M_033

Distributed Function 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 を用いて，必要な結果だけを取得する手法．**(2-2)** Oblivious Polynomial Evaluation 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],[3]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．

†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.

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章で基礎知識を述べた後，3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回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関数を持つとき，その合成関数を秘密関数計算した結果を得ることが可能な，分散関数 OPE へと拡張するとともに，計算例を示す．4章で安全性についての議論を行い，5章で分散関数OPEが有用性を持つ具体例として，安全なデータ加工サービスの構成を示す．6章で関連研究を述べ本拡張の新規性と有用性を裏付け，7章で本稿をまとめる．

2. 基礎知識

2.1. Oblivious Transfer [2],[11],[12]

Oblivious Transfer（紛失通信，以下，OT）は2者間のプロトコルであり，OT を利用すると汎用的なマルチパーティ・プロトコルを実現できることが知られている．OT については，[11]，[12]において最新の研究動向を踏まえて紹介されている．k-out-of-N Oblivious Transfer (OT) では，Bob は N 個の秘密 $m_1,m_2,\cdots,m_N$ を，Alice は $k(\leq N)$ 個の秘密 $a_1,a_2,\cdots,a_k(a_i\in \mathrm{N}, i=1,\cdots,k)$ を持っており，プロトコル終了後，Alice は $m_{a_1},m_{a_2},\cdots,m_{a_k}$ を取得する．その際，(1)Alice は，$m_{a_1},m_{a_2},\cdots,m_{a_k}$ 以外についてまったく分からない，(2)Bob は，$a_1,a_2,\cdots,a_k$ についてまったく分からない，という2つの要件を満たす．

2.2. Oblivious Polynomial Evaluation [2],[19]

OPE は2者間のプロトコルで，Alice は定数 α を，Bob は1変数多項式 $P(x)$ を持っており，プロトコル終了後，Alice は $P(\alpha)$ を取得する．その際，(1)Alice は，$P(x)$ のひとつの値 $P(\alpha)$ だけを得ることができる，(2)Bob は，α と $P(\alpha)$ についてまったく分からない，という2つの要件を満たす．次に，OPE のプロトコルについて述べる．**(Step 1) 両者の秘密を定義する**：Bob の秘密にしたい 1 変数多項式は，$P(x)=\sum_{i=0}^{d_P} a_i x^i$ で定義される．また，Alice の秘密にしたい値として α を定義する．**(Step 2) Bob は，2変数多項式の中に P を隠す**：Bob は d 次のランダムな多項式 $P'(x)=\sum_{i=1}^{d} b_i x^i (s.t. P'(0)=0)$ を生成する $(d=d_P * K)$．ここでセキュリティ定数を $K(\in \mathrm{N})$ とする．セキュリティ定数とは，Bob が任意に定める自然数で，セキュリティ定数が大きいほど P' の次数が高くなり，P の推測をより困難とするパラメータである．Bobは，2変数多項式を以下のように定義する．$Q(x,y)=P'(x)+P(y)$．2変数多項式 Q は全ての y において，$Q(0,y)=P(y)$ となる．**(Step 3) Alice は α を1変数多項式 S の中に隠す**：Alice はランダムに K 次の多項式 $S(x)(s.t. S(0)=\alpha)$ を生成する．Alice は $R(x)=Q(x,S(x))$ を用いて，$P(\alpha)$ を得ようとする．$R(0)=Q(0,S(0))=P(S(0))=P(\alpha)$ として求める．**(Step 4) Alice は Bob に値を送信**：$d_R=d=d_P*K$ と定義する．Alice は (d_R+1) 個のデータ $(x_i,S(x_i))$ を作成し，ダミーデータ (x'_j,S'_j) と混ぜて，Bob に送信する．**(Step 5) Bob は受け取ったデータを処理する**：Bob は，(Step 4)で Alice から送られたデータを計算し，$Q(x_i,S(x_i))$ を生成する．**(Step 6) Alice は Bob からデータを受け取り $R(x)$ を再構築する**：Bob が(Step 5)で処理したデータの中から，Alice は，(d_R+1)-out-of-N OT（N は，(d_R+1)+ダミーデータの数)を用

い，(Step 4)で Bob に送った x_i に対応する，$Q(x_i, S(x_i))$ を取得する．そこから $R(x)$（次数は d_R）を再構築し，$R(0) = P(\alpha)$ を得る．以上が OPE のプロトコルである．

3. 分散関数 OPE

本章では，多者間においてそれぞれが多項式関数を持つとき，その合成関数を処理した結果を得ることが可能な，分散関数 OPE (OPE on Distributed Function, DFOPE)を示す．なお，図1はその概念図である．

3.1. 定義

(1) ノード A:秘密にしたい α を持つ．
ノード群 B:ノード B_1 〜ノード B_n までの主体で，それぞれが，多項式を1つずつ持っている．
データの経路は，ノード $A \rightarrow$ ノード群 B（ノード $B_1 \rightarrow$ ノード $B_2 \rightarrow \cdots \rightarrow$ ノード B_n）とする．

(2) ノード A は定数 α を持っている．そして α を，他の誰にも知られずに $P_n \circ P_{n-1} \circ \cdots \circ P_2 \circ P_1(\alpha)$ を得たいとする．

(3) ノード群 B は，それぞれ1変数 m 次多項式 $P_1(x), P_2(x), \cdots, P_n(x)$ を持っている．多項式について処理をする過程で，$P_1(x), P_2(x), \cdots, P_n(x)$ を，自分以外の群 B のノードと A に知られたくないとする．

3.2. 前提条件

ノード A とノード群 B（ノード B_1 〜ノード B_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．

3.3. プロトコル

(Step 1) ノード A，ノード群 B の秘密を定義する:
ノード群 B の秘密にしたいそれぞれの1変数多項式は，

$$\begin{cases} P_1(x) = \sum_{i=0}^{d_{P_1}} a_{1i}x^i \\ P_2(x) = \sum_{i=0}^{d_{P_2}} a_{2i}x^i \\ \vdots \\ P_n(x) = \sum_{i=0}^{d_{P_n}} a_{ni}x^i \end{cases}$$

で定義される．
また，ノード A の秘密にしたい値として α を定義する．

(Step 2) ノード群 B は，それぞれ2変数多項式の中に $P_1(x), P_2(x), \cdots, P_n(x)$ を隠す:
ノードはそれぞれ d_i 次 $(i = 1,2,\cdots,n)$ のランダムな多項式

$$\begin{cases} P_1'(x) = \sum_{i=1}^{d_1} b_{1i}x^i \\ P_2'(x) = \sum_{i=1}^{d_2} b_{2i}x^i \\ \vdots \\ P_n'(x) = \sum_{i=1}^{d_n} b_{ni}x^i \\ (s.t.\ P_j'(0) = 0, j = 1,2,\cdots,n) \end{cases}$$

を生成する(ただし，$d_i = d_{P_i} * k_i$，($\prod_{i=1}^{n} k_i = K$ は，セキュリティ定数)，$i = 1,2,\cdots,n$)．
ノード群 B は，それぞれ P を隠した2変数多項式を以下のように定義する．

$$Q_i(x,y) = P_i'(x) + P_i(y)\ (i = 1,2,\cdots,n)$$

(Step 3) ノード A は α を多項式 S の中に隠す:
ノード A はランダムに K 次の多項式

$$S(x)\ (s.t.\ S(0) = \alpha)$$

を生成する．

(Step 4) ノード A はノード B_1 に値を送る:
ノード A は $(d_R + 1)$ 個のデータ $(x_i, S(x_i))$ を作成し，t_0 個のダミーデータ $(x_j', S_j')(j = 1,\cdots,t_o)$ と混ぜて，ノード B_1 に送る．

(Step 5) ノード B_1 は，送られてきたデータを処理する:
(Step 4)でノード A から送られたデータを自分の持つ多項式 $Q_1(x,y) = P_1'(x) + P_1(y)$ の入力値として，計算する．そして，出力された $(x_i, Q_1(x_i, S(x_i)))$ に，新たに t_1 個のダミーデータ $(x_j'', Q_{1j}'')(j = 1,\cdots,t_1)$ を混ぜてノード B_2 に渡す．

(Step 6) B_2 〜 B_n までの処理:
ノード B_2 は，(Step 5)と同様に，ノード B_1 から受け取った $(x_i, Q_1(x_i, S(x_i)))$ を，自分の持つ多項式の入力値として計算し，計算結果を新たに t_2 個のダミーデータ $(x_j''', Q_{2j}''')(j = 1,\cdots,t_2)$ と混ぜて，次のノードに渡す．このようにして，ノード B_n まで計算すると，

$$(x_i, Q_n(x_i, Q_{n-1}\cdots, Q_2(x_i, Q_1(x_i, S(x_i))\cdots)$$
$$(= (x_i, Q^n(x_i, S(x_i))))$$

と，$t(= t_o + t_1 + \cdots + t_n)$ 個のダミーデータが得られる．

(Step 7) ノード A はノード B_n からデータを受け取り $R(x)$ を再構築する:
ノード B_n が(Step 6)で最後に処理したデータの中から，ノード A は，$(d_R + 1)$ -out-of- N OT（N は，$(d_R + 1)$ +ダミーデータの総数 t）を用い，(Step 4)でノード B_1 に送った x_i に対応する，$(x_i, Q^n(x_i, S(x_i)))$ の値の組を受け取る．そこから $R(x)$ を再構築し，1変数多項式 OPE と同様に，$R(0) = P(\alpha)$ を得る．R の次数は，$d_R = d_{P_1} * d_{P_2} * \cdots * d_{P_n} * K$ となる．

以上が DFOPE のプロトコルである．

3.4. 計算例

ノード群 B (B_1, B_2) は，それぞれ秘密の多項式

$$\begin{cases} P_1(x) = 3x - 4 \\ P_2(x) = 2x + 5 \end{cases}$$

を持っている．ノード A は，ある秘密の値 $\alpha = 2$ を持っている．ノード A は自分の情報を B に知られることなく，$P_2(P_1(\alpha)) = P_2(P_1(2))$ の値を取得したい．DFOPE プロトコルを使用して実現する．

(Step 1) ノード群 B，ノード A はそれぞれの持つ多項式 $P_1(x) = 3x - 4, P_2(x) = 2x + 5$，値 $\alpha = 2$ を定義する．

(Step 2) B_1, B_2 はそれぞれ，$P_1'(x) = -2x^2 + x, P_2'(x) = 4x^2 - 3x$ を考える．$d_1 = 2, d_2 = 2$ であり，また，$k_1 = 2, k_2 = 2$ より，$K = 4$ である．

$Q_1(x,y) = -2x^2 + x + 3y - 4$

$Q_2(x,y) = 4x^2 - 3x + 2y + 5$

(Step 3) ノード A は，$S(x) = x^4 - x^3 + x^2 - x + 2$ を考える $(S(0) = \alpha = 2)$．

(Step 4) ノード A は，$(x_i, S(x_i))$ を $d_R + 1 = (d_{P_1} * d_{P_2} * K) + 1 = 5$ 個準備する．

$(x_1, S(x_1)) = (1,2), (x_2, S(x_2)) = (2,12)$,
$(x_3, S(x_3)) = (3,62), (x_4, S(x_4)) = (-1,6)$,
$(x_5, S(x_5)) = (-2,32)$

また，t_0 個のダミーデータ $(x'_j, S'_j)(j=1,\cdots,t_o)$ も準備して，ノード B_1 に送信する．

(Step 5) ノード B_1 はノード A から受信した $(1,2),\cdots,(-2,32),(x'_i,S'_i)$ を $Q_1(x,y)$ に代入する．

$Q_1(1,2)=-2+1+6-4=1\rightarrow(x_1,Q_1(x_1,S(x_1)))=(1,1)$

$Q_1(2,12)=-8+2+36-4=26\rightarrow(x_2,Q_1(x_2,S(x_2)))=(2,26)$

$Q_1(3,62)=-18+3+186-4=167\rightarrow(x_3,Q_1(x_3,S(x_3)))=(3,167)$

$Q_1(-1,6)=-2-1+18-4=11\rightarrow(x_4,Q_1(x_4,S(x_4)))=(-1,11)$

$Q_1(-2,32)=-8-2+96-4=82\rightarrow(x_5,Q_1(x_5,S(x_5)))=(-2,82)$

$Q_1(x'_i,S'_i)\rightarrow(x'_i,Q_1(x'_i,S'_i))$

そして，出力された値 $(1,1),\cdots,(-2,82),(x'_i,Q_i(x'_i,S'_i))$ に，新たに t_1 個のダミーを加えてノード B_2 に送信する．

(Step 6) (Step 5)と同様に，ノード B_2 はノード B_1 から受信した $(1,1),\cdots,(-2,82),(x'_i,Q_i(x'_i,S'_i))$ を $Q_2(x,y)$ に代入する．

$Q_2(1,1)=4-3+2+5=8\rightarrow(x_1,Q^2(x_1,S(x_1)))=(1,8)$

$Q_2(2,26)=16-6+52+5=67\rightarrow(x_2,Q^2(x_2,S(x_2)))=(2,67)$

$Q_2(3,167)=36-9+334+5=366\rightarrow(x_3,Q^2(x_3,S(x_3)))=(3,366)$

$Q_2(-1,11)=4+3+22+5=34\rightarrow(x_4,Q^2(x_4,S(x_4)))=(-1,34)$

$Q_2(-2,82)=16+6+164+5=191\rightarrow(x_5,Q^2(x_5,S(x_5)))=(-2,191)$

$Q_2(x'_i,S'_i)\rightarrow(x'_i,Q^2(x'_i,S'_i))$

そして，出力された値に，新たに t_2 個のダミーを加える．

(Step 7) $t=t_o+t_1+t_2$，$5+t=N$ とすると，ノード A は 5 -out-of- N OT を使用して，$(1,8),(2,67),(3,366),(-1,34),(-2,191)$ を取ってくる．この 5 個の値から $R(x)=ax^4+bx^3+cx^2+dx+e$ を推測する．

$8=a+b+c+d+e$

$67=16a+8b+4c+2d+e$

$366=81a+27b+9c+3d+e$

$34=a-b+c-d+e$

$191=16a-8b+4c-2d+e$

より，$a=6,b=-6,c=6,d=-7,e=9$ である．よって，

$R(x)=6x^4-6x^3+6x^2-7x+9$

したがってノード A は $R(0)=P_2(P_1(2))=9$ を取得する．

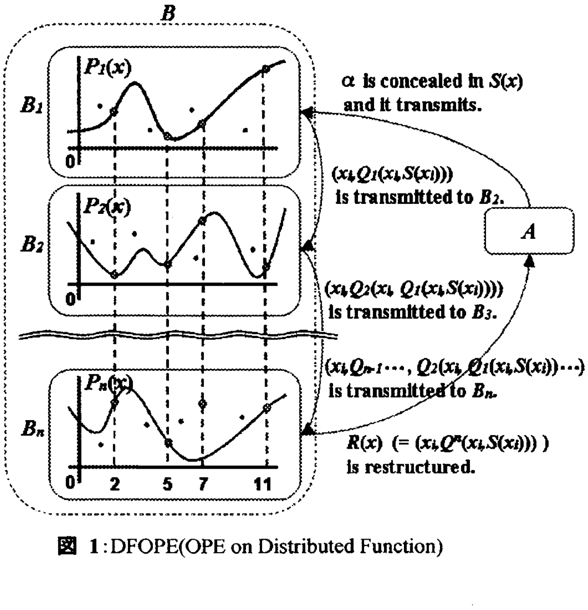

図 1:DFOPE(OPE on Distributed Function)

4. 分散関数 OPE の安全性についての議論

従来 OPE の安全性については，[19]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．

4.1. ノード A →ノード B_1 での α の秘匿性

ノード A →ノード B_1 では，ノード A のデータが，ノード B_1 に分からないことを要件とする．ノード A は，自らの持つ秘密にしたい α を，多項式 $S(x)$ の中に定数項として隠す．ノード B_1 に送るデータは，多数のダミーデータと $(x_i,S(x_i))$ であるので，ノード B_1 には，データは秘匿となる．これは，従来の OPE と同様，ダミーデータの組が多いほど秘匿性が高くなる．なお，安全な分散関数 OPE の構成に必要なダミー数は，[19]で述べられている手法と同様の手法で算出可能である．

4.2. ノード B_i →ノード B_{i+1} での $P_i(x)$ の秘匿性

ノード B_i →ノード B_{i+1} では，ノード B_i から受け取ったデータからノード B_i の多項式が，分からないことを要件とする．ノード B_{i+1} が分かることは，ノード B_i の処理したデータ $(x_i,Q^i(x_i,S(x_i)))$ である．$(x_i,Q^i(x_i,S(x_i)))$ と入力値 x_i からは，$Q^i(x,y)$ の再構成はできないので，ノード B_i のプライバシーも秘匿となる．

4.3. ノード B_n →ノード A での $P(x)$ の秘匿性

ノード B_n →ノード A では，ノード B_n から受け取ったデータからノード A がノード群 B の持つそれぞれの合成関数 $P(x)$ が分からないことを要件とする．ノード B_n →ノード A の通信では，OT を用いて，ダミーデータではない真の値だけを得る．ここからは従来の OPE と同様，$P(x),P'(x)$ は分からないので，秘匿となる．

4.4. 分散関数 OPE のセキュリティ定数 K について

DFOPE において，ノード B_i →ノード B_{i+1} 間での $P_i(x)$ の秘匿性が重要であるため，それぞれの B_i が持つ k_i は，B_1 が代表となり，各 B_i より渡されなければならない．B_1 が代表となる理由は，データの経路は，ノード B_1 →ノード B_2 →…→ノード B_n であるため，B_1 が，他の B_i の持つ多項式の次数を推測しても安全性は保たれるからである．

4.5. ノード群 B（ノード B_1 ～ノード B_n）の結託について

本プロトコルにおいて，仮にノード B_i とノード B_{i+2} が結託すると，ノード B_{i+1} の多項式がある範囲で推測されてしまう可能性がある．これはそれぞれのノードにおいて新たにダミーデータを混ぜることで推測をより困難にできるが，ダミーデータの数が多くなり効率が悪くなるという問題が発生する．Secure Multiparty Computation において一般に，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．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．

4.6. 単一ノードでの利用形態

特別な利用形態として，ノード群 B（ノード B_1 ～ノード B_n）を単一のノード B が担い，ノード A との間で分散関数 OPE を行う構成が考えられる．この場合，ノード A は定数 α を持ち，ノード B が多項式関数群 $P_1(x),P_2(x),\cdots,P_n(x)$ を持つとき，ノード A はノード B との間で安全に $P_n\circ P_{n-1}\circ\cdots\circ P_2\circ P_1(\alpha)$ を得ることができる．その際，従来の OPE を用いた場合，仮に $P_1 \sim P_n$ を1回ずつ秘密関数計算する方法の場合，途中結果の $P_1(\alpha),P_2(\alpha),\cdots,P_n(\alpha)$ の結果をノード A に知られてしまう問題があるため，ノード B において予め合成関数 $P_n\circ P_{n-1}\circ\cdots\circ P_2\circ P_1(x)$ を用意する必要があるが，分散関数 OPE

の場合は事前の関数合成を必要としない利点がある．

5. 安全なデータ加工サービスの構成

DFOPE により，例えば次のような安全なデータ加工サービスの構成が可能となる．端末 A が加工対象データを持ち，端末群 B はそれぞれ別々の，データ加工プログラムを持ち，A と $B_1 \sim B_n$ は互いにネットワークを通じて相互に接続されているとする． 端末 A から加工対象データを端末 B_1 が受け取り，端末 B_1 の持つデータ加工プログラムを用いて加工処理を行った後，端末 B_2 に渡す．以下，端末 B_n まで処理をし，最後に A に加工済データを返す一連の処理をデータ加工サービスと定義する．端末 A は，加工対象データ α を持っている．そして，α を端末群 B に知られずに，加工済データ $P_n \circ P_{n-1} \circ \cdots \circ P_2 \circ P_1(\alpha)$ を得たいとする．端末群 B はそれぞれ，データ加工を行う多項式（データ加工プログラム）$P_i(x)\ (i = 1,2,\cdots,n)$ を持っている．そして，プロトコル実行時，端末群 B のそれぞれが持つ多項式は，他の全ての端末に知られたくないとする．DFOPE により，次の2つの要件が満たされる．**(1)** 端末 A は端末群 B のそれぞれが持つデータ加工プログラム $P_i(x)\ (i = 1,2,\cdots,n)$ についてまったく分からない．**(2)** 端末群 B のそれぞれは，端末 A の持つ加工対象データ α 及び加工済データ $P_n \circ P_{n-1} \circ \cdots \circ P_2 \circ P_1(\alpha)$ と，他の端末群 B の持つデータ加工プログラム $P_i(x)$ についてまったくわからない．この構成は，端末 A が加工対象データ α を持ち，端末群 B のそれぞれが別々のデータ加工プログラム $P_i(x)$ を持つとき，自端末以外の全ての端末からの端末 A の加工対象データと加工済データの秘匿と，自端末以外の全ての端末からの端末群 B のそれぞれが持つデータ加工プログラムの秘匿による保護が両立可能であるという点で，安全である．

6. 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．[13]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．[14]では，OPE の効率の改善策について述べている．[15]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．[16]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．[17]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．[18]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．

本稿で述べた分散関数 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．

7. 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回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関数を持つとき，その合成関数を処理した結果を得ることが可能な，分散関数 OPE へと拡張した．また，分散関数 OPE が有用性を持つ具体例として，安全なデータ加工サービスの構成を示した．

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，今回拡張した分散関数 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，分散関数 OPE の特長を活かしたプライバシー重視のデータ活用などへの具体的な応用，安全かつ実用的なデータ加工サービスの実現などがある．

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] 黒澤 馨, 尾形 わかは: 暗号プロトコルの基礎数理, 特集 電子社会を推進する暗号技術, 情報処理, Vol.45, No.11, pp.1131-1133 (Nov. 2004).

[12] 今井秀樹, 花岡悟一郎: 情報量的安全性に基づく暗号技術, 電子情報通信学会論文, Vol. J87-A, No. 6, pp.721-733 (Jun. 2004).

[13] 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).

[14] 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).

[15] 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).

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

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

[18] 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).

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