6ZA-01 完全準同型暗号を用いた秘匿委託計算システムにおける動作ログデータのマイニング処理に関する検討

山本 百合† 小口 正人†
†お茶の水女子大学

1. はじめに

活動量計などから取得できる人間の動作のログデータは，健康維持や病状把握において重要な役割を持つ運動状態を表すため，データマイニングで有用な情報を得ることが期待できる．そのため，活動量計のデータをデバイス自体から，クラウド上のデータセンタにデータを転送し，データセンタ側が解析を行うシステムが考えられる．

しかし，動作ログデータは個人の生活の詳細が判明してしまうデータであるため，プライバシ保護の観点からデータの適切なセキュリティ管理が求められる．そのため，プライバシ保護データマイニングのデータや結果にノイズを加えることで匿名性を高める手法を適用することが考えられるが，それらの手法ではマイニング結果が正確ではなくなるため，適切な健康維持の知見が得られなくなってしまう．

本研究では，データを暗号化した状態で乗算と加算の操作が可能な完全準同型暗号を利用することで，安全に動作ログデータの活用が可能なシステムの構築を考えたい．特に完全準同型暗号を用いた秘匿委託データマイニング計算システムを応用することによって，Apriori アルゴリズムを用いて，一定期間における特定個人の動作ログデータから，頻出なパターンを発見する手法について検討を行う．また完全準同型暗号暗号同士の演算は計算量が大きいため，システムの高速化のためにサーバサイドにマスタ・ワーカ型分散処理を用いている．本報告では，委託データマイニング計算システムを，動作ログデータに適用する形に再設計し，クラウドコンピューティングを想定した環境下で実験を行った結果を掲載する．

2. 先行研究

筆者ら [1] は，完全準同型暗号を用いた委託 Apriori 計算の分散処理による高速化を適用したシステムを構築した．このシステムでは，トランザクションごとに購入したか否かを 0 または 1 のバイナリ表現されたバスケットデータを対象とし，Apriori 計算を行うサーバ・クライアント型のシステムとして設計されている．また完全準同型暗号は暗号文同士の演算の計算量が大きいことから，サーバサイドの実行時間を減少させるために，マスタ・ワーカ型分散処理を適用した．ただし完全準同型暗号は加算と乗算の機能を有するが，比較をすることは極めて困難な暗号である．そのため Apriori で必要とされるミニマムサポートとの比較は，クライアント側で結果を復号した上で比較を行うことにした．よってマスタ・クライアント間のコミュニケーションは最大でアイテム数分の回数生じる．

3. 提案手法

3.1 概要

本研究では，図 1 の動作ログデータに対する完全準同型暗号を用いた秘匿データマイニングのマスタ・ワーカ型の分散システムを提案する．

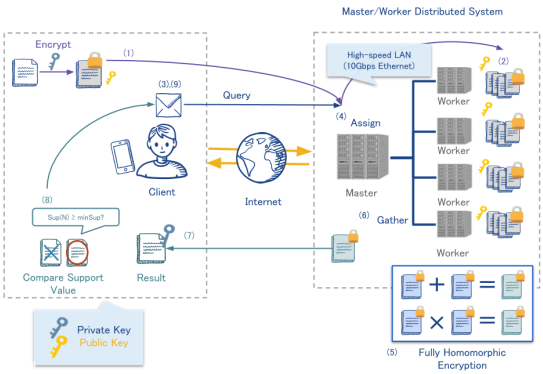

図 1: 提案手法概観

(1) クライアントはデータを秘密鍵で暗号化し，マスタに公開鍵と共に委託．
(2) マスタは受け取ったデータと公開鍵を各ワーカに転送．
(3) クライアントはマスタに長さ 1 のアイテムセットに対するサポート値の計算依頼を行う．
(4) マスタは各ワーカにタスクを割り振る．
(5) 各ワーカは完全準同型暗号を用いたサポート値計算を行う．
(6) マスタは各ワーカから結果を収集．
(7) マスタはクライアントへ収集した結果を送信．
(8) クライアントはデータを復号し，各アイテムセットのサポート値と閾値を比較．
(9) クライアントは閾値を超えたアイテムセットをマスタに送信することで，次の長さのクエリとする．
(10) (4)〜(9) を閾値を超えるアイテムセットが無くなるか，アイテムセットが最長になるまで繰り返す．

3.2 動作ログデータの入力

加速度センサによるデータ収集を行っている電気通信大学新谷准教授より，加速度センサにおけるデータと運動状態の分類を行う際の形式を示したダミーデータを提供して頂いた．本研究では，特に加速度センサの値から静止・安静・座位・立位・軽作業・作業・運動・歩行・ジョギング・非装着・データなしの 11 種類に運動状態を分類し，特定の運動状態がどの時間からどの時間まで発生していたかが 1 行ずつ記された形式のデータを活用する．

ある期間の特定の人物の動作ログデータから，その人物が最も頻繁に行う生活パターンを抽出するために，Apriori アルゴリズムで計算を行う．そのため，Python プログラムを用いて，動作ログデータの Apriori アルゴリズムに適したデータへの成形を行った．アイテムとして様々なカテゴリが考えられるが，今回は曜日 (日曜日〜土曜日)・1 時

A Study of Application of Secure Outsourcing Calculation System Using Fully Homomorphic Encryption for Mining Motion Log Data
† Yuri YAMAMOTO, † Masato OGUCHI
Ochanomizu University (†)

間ごとに区切った時間帯 (0時〜24時)，運動状態 (11種類) をそれぞれをアイテムとし，アイテム数 42 として 0 と 1 で表現するデータを生成した．

4. 実験

4.1 実験環境

C++で実装し，完全準同型暗号計算には HElib[2] を，分散化における各マシンの制御のために Open MPI を用いた．実装したマスタ・ワーカ側のプログラムをお茶の水女子大学内に構築されたサーバルーム内の同一性能の 4 台の Linux マシンに設置した．またクライアント側のプログラムを Amazon Web Services の EC2 m4.4xlarge インスタンス 1 台に設置した．マシン性能を表 1 に記す．

表 1: マシン性能

マスタ・ワーカ (お茶の水女子大学)	型式	Intel®Xeon®Processor E5-2643 v3
	コア	6
	スレッド	12
	CPU	3.40 GHz
	メモリ	512 GB
	OS	CentOS 6.9
クライアント (AWS EC2)	型式	Intel®Xeon®CPU E5-2686 v4
	コア	8
	スレッド	16
	CPU	2.30 GHz
	メモリ	64 GB
	OS	CentOS 6.10

マスタ・ワーカ側のマシン 4 台それぞれに対し，最大 2 スロット分のワーカの演算を行わせた．またそのうち 1 台にマスタの機能を持たせる．最大で 8 スロット分のワーカを稼働させてワーカ数ごとの実行時間を比較する実験を行なった．

4.2 実験結果

特定の人物の 15 日間の生活を想定した動作ログデータから，21,620 行からなる Apriori 計算用のデータを生成し，実験を行った．ワーカ数ごとの実行時間のグラフを図 2 に示す．

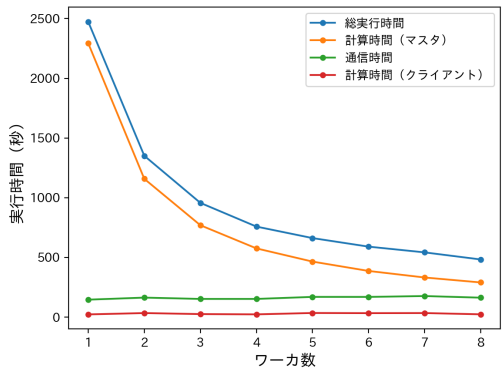

図 2: 動作ログデータに対する秘匿データマイニングシステム実行時間 (秒)

ワーカ数が増加するにつれて総実行時間を短縮されている．特にサーバ側の計算時間で分散化効果が顕著である．通信時間とクライアント上の計算に関しては，ワーカ数の増加に対してほとんど変化しないことが示された．

また分散処理化の評価として，高速化率を式 (1) より算出し，式 (2) に基づいた Amdahl の法則による並列度と高速化率の関係 [3] と共に図 3 に示す．

$$\text{高速化率} \leq \text{逐次実行時間 (秒)} \,/\, \text{並列実行時間 (秒)} \tag{1}$$

$$\text{高速化率} \leq \frac{1}{(1 - \text{並列実行時間の割合}) + \frac{\text{並列実行時間の割合}}{\text{ワーカ数}}} \tag{2}$$

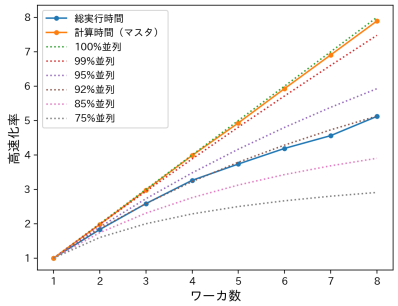

図 3: 本手法の高速化率と Amdahl の法則による並列度に対する高速化率

サーバ側の完全準同型暗号の暗号文同士の計算時間は 99%並列時に近い高速化率で分割されていることが示された．ただし，通信時間，クライアント側での暗号化や復号に伴う計算時間，ファイル入出力等に必要な時間も含めた総実行時間では，92%並列時の高速化率となる．したがって，今回の実験環境においては，本システムのワーカ数を最大限に増やす場合，最大で約 12.5 倍の高速化率が期待できる．

5. まとめと今後の課題

完全準同型暗号を用いた秘匿データマイニングシステムを人間の動作ログデータに対して適用し，クラウドコンピューティングを想定した環境下で実験を行った．今後は動作ログデータの Apriori 計算結果を活用し，動作ログデータのクラス分類など，更に実用的なデータマイニングへの応用を考えたい．

6. 謝辞

本研究を進めるにあたり，大変有益なアドバイスを頂いた電気通信大学新谷隆彦准教授，早稲田大学山名研究室並びに工学院大学山口研究室の皆様に感謝いたします．

本研究は一部，JST CREST JPMJCR1503 の支援を受けたものである．

参考文献

[1] Yuri Yamamoto and Masato Oguchi. Distributed secure data mining with updating database using fully homomorphic encryption. In *"IMCOM 2019", "9-4"*, 2019.

[2] Shoup V. and Halevi S. HElib. `http://shaih.github.io/HElib/index.html`. Accessed: 2017-1.

[3] Clay Breshears. *The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications.* " O'Reilly Media, Inc.", 2009.