$a_j$ の条件の下で，学習データ $T$ における目的属性 $C$ のエントロピーの総和は，$H_C(T\mid A)=\sum_{j=1}^{m}\{\frac{|T(a_j)|}{|T|}H_C(T(a_j))\}$ となり，条件属性 $A$ による識別で期待されるエントロピーの削減量を表すゲイン(情報利得)は，$Gain_C(A)=H_C(T)-H_C(T\mid A)$ で定義される値で表される．

### 4.2. データマイニングの方針

データマイニングに参加するノード $A_1,\cdots,A_n$ はそれぞれ学習データ $T_1,\cdots,T_n$ を持っている $(T=T_1+\cdots+T_n)$．これらの学習データをお互いに秘匿にしながら，ID3 アルゴリズムを用いて，ゲイン $Gain_C(A)$ を最大とする条件属性 $A$ を求めることが目的である．ここで，$H_C(T)$ は調べていく条件属性 $A$ と独立であるため，

$$
\begin{aligned}
&H_C(T\mid A)=\sum_{j=1}^{m}\{\frac{|T(a_j)|}{|T|}H_C(T(a_j))\}\\
&=\frac{1}{|T|}\sum_{j=1}^{m}|T(a_j)|\sum_{k=1}^{t}\{-\frac{|T(a_j,c_k)|}{|T(a_j)|}\log(\frac{|T(a_j,c_k)|}{|T(a_j)|})\}\\
&=\frac{1}{|T|}[-\sum_{j=1}^{m}\{\sum_{k=1}^{t}|T(a_j,c_k)|\log(|T(a_j,c_k)|)\}+\sum_{j=1}^{m}|T(a_j)|\log(|T(a_j)|)]
\end{aligned}
$$

が最小となる場合を考えればよい．つまり，複数のノード $A_1,\cdots,A_n$ がそれぞれ秘密情報 $\alpha_1,\cdots,\alpha_n$ を持っていて，お互いの入力値 $\alpha_i(i=1,\cdots,n)$ を秘密にしながら，$P(\alpha_1+\cdots+\alpha_n)=(\alpha_1+\cdots+\alpha_n)\log(\alpha_1+\cdots+\alpha_n)$ を計算す

る問題と考えることができる(例えば，$\alpha_i = |T_i(a_j)|$)．

### 4.3. データマイニングの流れ

DPOPE を使って，多者間におけるプライバシーを保護したデータマイニングを構成する．まず，秘密情報 $\alpha_1,\cdots,\alpha_n$ をそれぞれ持つ各ノードは協力して DPOPE を実行する．実行中に OPE を使ったランダムシェアの計算法[11]を使用し，$x=\alpha_1+\cdots+\alpha_n=v_1+\cdots+v_n$ を満たすランダムシェア $v_1,\cdots,v_n$ をそれぞれ求める．また同様に，$\log x=\log(\alpha_1+\cdots+\alpha_n)=l_1+\cdots+l_n$ を満たすランダムシェア $l_1,\cdots,l_n$ をそれぞれ求める．これらのランダムシェアを使って，$u_1+\cdots+u_n=x\log x$ を満たすランダムシェア $u_1,\cdots,u_n$ を求めることを考える．

$$\begin{aligned}x\log x&=(v_1+\cdots+v_n)(l_1+\cdots+l_n)\\&=v_1l_1+v_1l_2+\cdots+v_1l_n+v_2l_1+v_2l_2+\cdots+v_2l_n\\&\quad+\cdots+v_nl_1+v_nl_2+\cdots+v_nl_n\end{aligned}$$

より，各ノード $A_i$ は $v_il_i$ をそれぞれ独自に計算することができる．$v_il_j$ については $A_i$ と $A_j$ において OPE を使ったランダムシェアの計算法を使って，$r^i_{v_il_j}+r^j_{v_il_j}=v_il_j$ となるランダムシェアを求め，$A_i$ は $r^i_{v_il_j}$ を，$A_j$ は $r^j_{v_il_j}$ を得る．全てのノードとのランダムシェアを求めたら，独自で計算した $v_il_i$ とランダムシェアを使って，$u_i=r^i_{v_il_1}+\cdots+r^i_{v_il_{i-1}}+v_il_i+r^i_{v_il_{i+1}}+\cdots+r^i_{v_il_n}$ とする．これらの値を全てのノードが公開することで，$u_1+\cdots+u_n=x\log x$ を計算する．

以上の処理を繰り返し実行することで，学習データ $T_i$ に関する情報を一切漏らさずに，全ての条件属性 $A$ についてのゲインを求め，決定木を作成することができる．

## 5. 関連研究

1999 年に Naor らによって提案された Oblivious Polynomial Evaluation は，Oblivious Transfer を基礎プロトコルとし，1 変数多項式関数という比較的幅広い問題を安全に秘密関数計算できるという汎用性の高さと，その効率性の良さにおいて現在注目されている，比較的新しいプロトコルである．これまで OPE を題材とした研究がいくつか存在するが，大きく，OPE そのものを改良する基礎研究と，OPE をツールとして用いた応用研究が存在する．

OPE そのものを改良する基礎研究として，既存の研究では主に次のものが挙げられる．[3]では，検証可能な紛失多項式評価の構成について提案している．提案手法は，OPE を拡張し，両者があらかじめ入力値をコミットしておき，OPE への入力値がコミット値と同一であることを，互いの入力値を相手に漏らさずに両者が検証可能なプロトコルである．[15]では，情報量的に安全な OPE を提案し，これに基づく電子投票方式の構成法を提案している．提案手法は，攻撃者の計算能力／記憶能力などに一切の仮定をおかずに安全性を保証できる OPE である．[16]では，OPE の効率の改善策について述べている．[17]では，OPE の多項式を浮動小数点の数を扱えるように拡張している．

また，OPE をツールとして用いた応用研究は，現在多数が存在し，特に Privacy-Preserving Data Mining の分野では，対象とする問題の解決において，全体の中の部分的な問題について OPE を用いることで効率的に解決するなど，全体の目的を遂げるための要素技術として OPE が頻繁に用いられている．ここでは，OPE の応用研究の中でも比較的 OPE に対する重点が大きいものについてその一部を紹介する．[18]で，検索サービスにおいて，検索サービスの提供者が何も得られずに，利用者が検索結果を得られる手法を提案し，その中で，OPE を基にしたプロトコルについて述べている．[19]では OPE を基にしたプライバシーを保護したクラスタリングを実現する手法について提案している．[20]では，OPE を利用した，非対称不正者追跡機能と不正の自己防止力を付加したコンテンツ配信法について述べている．

本稿で述べた分散秘密 OPE は，OPE そのものを改良する基礎研究に位置し，秘密関数計算を必要とする今後の様々な応用研究への活用が期待できる．

## 6. まとめ

本稿では，これまで2者間に限定され，秘密関数計算に使用できる多項式が 1 変数に限定されていた従来の OPE を，初めて，多者間においてそれぞれが入力値である秘密を持つとき，お互いの入力値の秘密を保持したまま，秘密関数計算の結果を得ることが可能な，分散秘密 OPE へと拡張した．また，分散秘密 OPE が有用性を持つ具体例として，分散データマイニングの構成を示した．

冒頭で述べたように，OPE は Privacy-Preserving Data Mining の要素技術として用いられることも多く，今回拡張した分散秘密 OPE も，今後，プライバシー重視のデータ活用などの応用分野への幅広い貢献が期待できる．

今後の課題として，OPE のさらなる拡張と適用範囲の拡充，分散秘密 OPE の特長を活かしたプライバシー重視のデータ活用への応用，などがあり，安全かつ実用的なプライバシー保護データマイニング手法の構築に向けさらに取り組んで行きたい．

## 参考文献


[1] http://www.cs.ut.ee/~lipmaa/crypto/link/mpc/

[2] Moni Naor, Benny Pinkas: Oblivious transfer and polynomial evaluation, Proc. of the 31st Symp. on Theory of Computer Science (STOC'99), pp.245-254 (1999).

[3] 駒木 寛隆, 渡邊 裕治, 花岡 悟一郎, 今井 秀樹: 検証可能な紛失多項式評価, SCIS2001, pp.471-476 (2001).

[4] 菊池 浩明: データマイニングと個人情報保護, FIT2004, プレミアワークショップ: ユビキタス・モバイルネットワークとセキュリティ, 招待講演 4 (2004).

[5] 小瀬木 浩昭, 折笠 大典, 鎌田 浩嗣, 大矢 健太, 須合 太一, 武田正之: 顧客データと事業者側アルゴリズムの保護を両立するホスティング型情報埋め込みサービス提供モデル, データベースと Web 情報システムに関するシンポジウム 2005(DBWeb2005), pp.81-86 (Nov. 2005).

[6] 折笠 大典, 小瀬木 浩昭, 武田 正之: 顧客データと事業者側アルゴリズムの保護を両立するホスティング型サービス提供モデル, コンピュータセキュリティシンポジウム 2005(CSS2005), 5B-5, pp.367-372 (Oct. 2005).

[7] 須合 太一, 小瀬木 浩昭, 武田 正之: 多者間紛失多項式評価手法の提案とプライバシ保護データマイニングへの適用, 暗号と情報セキュリティシンポジウム 2006(SCIS2006), 3F2-4, p.217 (Jan. 2006).

[8] 平原 耕一, 折笠 大典, 小瀬木 浩昭, 武田 正之: 紛失多項式評価の拡張と安全な情報埋め込みサービスの一構成, 情報処理学会第 68 回全国大会, 7V-11 (Mar.2006).

[9] 鎌田浩嗣, 小瀬木浩昭, 大矢健太, 武田正之: 重み付き Oblivious Transfer の提案と電子コンテンツサービスへの応用, データベースと Web 情報システムに関するシンポジウム 2005(DBWeb2005), pp.87-92 (Nov. 2005).(学生研究奨励賞受賞)

[10] 鎌田浩嗣, 小瀬木浩昭, 武田正之:重み付き Oblivious Transfer, コンピュータセキュリティシンポジウム 2005(CSS2005), 5B-1, pp.343-348 (Oct. 2005).

[11] Y.Lindell and B.Pinkas: Privacy Preserving Data Mining, Journal of Cryptology, Vol.15, No.19, pp.177-106 (2002).

[12] B.Pinkas: Cryptographic techiniques for privacy-preserving data mining, Trusted Systems Laboratory HP Laboratories Palo Alto, HPL-2003-22 (Jan. 2003).

[13] Quinlan, J.R.: Induction of Decision Threes, Machine Learning, Vol.1, pp.81-106 (1986).

[14] Tom Mitchell: Decision Tree Learning, Machine Learning, McGraw-Hill, pp.52-79 (1997).

[15] 大塚 玲, Anderson C.A. Nasciment, 今井 秀樹: 情報量的に安全な秘密多項式評価法と電子投票への応用, 情報処理学会研究報告, CSEC, Vol.2004, No.75, pp.351—358 (July. 2004).

[16] G. Hanaoka, H. Imai, J. Mueller-Quade, A. Nascimento, A. Otsuka, A. Winter: Information Theoretically Secure Oblivious Polynomial Evaluation: Model, Bounds, and Constructions , 9th Australasian Conference, ACISP, LNCS (2004).

[17] Yan-Cheng Chang, Chi-Jen Lu: Oblivious Polynomial Evaluation and Oblivious Neural Learning, Advances in Cryptology, Asiacrypt '01, Lecture Notes in Computer Science Vol.2248, pp. 369-384 (2001).

[18] Wakaha Ogata, Kaoru Kurosawa: Oblivious Keyword Search, Journal of Complexity, Vol.20, pp. 356-371 (2004).

[19] S. Jha, L. Kruger, P. McDaniel: Privacy Preserving Clustering, 10th European Symposium On Research In Computer Security (ESORICS) (2005).

[20] 光成 滋生, 渡辺 秀行, 古田 真紀, 境 隆一, 笠原 正雄: 楕円曲線上のペアリングを用いた不正者追跡法の拡張, コンピュータセキュリティ (CSEC), 18-38, pp. 261-266 (2002.7.19).

[21] D. Bleichenbacher and P. Q. Nquyen, "Noisy Polynomial Interpolation and Noisy Chinese Remaindering,"EUROCRYPT 2000, LNCS 1807, pp. 53-69 (2000).