M_034

Multi-Variable 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，個人の趣味・趣向や購買記録などの個人情報を含む大規模なデータベースの構築が進んでいる．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．情報漏えいに対して，個人情報管理のポリシーを定めることで防止しようとするコンプライアンスルール（法令順守）の導入も進んでいるが，秘密を管理する内部の者による漏洩事件の頻発に対して，ルールによる防止の限界が認識されてきている．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．

†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.

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．

OPE は，2 者間による秘密通信プロトコルである．Alice は入力値 α を持ち，Bob は1変数多項式 $P(x)$ を持つ．プロトコルを実行後，Alice は $P(\alpha)$ を得る．その際，次の2つの条件を満たす．(i) Alice は $P(\alpha)$ 以外の情報を一切得ることができない，(ii) Bob は α と $P(\alpha)$ について全くわからない．OPE を用いることで，2者間においてお互いの秘密情報を保護したまま秘密関数計算を行うことができる．OPE において従来，上述の $P(x)$ として利用可能な関数が，プロトコル上の制約により，1 変数多項式に限定されていた．

本稿では，1，2章で既存研究の動向を述べ本研究の位置づけを明らかにする．そして3章で基礎知識を述べた後，4章で，上述の $P(x)$ として利用可能な関数が1変数多項式に限定されていた従来の OPE を，初めて，多変数多項式が利用可能な，多変数 OPE へと拡張する．この拡張により，一度に2変数以上の多項式関数が必要な処理を，従来の OPE と同様の安全性を保ったまま実現可能となる．また，5章で，多変数 OPE を実装し，その評価を行う．さらに，6章で，多変数 OPE が有効性を持つ具体的な例として，安全な情報埋め込みサービスの構成を示し，7章で本稿をまとめる．

2. 既存研究の動向

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．[11]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．[12]では，OPE の効率の改善策について述べている．[13]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．[14],[15]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．[16]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．[17]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.

本稿で述べる多変数 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.

3. 基礎知識

3.1. Oblivious Transfer [2],[20],[21]

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. また, オークション, RSA 暗号の鍵 2 者生成, データマイニングなどにも利用されている. このように, OT は, 暗号プロトコルにおけるある意味での最小単位と考えることができる. OT については, [18], [19]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.

3.2. Oblivious Polynomial Evaluation [2],[26]

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.

4. 多変数 OPE への拡張

従来の OPE では, 2者間において, Bob の秘匿できる関数が, 1変数多項式という制約があった. そこで, OPE を多変数でも使えるよう, n 変数 k 次多項式 ($k = n * m$, m は1つの変数の最大次数) の多変数 OPE (MVOPE) への拡張について述べる.

4.1. 定義

多変数 OPE プロトコルは次の機能を実現する.

Alice は定数列 $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ を持っており, α を Bob に知られずに, $P(\alpha)$ を得たいとする. Bob は, n 変数 k 次多項式 $P(x) (x = (x_1, x_2, \cdots, x_n))$ を持っており, 多項式 $P(x)$ については, Alice に知られたくないとする. 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1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.

4.2. プロトコル

(Step 1) Bob が持っている多項式 $P(d_P = k)$ を定義する:

この多項式

$$P(x_1, x_2, \cdots, x_n) = \sum_{k_1=0}^{m} \sum_{k_2=0}^{m} \cdots \sum_{k_n=0}^{m} a_{k_1 k_2 \cdots k_n} x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$$

は Alice に知られたくないとする.

(Step 2) Bob は P を隠す多項式 P' $(P'(0,0,\cdots,0) = 0)$ を準備する: $k' = k \cdot K$ と定義すると,

$$P'(x_1, x_2, \cdots, x_n) = \sum_{k_1=0}^{k'} \sum_{k_2=0}^{k'} \cdots \sum_{k_n=0}^{k'} b_{k_1 k_2 \cdots k_n} x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$$

$((k_1, k_2, \cdots, k_n) \neq \vec{0})$. また, 次の多項式 Q を定義する.

$$Q((x_1, x_2, \cdots, x_n), (y_1, y_2, \cdots, y_n)) = P'(x_1, x_2, \cdots, x_n) + P(y_1, y_2, \cdots, y_n)$$

.

(Step 3) Alice は定数列 $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ を隠す多項式関数列 $S = (S_1, S_2, \cdots, S_n)$ を準備する:

各多項式は, $S_i(0) = \alpha_i$ $(\deg S_i = K, i = 1, \cdots, n)$ とする.

(Step 4) Alice は Bob にデータを送信:

Alice の用意する, 真の値の数 s は, $s = \sum_{i=0}^{k'*n} {}_{(n+i-1)}C_i$ となる(なお, 厳密には s はこの数より少なくてよい場合がある). $\{\gamma_i^j \mid i = 1, \cdots, n, j = 1, \cdots, s\} (\gamma_i^j \neq 0)$ を任意に生成し, $(\gamma_1^j, \gamma_2^j, \cdots, \gamma_n^j)$ と, それらを $(S_1, S_2, \cdots, S_n)$ に代入した値 $(S_1(\gamma_1^j), S_2(\gamma_2^j), \cdots, S_n(\gamma_n^j))$ からなる組を s 個作る. また, t 組のダミーデータを同様に準備する. この真の値からなる s 組と, ダミーデータ t 組とを混ぜて Bob に送信する. なお, s 個の入力列 $(\gamma_1^j, \gamma_2^j, \cdots, \gamma_n^j)$ は, 互いに線形独立となる値を設定する必要がある.

(Step 5) Bob は Alice から送られてきたデータを処理する: Bob は, (Step 4)で送られてきたデータを Q に代入し計算する.

(Step 6) Alice は $P(\alpha_1, \alpha_2, \cdots, \alpha_n)$ を得る: $s + t = u$ とすると, Alice は(Step 4)で Bob に送った $(\gamma_1^j, \gamma_2^j, \cdots, \gamma_n^j)$ に対応する, $Q((\gamma_1^j, \gamma_2^j, \cdots, \gamma_n^j), (S_1(\gamma_1^j), S_2(\gamma_2^j), \cdots, S_n(\gamma_n^j)))$ を s-out-of-u OT を行い取得し, OPE プロトコルにより $R(x)$ $(= Q(x_1, x_2, \cdots, x_n, (S_1(x_1), S_2(x_2), \cdots, S_n(x_n))))$ を再構築し, $x = (0,0,\cdots,0)$ を代入する.

$$R(0,0,\cdots,0) = Q((0,0,\cdots,0), (S_1(0), S_2(0), \cdots, S_n(0)))$$
$$= P(S_1(0), S_2(0), \cdots, S_n(0))$$
$$= P(\alpha_1, \alpha_2, \cdots, \alpha_n)$$

よって, Alice は $P(\alpha_1, \alpha_2, \cdots, \alpha_n)$ を得る.

4.3. 多変数 OPE の安全性についての議論

従来 OPE の安全性については，[26]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．ここでは多変数 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．

4.3.1. Bob の多項式関数 $P(x)$ の秘匿性

Alice が，Bob から受け取るデータは，s -out-of- u OT から，$R(\gamma)=Q(\gamma,S(\gamma))$ $(\gamma=\{\gamma_i^j \mid i=1,\cdots,n, j=1,\cdots,s\})$ だけであり，Alice が得ることができるのは，$P(x)$ の1つの値である $P(\alpha)$ だけである．よって P, P' についての詳細は分からない．これにより，従来の OPE と同様に安全性が保たれる．

4.3.2. Alice の定数列 α の秘匿性

Alice が，Bob に送信する定数列 α は，多項式関数列 $(S_1(x),S_2(x),\cdots,S_n(x))$ の中に隠す．また，$((\gamma_1^j,\gamma_2^j,\cdots,\gamma_n^j),(S_1(\gamma_1^j),S_2(\gamma_2^j),\cdots,S_n(\gamma_n^j)))$ の組は，実際の値とは異なるダミーデータと混ぜて送信されるため，従来の OPE と同様，Bob には，Alice の定数列 α は秘匿される．ただし，ダミーデータの数は，多いほど秘匿性が高くなる．なお，ここでは紙面の都合上詳しく述べないが，安全な多変数 OPE の構成に必要なダミー数は，[26]で述べられている手法と同様の手法で算出可能である．

5. 多変数 OPE の実装と評価

5.1. 実装

4 章で述べた多変数 OPE について，Java 2 Standard Edition 5.0 を利用して実装し，評価を行った．なお，今回実装したプログラムでは，公開鍵の生成やそれ以外の OT，OPE プロトコル中で，整数の桁数を制限せずに乗算などの計算を行う必要があるため，java.math.BigInteger クラスを使用している．

5.2. 計算量の予測

多変数 OPE におけるダミーデータ数と，計算量との関係を予測する．ここで，セキュリティ係数 $K=1$ で一定，また，多項式の1つの変数の最大次数 m は一定であると仮定する．また，この章では真の値の数を s，ダミーデータ数を t とおく(4.2 節(Step 4)参照)．

ダミーデータ数による処理時間(計算量)の増加に最も関係しているのは，3.2 節(Step 4)でのダミーデータの生成である．Bob が送信する $\{\gamma_i^j\}$ は $\{\gamma_1^k,\gamma_2^k,\cdots,\gamma_n^k\}\neq\{\gamma_1^l,\gamma_2^l,\cdots,\gamma_n^l\}(k\neq l,k,l=1,\cdots,s)$ を満たす必要がある．この条件を満たすため，値の組の生成時にすでに生成されている各組との比較が必要になる．$(s+t)$ 組のデータの総比較回数は，$\sum_{i=1}^{(s+t)-1} i=(s+t)(s+t-1)/2$ 回となり，生成するダミーデータの数による処理時間の増加分は，2 次関数に従うと推測される．

5.3. 計算量の評価

5.2 節で行った予測との比較のために，多変数 OPE のプログラムの計算量の評価を実際に行った．評価環境は以下の通りである．CPU: Xeon 2.40GHz，RAM : 4GB，OS : Windows 2000 SP4，VM: JRE 1.5.0_04，実行メモリ:512MB．ここで，セキュリティ係数 $K=1$，多項式 P の1つの変数の最大次数 $m=1$ で一定であるとし，対象の多項式は $P(x_1)=x_1+1$，$P(x_1,x_2)=x_1x_2+x_1+x_2+1$ とした．また，この場合の s は順に，$\sum_{i=0}^{1*1}{}_{(1+i-1)}C_i=2$ 個，$\sum_{i=0}^{2*2}{}_{(2+i-1)}C_i=15$ 個(4.2 節(Step 4)の式より)である．ダミーデータ数を 10 から 10^6 まで変えて実行時間を計測した．結果は次の図 1 の通りである．

図 1 より，ダミーデータ数が 10^4 個までは処理時間の増加はほとんど見られない．よって，実験結果からは，ダミーデータ数が 10^4 個以下において，実装したプログラムの処理時間に，ダミーデータの生成による影響があまり現れていないといえる．一方 10^4 個から 10^5 個への増分は大きく，10^5 個以上では処理時間が増加しているため，ダミーデータの生成による処理時間への影響が現れていると考えられる．

OPE では Bob が Alice から受信した $(x_i,S(x_i))$ (多変数 OPE では $(\gamma_1^j,\gamma_2^j,\cdots,\gamma_n^j,(S_1(\gamma_1^j),S_2(\gamma_2^j),\cdots,S_n(\gamma_n^j)))$) (3.2 節 (Step 4), 4.2 節(Step 4)参照)の中から Alice の S を推測するために，最大 ${}_{(s+t)}C_s$ 回真の値の組の候補を選んで計算する必要がある．推測に要する最大計算回数の具体的な数値は，それぞれ ${}_{25}C_{15}=3.27\times10^6$ $(t=10^1)$ ${}_{115}C_{15}=2.40\times10^{18}$ $(t=10^2)$，${}_{1015}C_{15}=8.61\times10^{32}$ $(t=10^3)$，${}_{10015}C_{15}=7.73\times10^{47}$ $(t=10^4)$，${}_{100015}C_{15}=7.66\times10^{62}$ $(t=10^5)$，${}_{1000015}C_{15}=7.65\times10^{77}$ $(t=10^6)$ であり，比較のために 2 の累乗を考えると，$2^{128}=3.40\times10^{38}$，$2^{256}=1.16\times10^{77}$ である．

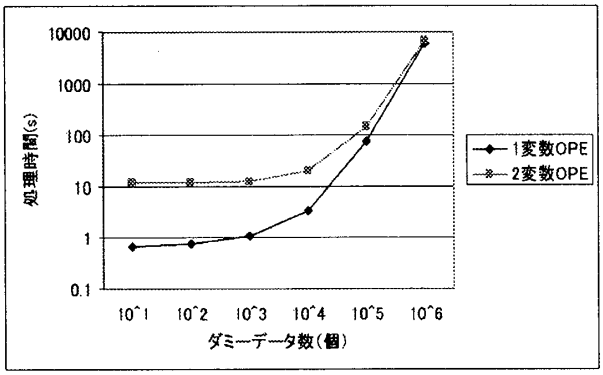

図 1:ダミーデータ数と処理時間との関係

なお，今回は単一の計算機上で計算量の比較だけを述べたが，通信コストのうち通信回数は従来と同一と考えられ，通信量はそのほとんどがダミーの数に依存する．また，今回は公開鍵の生成，k-out-of-N OT の基礎プロトコルから Java で実装したため，単純な処理でも実行時間が比較的長くなる傾向にあるが，これは実装系の変更により短くなると考えられる．今回の評価はあくまで，従来 OPE と多変数 OPE について理論的な負荷予測と，実装上の傾向との相違を確認するために行った．

6. 安全な情報埋め込みサービスの構成

多変数 OPE の特長を特に生かした応用例として，データとアルゴリズムの双方を保護する，安全な情報埋め込みサービスの構成を述べる．なお，1 変数多項式に限定されていた従来の OPE では，2つ以上の情報を入力として合成するような秘密関数計算は不可能である．

6.1. 従来手法の問題点

ネットワーク上で情報埋め込みサービスを提供する場合の問題点を述べる．

6.1.1. クライアント処理型

クライアント処理型は，サーバの持つ埋め込みプログラム(アルゴリズム)の複製をクライアントが受け取り，クライアント側で情報埋め込み処理を行う形態である．この場合，クライアントの埋め込み対象データと埋め込み情報のサーバからの秘匿は可能である．しかし，サーバは不特定多数のクライアントに対して埋め込みプログラムの配布をした場合でも，不正なクライアントによる解読などの攻撃に十分な耐性を確保することが必要になる．埋め込みプログラムは，埋め込みに使用するアルゴリズムが知られてしまうと埋め込み情報を取り除かれるなどの攻撃が容易になる脆弱性を抱えていることが指摘されている[22]，[23]，[24]．また，ソフトウェアの難読化技術には限界があり，難読化によってプログラムをブラックボックス化することは不可能であることが，ある計算モデル上で証明されている[25]．そのため，クライアント処理型の，サーバが不特定多数のクライアントに埋め込みプログラムを使用させるモデルは，埋め込みプログラムの解析に対する危険性を高めてしまう．そのため，情報埋め込み技術に高い解読耐性を要求する場合，次に述べる，埋め込みプログラムを非公開にできるサーバ処理型の形態をとることが必要である．

6.1.2. サーバ処理型

サーバ処理型の場合，クライアントの持つ埋め込み対象データと埋め込み情報をサーバが受け取り，サーバの埋め込みプログラム(アルゴリズム)を用いて埋め込み処理を行う．サーバの埋め込みプログラムをクライアントから秘匿することはできるが，クライアントの持つ埋め込み対象データと埋め込み情報をサーバから秘匿できないという問題がある．

6.2. 構成

この章では，埋め込みサービスを，クライアントから埋め込み対象データと埋め込み情報をサーバが受け取り，サーバの埋め込みプログラム（アルゴリズム）を用いて埋め込み処理を行った後，埋め込み済データをクライアントに返す一連のサービスと定義する．また，提案モデルの構成を図 2 に示す．提案モデルはプログラムの提供者，サーバ（受託業者），クライアントの3つの主体からなる．なお，サーバがプログラムの提供者を兼ねる2主体の形態でも有効性を持つ．

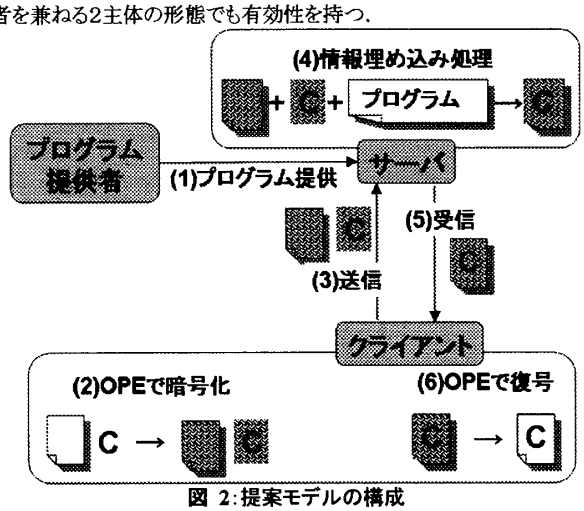

図 2：提案モデルの構成

多変数多項式 OPE を用いた情報埋め込み処理プロトコルでは，Alice は埋め込み対象データ列 $\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_m)$ と，埋め込み情報列 $\beta=(\beta_1,\beta_2,\cdots,\beta_n)$ を持っており，α と β を Bob に知られずに，埋め込み済データ $P(\alpha,\beta)$ を得たいとする．Bob は，埋め込み処理を行うアルゴリズムである $(m+n)$ 変数多項式 $P(x)$ $(x=(x_1,x_2,\cdots,x_{(m+n)}))$ を持っており，埋め込み処理をする際，埋め込み処理関数 $P(x)$ については，Alice に知られたくないとする．多変数 OPE により，(1)Alice は，$P(x)$ の1つの値である $P(\alpha,\beta)$ しか得ることはできない，(2)Bob は，α，β と $P(\alpha,\beta)$ についてまったく分からない，という2つの要件が満たされる．

6.3. 手順

提案モデルの手順は，以下のようになる．

(Step 1) プログラムをサーバに設置：
プログラムの提供者は，多変数 OPE を利用可能な，埋め込みプログラムを作成し，サーバに設置する．

(Step 2) クライアントがデータを送信：
クライアントは，埋め込み対象データと埋め込み情報を，多変数 OPE を用いて暗号化してサーバに送信する．

(Step 3) サーバによる埋め込み処理：
サーバは，(Step 2)でクライアントから送られてきた埋め込み対象データと埋め込み情報を，(Step 1)で委託された埋め込みプログラムを用いて，埋め込み処理を行う．

(Step 4) クライアントが埋め込み済データを受信：
クライアントは，暗号化された埋め込み済データをサーバから受信し，多変数 OPE を用いて復号する．

なお，ここでは便宜上，暗号化，復号という用語を用いて解説しているが，OPE は真のデータとダミーデータを区別させない技術であることを断っておく．また，より複雑な情報埋め込み処理を行う場合，埋め込み処理を単一の関数にすることが困難と予想されるが，例えば埋め込み処理全体の中の秘密にしたい重要な計算（アルゴリズム）部分だけを関数化して，その部分を多変数 OPE を用いて保護することで，アルゴリズム解読に強い情報埋め込みサービスの構成が可能となると考えられる．また，電子透かしやステガノグラフィなどのより実用的な情報埋め込みサービスへの適用可能性についても今後さらに検討していきたい．

7. まとめ

本稿では，利用可能な関数が1変数多項式に限定されていた従来の OPE を，多変数多項式が利用可能な，多変数 OPE へと拡張した．この拡張により，一度に2変数以上の多項式関数が必要な処理を，従来の OPE と同じ安全性を保ったまま実現可能となる．また，実装した多変数 OPE プログラムの処理時間について評価を行った．そして，拡張した多変数 OPE が特に有効性を持つ具体的な例として，クライアント側の埋め込み対象データと埋め込み情報，サーバ側の埋め込みプログラムの双方を保護可能な，安全な情報埋め込み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，今回拡張した多変数 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，多変数 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，安全かつ実用的な情報埋め込み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, CSS2005, 5B-1, pp.343-348 (Oct. 2005).
[11] 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).
[12] 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).
[13] 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).
[14] Wakaha Ogata, Kaoru Kurosawa: Oblivious Keyword Search, Journal of Complexity, Vol.20, pp. 356-371 (2004).
[15] Michael J. Freedman, Yuval Ishai, Benny Pinkas, Omer Reingold: Keyword Search and Oblivious Pseudorandom Functions, Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, Proceedings, p. 303 (2005).
[16] S. Jha, L. Kruger, P. McDaniel: Privacy Preserving Clustering, 10th European Symposium On Research In Computer Security (ESORICS) (2005).
[17] 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).
[18] 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).
[19] 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).
[20] Shimon Even, Oded Goldreich and Abraham Lempel: A Randomized Protocol for Signing Contracts, Comm. of the ACM, Vol.28, No.6, pp.637-647 (1985).
[21] Sheng Zhong and Yang Richard Yang: Verifiable Distributed Oblivious Transfer and Mobile Agent Security, DIALM-POMC'03, pp.12-21 (2003).
[22] 門田暁人: ソフトウェアプロテクションの技術動向（前編）－ソフトウェア単体での耐タンパー化技術－, 情報処理, Vol.46, No.4, pp. 431-437 (Apr. 2005).
[23] 門田暁人: ソフトウェアプロテクションの技術動向（後編）－ハードウェアによるソフトウェア耐タンパー化技術－, 情報処理, Vol.46, No.5, pp. 558-563 (May. 2005)
[24] 村瀬一郎: 特集 インフォメーションハイディング, 情報処理, Vol.44, No.3, pp.225-259 (Mar. 2003).
[25] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S. and Yang, K.: On the (im)possibility of Obfuscating Programs, Lecture Notes in Computer Science, Vol.2139, pp. 1-18 (2001).
[26] D. Bleichenbacher and P. Q. Nguyen, "Noisy Polynomial Interpolation and Noisy Chinese Remaindering," EUROCRYPT 2000, LNCS 1807, pp. 53-69 (2000).