様 式 C－１９、F－１９、Z－１９ （共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ６ 月 ２０ 日現在

機関番号：３２６８９
研究種目：挑戦的萌芽研究
研究期間：2013～2015
課題番号：２５５４０１３１

研究課題名（和文）プライバシー保護バイオインフォマティクス基盤技術の開発と応用

研究課題名（英文）Development of basic technology for privacy-preserving bioinformatics and its application

研究代表者
浜田　道昭（Hamada, Michiaki）

早稲田大学・理工学術院・准教授

研究者番号：００５９６５３８

交付決定額（研究期間全体）：（直接経費）　2,900,000円

研究成果の概要（和文）：個人のゲノム情報や薬のたねとなる化合物情報などは，機密情報として取り扱うことが必要となる．一方，オープンサイエンスの立場からは，これらの情報を積極的に利用して他の情報と合わせてデータマイニングを行うことが重要である．本研究では，これらの生物分野の重要情報を秘匿したまま様々なデータマイニングを行う方法論の開発を行った．具体的には，化合物データベースの秘匿検索，隠れマルコフモデルを用いたゲノム情報の秘匿検索，秘匿配列アラインメントの技術を開発した．

研究成果の概要（英文）：It is highly demanded to deal with the information of personal genome and chemical compound secretly, because they are sensitive information that should not be leaked. On the other hand, from a viewpoint of "open" science, it is important to perform data-mining by combining those sensitive information with other data. In this study, we have developed several methods to perform data-mining, making those information secret. Specifically, we developed (i) privacy-preserving search for chemical database, (ii) privacy-preserving genome sequence search with hidden Markov Model (HMM) and (iii) privacy preserving sequence alignment, all of which will be useful toward open science of biology.

研究分野：バイオインフォマティクス

キーワード：プライバシー保護　化合物検索　アラインメント　隠れマルコフモデル

１．研究開始当初の背景

バイオインフォマティクス分野において，ユーザやデータベース／Web サーバの秘密情報を開示すること無く，これらの間でデータマイニングを行うこと（プライバシー保護データマイニング;PPDM）が可能となれば様々な新しいビジネスやオープン・イノベーションが見込まれる．しかしながら，バイオインフォマティクス分野では，大規模な Web サーバやデータベースが多く，かつ製薬企業などが要求するセキュリティレベルも極めて高い．

２．研究の目的

　バイオインフォマティクス分野で利用可能な PPDM 基盤技術の開発およびその応用を行う．

３．研究の方法

本研究では，主に以下の２つの研究を行った：(i) 化合物データベースの秘匿検索と，(ii) 配列データベースへの秘匿検索．いずれも，加法準同型性暗号に基づいたプロトコル設計を行った．

４．研究成果

(1)　化合物データベースの秘匿検索技術の研究開発

化合物データベースの内容とユーザのクエリ化合物を互いに秘匿したまま，データベース検索を行うための方法を開発した．検索の指標としては，下記の Tversky index を用いた．

$$\mathrm{TI}_{\alpha,\beta}(p,\ q)=\frac{|p\cap q|}{|p\cap q|+\alpha|p\backslash q|+\beta|q\backslash p|}.$$

ここで p， q は化合物のフィンガープリント（ビットベクトル）である．秘匿検索は加法準同型性暗号に基いて行われている．

　実際の化合物データベースである ChEMBL を用いた実験の結果，提案検索手法は既存技術 GP-MPC(General purpose multi-party computation)に比べて大幅に速度が早いことが示された（図 1)．

	ChEMBL_1000	ChEMBL_Full
CPU time (s)		
SSCC (server)	0.69	167.19
SSCC (client)	1.53	172.37
GP-MPC (server)	4, 075.15	–
GP-MPC (client)	4, 366.18	–
Communication size (MB)		
SSCC (server → client)	2.24	265.33
SSCC (client → server)	0.03	0.03
GP-MPC (server → client)	42.50	–
GP-MPC (client → server)	2, 128.00	–

The experiment on ChEMBL Full by GP-MPC did not finish within 24 hours.

図 1 既存手法と提案手法の速度の比較

(2)　疾患リスクの評価に向けた加法準同型性暗号によるプライバシー反故 HMM の実装と評価

① 概要：ゲノムシーケンシングコストの低下により，疾患に関する遺伝子の研究や，個人ゲノムを用いた診断などが行われるようになってきた．しかしゲノムは個人に関する非常に多くの重要な情報を含んでいるため扱いが難しく，プライバシー上の問題が研究や遺伝子診断の普及の妨げになっている．そこで，本研究では安全にゲノム解析を行う方法として，HMM に対して加法準同型性暗号の一つである Paillier 暗号と， 暗号プロトコルの一つである 1-out-of-n 紛失通信を適用した Privacy-preserving HMM のための Secure Forward Algorithm を提案する．Secure Forward Algorithm では， ゲノムを所持する人と HMM を所持する人の二者が， 互いの持っている情報を一切共有せずに計算結果のみを得ることができる． 暗号化を適用しない通常の Forward Algorithm と比較して， 誤差率 0.0465%という高い精度で安全な計算を行うことができた．これによって，プライバシーを保護した状態でゲノムから疾患リスクの評価などを行うことができると考えられる．

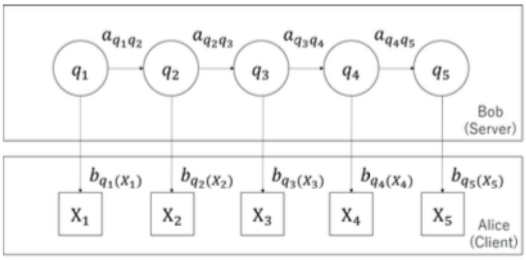

図 2 隠れマルコフモデル（HMM）

②手法：本研究では，確率モデルの１種である隠れマルコフモデル（図 2）を用いる．クライアントの Alice が持っているゲノム配列断片を，サーバの Bob が持っている HMM で評価を行い，評価値を Alice に返す．

③プロトコル：プロトコルは Pathak らに基づき設計した．本プロトコルは５つにわけられる：(i)安全に対数計算を行うプロトコル，(ii)安全に指数演算を行うプロトコル，(iii)安全に logsum 演算を行うプロトコル，(iv)紛失通信プロトコル（暗号プロトコルの一種であり，データの送信者が n 個のデータ，データの受信者がそれに 紐づく n 個のインデックスを所持しているとき，受信者は n 個のうちの何番目のデータを抜き取ったかを送信者に知られることなく希望のデータのみを得ることができる．)，(v) secure な前向きアルゴリズムを構築するのが目的である．アルゴリズムの詳細を以下に示した．

Algorithm 1 Secure Logarithm Protocol

Require: $\xi(\theta)$
Ensure: $\xi(\log\theta)$
1: Bob は正の整数の乱数 β を選び，$\xi(\theta)^{\beta} = \xi(\beta\theta)$ を計算して Alice に送る
2: Alice は $\xi(\beta\theta)$ を復号して得た $\beta\theta$ から $\log\beta\theta$ を計算
3: Alice は $\log\beta\theta$ を暗号化し，$\xi(\log\beta\theta)$ を Bob に送る
4: Bob は手順 1 で選んだ β から $\xi(-\log\beta)$ を計算する
5: Bob は $\xi(\log\beta\theta)\cdot\xi(-\log\beta) = \xi(\log\theta)$ を計算する

Algorithm 2 Secure Exponent Protocol

Require: $\xi(\log\theta)$
Ensure: $\xi(\theta)$
1: Bob は正の整数の乱数 β を選んで $\xi(\log\beta)$ を計算し，$\xi(\log\beta\theta)$ を Alice に送る
2: Alice は $\xi(\log\beta\theta)$ を復号して得た $\log\beta\theta$ から $\beta\theta$ を計算
3: Alice は $\beta\theta$ を暗号化し，$\xi(\beta\theta)$ を Bob に送る
4: Bob は $\xi(\beta\theta)^{\frac{1}{\beta}} = \xi(\theta)$ を計算する

Algorithm 3 Secure Logsum Protocol

Require: $\xi(\log\theta_1), \xi(\log\theta_2), \cdots, \xi(\log\theta_n), a_1, a_2, \cdots, a_n$
Ensure: $\xi(\log\sum_{i=1}^{n} a_i\theta_i)$
1: Bob と Alice は $\xi(\log\theta_1), \xi(\log\theta_2), \cdots, \xi(\log\theta_n)$ に対して Secure Exponent protocol を n 回行い，$\xi(\theta_1), \xi(\theta_2), \cdots, \xi(\theta_n)$ を計算する.
2: Bob は $\xi(\theta_1), \xi(\theta_2), \cdots, \xi(\theta_n)$ と $a_1, a_2, \cdots, a_n$ から，Paillier 暗号の加法準同型性を利用して $\xi(\sum_{i=1}^{n} a_i\theta_i)$ を計算する.

$$\xi(\sum_{i=1}^{n} a_i\theta_i) = \prod_{i=1}^{n} \xi(a_i\theta_i) = \prod \xi(\theta_i)^{a_i}$$

3: Alice と Bob は Secure Logarithm Protocol を行い $\xi(\sum_{i=1}^{n} a_i\theta_i)$ から $\xi(\log\sum_{i=1}^{n} a_i\theta_1)$ を計算する.

Algorithm 4 1-out-of-n 紛失通信

1: Alice はインデックス $\beta_1, \beta_2, \cdots, \beta_n (\beta_i = 1, \beta_j = 0 (j \neq i))$ を作る
2: Alice はインデックスを暗号化し，Bob に送る.
3: Bob は受け取ったインデックス $\xi(\beta_1), \xi(\beta_2), \cdots, \xi(\beta_n)$ と自分の持っているデータ $\alpha_1, \alpha_2, \cdots, \alpha_n$ から以下の計算を行う

$$\xi(\beta_1)^{\alpha_1}, \xi(\beta_2)^{\alpha_2}, \cdots, \xi(\beta_n)^{\alpha_n} = \xi(\alpha_1\beta_1), \xi(\alpha_2\beta_2), \cdots, \xi(\alpha_n\beta_n)$$

4: Bob は計算した $\xi(\alpha_1\beta_1), \xi(\alpha_2\beta_2), \cdots, \xi(\alpha_n\beta_n)$ を Alice に送り返す.
5: Alice が受け取った $\xi(\alpha_1\beta_1), \xi(\alpha_2\beta_2), \cdots, \xi(\alpha_n\beta_n)$ を復号すると，i 番目が α_i となり，その他は 0 となっているため希望の i 番目のデータのみを入手することができる.

Algorithm 5 Secure Forward Algorithm Protocol

Require: 配列 $x_1, x_2, \cdots, x_T$ および HMM $\lambda = (A, B, \Pi)$
Ensure: $\xi(\log Pr\{x_1, x_2, \cdots, x_T|\lambda\})$
1: Bob は乱数 γ を選び，$\log b_j(v_k) + \gamma$ を計算する ($for\ 1 \le k \le K, 1 \le j \le N$)
2: Alice は自分の $x_1, x_2, \cdots, x_T$ に基づいて，対応する $\log b_j(x_t) + \gamma$ を 1-out-of-n 紛失通信を用いて入手する ($for\ 1 \le t \le T$)
3: Alice は受け取った $\log b_j(x_t) + \gamma$ を暗号化し，$\xi(\log b_j(x_t) + \gamma)$ を Bob に送る ($for\ 1 \le t \le T, 1 \le j \le N$)
4: Bob は $\xi(\log b_i(x_T) + \gamma)\cdot\xi(-\gamma) = \xi(\log b_i(x_T))$ を計算する ($for\ 1 \le t \le T, 1 \le j \le N$)
5: Bob は $\xi(\log\alpha_1(j)) = \xi(\log\pi_j)\cdot\xi(\log b_j(x_1))$ を計算する ($for\ 1 \le j \le N$)
6:

$$\xi(\log\alpha_{t+1}(j)) = \xi(\log\sum_{l=1}^{N}\alpha_t(l)a_{ij})\cdot\xi(_j(x_{t+1}))$$

に基づいて，Alice と Bob は Secure LOGSUM protocol を用いて $\xi(\log\alpha_T(j))$ を計算する ($for\ 1 \le t \le T-1, 1 \le i, j \le N$)
7: Secure LOGSUM protocol を用いて，$\xi(\log\alpha_T(j))$ から $\xi(\log\sum_{j=1}^{N}\alpha_T(j)) = \xi(\log Pr\{x_1, x_2, \cdots, x_T|\lambda\})$ を計算する

④評価：1-out-of-n 紛失通信について，各 bi 数におけるプログラム全体の処理にかかった時間，鍵生成にかかった時間および計算部分にかかった時間の平均と標準偏差を図 3 に纏めた．以下に示す結果は，Bob と Alice の持っているデータ数及びインデックス数 n = 15 の場合の測定結果である．

		全体	鍵生成	計算部分
256bit	平均	16.61	2.129	14.48
	偏差	2.225	1.153	1.753
512bit	平均	32.03	8.785	23.25
	偏差	7.692	7.144	2.392
1024bit	平均	135.3	57.44	77.84
	偏差	51.67	50.89	5.156
2048bit	平均	1020	571.7	448.7
	偏差	513.4	512.0	24.05

図 3 1-out-of-n 紛失通信計算時間

また，前向きアルゴリズム全体の計算時間を図 4 に示してある．2048bit の場合でも 8.7 秒程度で計算ができていることが分かる．また bit 数を増やすと鍵生成にかかる時間が全体に占める割合が増加する傾向は 1-out-of-n 紛失通信と同じで，原因も同じであると考えられる．

		全体	鍵生成	計算部分
256bit	平均	226.5	2.915	223.6
	偏差	33.08	1.593	32.76
512bit	平均	375.7	11.36	364.3
	偏差	50.08	8.809	48.45
1024bit	平均	1460	85.30	1374
	偏差	197.9	81.20	170.5
2048bit	平均	8666	820.2	7845
	偏差	931.5	781.2	513.0

図 4 前向きアルゴリズムの計算時間

どの bit 数でも誤差率の平均値は同じ値となり，暗号の bit 数と誤差の相関関係は見られなかった．計算の途中で小数から整数へ直す際に切り捨てが生じていることが誤まりを減らせる可能性はある． 最後に Secure Forward Algorithm Protocol と，暗号化 を適用していない Forward Algorithm の処理時間の比較を 行った．暗号化を適用した場合の計算時間が，暗号化を適 用していない場合の何倍になるかを計算したものを以下の 表 7 に示す．暗号化を適用しない場合の処理時間は，1000 回の平均で 0.0426[msec]であった．差の主な原因であると考えられる．今回は 10 の 6 乗 をかけて整数化したが，この数字の桁数を増やすことによって更に誤差を減らせる可能性がある．

最後に Secure Forward Algorithm Protocol と，暗号化 を適用していない Forward Algorithm の処理時間の比較を 行

った.暗号化を適用した場合の計算時間が,暗号化を適 用していない場合の何倍になるかを計算したものを以下の 図５ に示す.暗号化を適用しない場合の処理時間は,1000 回の平均で 0.0426[msec]であった. 2048bit では暗号化した場合としない場合で 20 万倍程度の差が出たが,暗号化していない場合の処理時間が非常に 短いことと,暗号化を適用しても 8.7 秒程度で終わっていることから Secure Forward Algorithm も十分に実用的で あると考えられる.

256bit	512bit	1024bit	2048bit
5318	8820	3427×10	2034×10^2

図 5 Secure 前向きアルゴリズムの暗号化なしとの比較

④結論：暗号化を適用した Forward Algorithm に対して処理時 間と誤差の測定を行った結果,誤差率 0.0465%という高い精度で計算を行うことができた(図６).また計算時間に関してはパソコンの性能次第で短縮可能であると考えられる.しかし,本研究では入力の配列長 T = 5,隠れ状態数 N = 5 という小さな値での実験を行ったが,実際に疾患リスクの評価を行うと想定した場合,配列長が数 100 程度必要となってくる. Forward Algorithm では計算量が O(N^2T)であるため,入力の配列長や隠れ状態数によっては処理時間が大幅に増加すると考えられる.配列長や状態数を増やした場合についても追加で実験を行い,その際の処理時間の変化についても関係を調べる必要がある.今後の課題は本研究で省いた通信部分の実装をし,通信時間も含めた処理時間の測定を行うことである現在 TCP/IP を用いて通信部分の実装を行っている.また疾患リスクの評価を行うためには,適切なパラメータの推定と状態数を考えることも必要となる.パラメータの推定には,ゲノムを秘匿したまま計算することができる Secure Baum-Welch Algorithm の実装が必要となってくるが,本研究で Secure Forward Algorithm の実装を行ったため, Secure Baum-Welch Algorithm の実装も容易にできる考えられる.

(3) プライバシー保護アラインメント技術の

256bit	512bit	1024bit	2048bit
0.0465	0.0465	0.0465	0.0465

図 6 アルゴリズムの誤差

開発と評価

配列ペアワイズアラインメントを，準同型暗号を用いて安全に計算する方法の実装を行った.

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計２件）
① Kana Shimizu, Koji Nuida, Hiromi Arai, Shigeo Mitsunari, Nuttapong Attrapadung, Michiaki Hamada, Koji Tsuda, Takatsugu Hirokawa, Jun Sakuma, Goichiro Hanaoka, Kiyoshi Asai, Privacy-preserving search for chemical compound databases, BMC Bioinformatics (2015)16 Suppl 18:S6. ［査読有り］

〔学会発表〕（計１件）
① 三品気吹・浜田道昭, 疾患リスクの評価へ向けた加法準同型性暗号によるプライバシー保護HMMの実装と評価, 第 108 回 MPS・第 46 回 BIO 合同研究発表会. 沖縄県、OIST. 2016 年７月５日（発表決定)。

〔図書〕（計０件）

〔産業財産権〕
○出願状況（計０件）

○取得状況（計０件）

〔その他〕
ホームページ等：特になし

６．研究組織
(1)研究代表者
浜田 道昭（Hamada, Michiaki）
早稲田大学・理工学術院・准教授
研究者番号：00596538

(2)研究分担者
清水　佳奈（Shimizu, Kana）
早稲田大学・理工学術院・准教授
研究者番号：60367050

(3)連携研究者
花岡 悟一郎（Hanaoka, Goichiro）
独立行政法人産業技術総合研究所・研究グループ長
研究者番号：30415731

津田　宏治（Tsuda, Koji）
東京大学・大学院新領域創成科学研究科・教授
研究者番号：90357517

フリス　マーティン（Frith, Martin）
産業技術総合研究所・人工知能研究センター・主任研究員
研究者番号：40462832

浅井　潔（Asai, Kiyoshi）
東京大学・大学院新領域創成科学研究科・
教授
研究者番号：30356357