平成 30 年度 東邦大学理学研究科情報科学専攻 修士論文

マルチパーティ計算を利用した秘匿情報統合分析に対するゲーム理論の適用

学籍番号　6517002

作本 壮志朗

金岡研究室

目 次

1 はじめに

データを秘匿したまま計算する手法の 1 つであるマルチパーティ計算（Secure Multi-Party Computation, MPC）では和や積の基本演算の高速化の研究[1]が主に行われている。MPC を実現するアプローチとしては主にガーブルドサーキットによる方法、秘密分散を用いた方法で情報論的な部品を活用する方法、秘密分散を用いてかつ準同型暗号を用いた方法、完全準同型暗号を用いる方法がある。本研究では、秘密分散を用いた方法で情報論的な部品を活用する方法に焦点を当てる。MPC の正常な実施には複数の組織が秘密分散された入力情報を用いて協力して MPC を実行する必要がある。これまでの研究では MPC 技術そのものの研究が盛んに行われており、機能や性能の向上がもたらされた。現実的な利用が視野に入ってきたものの、実社会での実施方法については十分に検討されているとは言い難い。

これらを踏まえ本研究の目的を以下と置いた。

- 実社会での MPC 実施に向けた議論
- MPC 実施にあたって各組織が提供するリソースの明確化
- MPC 実施にあたってのモデルの明確化

そこで本論文では、MPC の実社会での実施方法の検討の 1 つとして、MPC を行う各組織が得る利益に着目し、その利益分配の方法についてゲーム理論を適用したものを提案する。MPC に参加する複数の組織の特性を考慮し、各組織の行動をゲーム理論を用いた分析をすることで社会的側面の議論が可能となる。具体的には本研究の提案手法により、MPC 実施に際して各組織が提供する計算量や事前計算量、提供計算リソース、通信帯域といった利得要素をもとにした利益の分配を検討することが可能になる。

各利得要素（計算量, 事前計算量, 提供計算リソース, 通信帯域）とは MPC 実行の処理時間に影響を与える要素である。先に述べたように MPC を実現するアプローチは複数あり、MPC の手法によって各利得要素が与える利得への影響が異なる可能性があるため調査を行った。秘密分散を用いた方法で情報論的な部品を活用する方法の MPC を調査した結果、計算量と事前計算量は各組織間で等量に負担することが分かった。また、提供計算リソースと通信帯域に関しては、実機を用いた実験を行い、各組織が提供した利得要素の性能の違いが処理時間に影響を与えることを確認した。これらの結果を踏まえ利益の分配方法を提案した。

さらに、MPC のユースケースとして情報統合分析に焦点を当て、複数の組織が協力して MPC を実施するモデルの提案を行う。そして、情報統合分析の応用的なモデルとして、プライバシ保護型リスク分析へのゲーム理論を適用した利得の配分方法を提案する。Liu ら [1] によって提案、実装が行われたプライバシ保護型リスク分析は作本ら [2] によって性能向上が果たされたが、リスク分析に使用される分析ルール情報がプライバシ保護型リスク分析を行う組織間で共通するといった問題があった。本研究ではこの問題の指摘も行い、プライバシ保護型リスク分析の新たなシステムモデルの提案も行った。先行研究からのシステムモデルの変更に伴い各組織が持つ分析ルール情報の統合が必要となる。分析ルール情報は各組織ごとに独自の情報であるため秘匿化したままの統合が不可欠である。本研究では簡易的な分析ルール情報の統合方法の提案し、実装と評価を行った。評価の結果、作本らが実装したシステムと同等の性能であることが分かった。また、各組織が提供した分析ルール情報の量を算出する方法の提案も行った。

[1] 高速秘密計算説明資料 https://jpn.nec.com/press/201612/images/1502-01- 01.pdf

これらの提案手法により、計算量や事前計算量、提供計算リソース、通信帯域、各組織が持つ分析ルール情報といった利得要素をもとにプライバシ保護型リスク分析実施による利益の分配を検討することが可能になる。そしてプライバシ保護型リスク分析というユースケースに焦点を当てて議論をすることで、その利益分配の合理性を議論することを可能にした。

本研究の貢献は以下である。

- MPC 実施における利益分配の提案
- MPC 実施のユースケースとしてプライバシ保護型リスク分析実施における利益分配の提案
- 上記 2 つの貢献による実社会での MPC 実施に向けた議論の実現

論文の構成は以下の通りである。まず、第 2 章で本研究に関連する前提知識について解説する。次に、第 3 章で本研究の関連研究について述べる。第 4 章でゲーム理論を適用した MPC 参加者利得の検討について述べる。第 5 章でプライバシ保護型リスク分析におけるシステムモデルについて述べる。第 6 章でゲーム理論を適用したプライバシ保護型リスク分析参加者利得の検討について述べる。第 7 章で本研究から明らかになった課題について述べる。最後に、第 8 章で本研究についてまとめる。

2　前提知識

2.1　秘密分散と秘密計算

秘密分散とは秘匿したい情報を複数に分割し、それらのうちいくつかを集めると元の情報が復元できるという方法である。それぞれの分散した情報からは元の情報が推測できず、元の情報はこの意味で秘匿される。このような性質は機密性を持ちつつ鍵を用いずにデータを復号できるという可用性を持つため、データ管理のために利用できる。

分散した情報を用いて秘密計算[2]を行える方式も提案されている。現在、秘密分散を用いた加算や乗算、さらに除算や比較、論理演算、ソートなどの秘密計算が可能である。マルチパーティ計算を用いることで、通常のデータ分析と同等の結果を高いプライバシ保護の下で得ることができる。しかし、マルチパーティ計算では通常の計算機上の計算に比べて処理速度が低下してしまうことが実用上の課題となっている。

処理速度の低下の大きな要因は基本演算のオーバーヘッドである。マルチパーティ計算ではデータの秘匿性を保つために、乗算のような通常の計算機では一命令で実行可能な基本演算にも複雑な処理を必要とし、その結果、処理時間も大きくなってしまう。これに対応するために 2.1.1 項で紹介する SEPIA といった、効率のよい基本演算を備えたフレームワークの提案、実装が行われている。

2.1.1　SEPIA

SEPIA（Security through Private Information Aggregation）[3] は Burkhart らによって提案、実装されたマルチパーティ計算のための Java フレームワーク[3]である。秘密分散法に Shamir の秘密分散法 [4] を使用しており、加算, 乗算, 比較といった基本演算が可能である。

本研究での実装や実験ではマルチパーティ計算のフレームワークとして SEPIA を利用した。

2.1.2　Damgard らの方式

近年マルチパーティ計算の研究が進み、特に不正者が参加者の半数を下回る場合に安全な手法については、実用的な速度の方法が提案されてきている。しかし、半数以上の参加者が結託してしまうと情報を復元できてしまうことから適用場面が限定されていた。これに対して、自身以外のすべての参加者が結託しても安全なマルチパーティ計算が Damgard らによって提案された [5]。この手法は所望のマルチパーティ計算の実行前に事前計算をしておくことにより、不正者が参加者の半数を下回る場合に安全な手法と同程度の実行時の速度を達成している。

Damgard らの方式は、秘密分散に基づいたマルチパーティ計算であり、能動的な攻撃者による n パーティのうち $n-1$ 人までの買収に対して安全である。

2.2　ゲーム理論

ゲーム理論[4]とは複数の意思決定をする主体が、その意思決定に関して相互作用する状況を研究する理論である。ここで意思決定をする主体とは、個人であったり、企業であったり、時には国家

[2]秘密計算は、Secure Multi-Party Computation, MPC, マルチパーティ計算とも呼ばれる
[3]「https://sites.google.com/view/sepia-mpc」にて公開されている
[4]渡辺隆裕. ゼミナール ゲーム理論入門. 日本経済新聞出版社, 2008 を参考

であったりし、現在考えている問題で、1 つのまとまった意思決定ができると認識できる単位である。このような、意思決定をする主体が 2 つ以上あり、それらが相互に及ぼし合いながら意思決定を行うときに、どのように行われるか、またはどのように行われるべきか、について考察する。

本節ではゲーム理論の中でも本論文で使用した用語を簡単に紹介する。

2.2.1 非協力ゲームと協力ゲーム

ゲーム理論は大きく、非協力ゲームと協力ゲームに分けることができる。

- 非協力ゲーム
  個人 1 人 1 人を社会の構成単位と考えて、その個人がどのような行動を選択するかについて扱う理論。各個人には、行動の選択肢が与えられていて、各個人がどのような行動を選べば、どのような利益があるかが表現されている。そして、個人の選ぶ行動が焦点となる。

- 協力ゲーム
  個人ではなく提携[5]を構成単位の基礎としている。さらに非協力ゲームと違い、各個人や提携が行動を選択するという概念はなく、選択肢を選ぶような形式にはなっていない。協力ゲームでは、各提携が組まれたときの利益がいくつあるかが、行動に関係なく与えられている。その下で、どのような提携が組まれるか、または提携を考えたときに個人に分配される利益はどのような結果となるかが焦点となる。

2.2.2 戦略形ゲームと展開形ゲーム

非協力ゲームは、戦略形ゲームか展開形ゲームの 2 つに大別される。

- 戦略形ゲーム
  すべてのプレイヤーが、同時に行動をするゲーム
  例. じゃんけん

- 展開形ゲーム
  あるタイミングで何人かのプレイヤーが同時に行動したり、順番に行動したり、同時の行動と交互の行動が混合して行われるような複雑なゲーム
  例. チェス、将棋

戦略形ゲームと展開形ゲームの解はどちらもナッシュ均衡[6]である。

2.3 ブルームフィルタ

Bloom が考案 [6] した空間効率の良い確率的データ構造であり、要素が集合のメンバーであるかどうかのテストに使われる。偽陽性（False Positive）による誤検出の可能性があるが、偽陰性（False Negative）はない。要素を集合に追加することができるが、削除することはできない。また、ブルームフィルタのデータ構造からは正しい集合を取り出すことはできない特性持つ。これは偽陽性の要素も無数に取り出されてしまうからである。

[5] 3 人以上の意思決定においては、個人が個々に行動するだけではなく、「誰と誰が組むか」という提携という問題が重要になる

[6] ある戦略の組に対して、各プレイヤーが、どんな他の戦略を選んでも利得を高くできないような状態

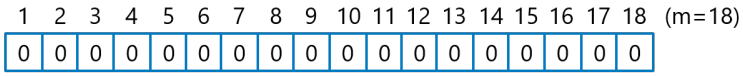

図 1: 空のブルームフィルタ

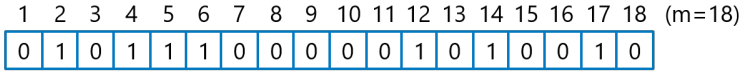

図 2: 要素追加後のブルームフィルタ

空のブルームフィルタは全て 0 に設定された m ビットのビット配列である。ブルームフィルタには k 個のハッシュ関数が定義されており、キー値を k 個のハッシュ関数を通し m 個の配列位置に 1 を設定する。図 1 は、$m = 18$ の空ブルームフィルタである。

集合 $\{x, y, z\}$ を空ブルームフィルタに追加するとし、それぞれキー値を $k = 3$ 個のハッシュ関数に通した配列位置を以下とする。

- x: 2, 6, 14
- y: 5, 12, 17
- z: 4, 6, 12

追加後のブルームフィルタは図 2 となり、正しい集合は $\{x, y, z\}$ となる。

また、ブルームフィルタに追加しない集合要素として v と w を考える。それぞれのハッシュ関数を通した配列位置は以下とする。

- v: 2, 4, 5
- w: 5, 14, 16

あるキー値がブルームフィルタが含まれているかは、配列位置の数値がすべて 1 であるかを確認すればよい。その結果、要素 v, x, y, z は要素追加後のブルームフィルタに含まれ、w は含まれないと判定される。このように偽陽性による誤検出の可能性があり、ブルームフィルタのデータ構造からは正しい集合を取り出すことはできない。

3 関連研究

3.1 秘密計算の実用可能性の研究

荒木らの論文 [7] では秘密計算の実用可能性について議論がされている。そこでは、秘密計算の中心をなす技術である MPC を実現する複数のアプローチを大まかに分類し、各アプローチの評価軸として処理性能、参加者モデルなどを述べている。また、MPC の実用的なサービス形態についても述べられている。

MPC を実現するアプローチとしては、ガーブルドサーキットによる方法、秘密分散を用いた方法で情報論的な部品を活用する方法、秘密分散を用いてかつ準同型暗号を用いた方法、完全準同型暗号を用いる方法の 4 分類が紹介されている。そして各アプローチの評価軸の 1 つである処理性能として、オンライン計算時間（入力が与えられてから出力するまでの時間）、オフライン計算時間（入力とは独立に事前に計算する時間）、通信回数、通信量、スループット（単位時間当たりに並列で計算できる回数）が紹介されている。また各アプローチの評価軸の 1 つである参加者モデルとして、クライアント-サーバ・モデル、P2P・モデル、クライアント-マルチサーバ・モデル、非結託 P2P・モデル、クラウド内プロセス・モデルが紹介されている。最後にマルチパーティ計算の実用的なサービス形態として、サーバの安全管理措置、情報統合分析、認証サーバ、耐サイドチャネル攻撃が紹介されている。

荒木らは、上記を踏まえた考察の結果として MPC への参加者が 3 者である秘密分散を用いた 3 者 MPC が最も高速であり、現実のサービスに適用できると結論づけている。

3.2 Katz らの研究

我々の知る限りでは、暗号手法とゲーム理論を組み合わせる研究は Katz の論文 [8] により議論が始まった。Katz は互いに信頼できない当事者間のため現実的なモデルとプロトコルの設計を目的として、ゲーム理論のアプローチとテクニックを暗号プロトコルの設計[7]と組み合わせることに大きな関心を寄せ、方向性として以下の 2 つを示した。

- Applying cryptography to game theory（ゲーム理論への暗号手法の適用）
- Applying game-theory to cryptography（暗号手法へのゲーム理論への適用）

Katz は、信頼できる仲介者が存在する環境であれば、特定のゲーム理論的均衡が達成可能であるとした。ゲーム理論への暗号手法の適用では、仲介者を当事者自身が運営する分散暗号プロトコルにどの程度置き換えることができるかが焦点となる。

Katz の論文では暗号手法へのゲーム理論の適用例として合理的な MPC について述べられている。しかし、合理的な MPC の詳細な定義は定められていない。

また、暗号手法へのゲーム理論の他の適用例として、Gordon と Katz[9] によって合理的な秘密分散のプロトコル設計が提案されている。これについては 3.2.1 項で紹介する。

[7]暗号学におけるプロトコル設計；参加者の振舞いが semi-honest や malicious であると仮定したときにプライバシー情報の漏洩を防ぐことが目的
ゲーム理論におけるプロトコル設計：参加者は合理的に意思決定を行うと想定したうえで、参加者にプロトコルに従うことが合理的であると判断させることが目的

3.2.1 Gordon-Katz プロトコル

2 者間の合理的な秘密分散プロトコルを紹介する。

プレイヤー 1 と 2 のために、各 $i = 1, 2, ...$ について、確率 δ で $a_i + b_i = s$, 確率 $1 - \delta$ で $a_i + b_i = \perp$ を満たす満たす無限個のシェア $(a_1, a_2, ...), (b_1, b_2, ...)$ を用意する。このとき、s は復元したい秘密であり、$\perp$ は復元を失敗を意味する。各ラウンド $i = 1, 2, ...$ において両プレイヤーは a_i と b_i を同時に出す。$a_i + b_i = s$ であれば秘密が復元されるのでプロトコルを終了する。$a_i + b_i = \perp$ であれば、次のラウンドへと進む。ただし、1 人でもシェアを出していなければ、そこでプロトコルは終了する。パラメータ δ を適切に設定すれば、このプロトコルは合理的なプレイヤーによって正しく実行されることになる。

次にプレイヤーに対して以下の利得を考える。自分だけが秘密を復元し、相手が復元できない場合は U^+, 両プレイヤーとも復元した場合は U, どちらも復元できない場合は U^- を得るとする。利得の大小関係は $U^+ > U > U^-$ とする。両プレイヤーがプロトコルに従っていれば、いつか秘密が復元されるため、利得 U を得る。一方、プロトコルから逸脱してシェアを出さない場合、確率 δ で U^+ を、確率 $1 - \delta$ で U^- を得ることになるため、期待利得は $\delta\ U^+ + (1 - \delta)U^-$ である。したがって、$\delta\ U^+ + (1 - \delta)U^- < U$ を満たすように δ を十分小さく設定すれば、プレイヤーは逸脱する理由がなくなる。

ただし、Gordon-Katz プロトコルには以下の 2 つの望ましくない性質がある。

- パラメータ δ を設定するために、利得 U, U^+, U^- の値を知る必要があること
- シェアの同時提出が必要であること

後の研究でこれらの問題は Gordon-Katz プロトコルに特有のものではなく、合理的なプレイヤーによる秘密分散においては避けられないことが判明した。

3.3 プライバシ保護型リスク分析の研究

Liu らは MPC を利用したプライバシ保護型リスク分析[8]を提案した [1]。荒木らの研究で示されたマルチパーティ計算の実用的なサービス形態としては情報統合分析[9]に該当する。

プライバシ保護型リスク分析では、以下の 3 つの Provider を必要とする。

- Network System Information Provider
  ネットワークシステムに関する情報を提供
  例. 各ホストの構成、アクセス制御リスト、ネットワークポリシー
- Risk Analysis Rule Provider
  リスク分析に使用するルール情報を提供
  例. MulVAL[10] の Interaction Rules
- Computation Engine Provider
  Network System Information Provider 及び Risk Analysis Rule Provider から情報を受け取り、MPC protocol をベースとしたリスク分析処理を提供

[8]分析依頼者はリスク分析に使用する情報を秘密分散し、分散した情報をリスク分析を行う複数の専門組織に提供する。複数の専門組織は MPC を利用して分散前の情報を知ることなくリスク分析を行う

[9]異なるデータ所有者の異なるデータベースをマージして統合分析をする時に、全てのデータを 3 個のサーバに分散してから MPC にて分析する。いずれのサーバーもデータを直接見ることが出来ないため、データを直接共有して分析するのと比較して、情報漏洩対策が容易になる

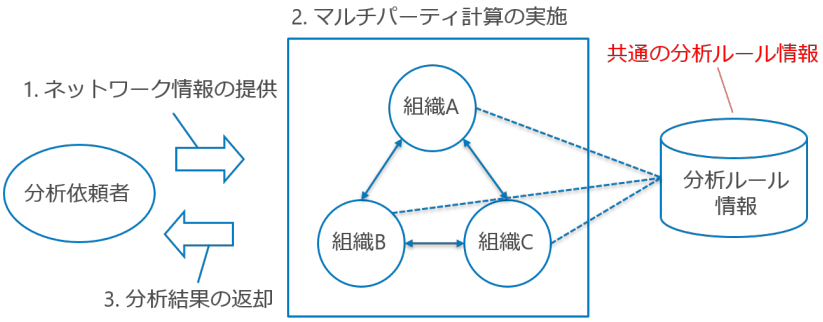

図 3: Liu らのシステムモデル

そして、3 つの Provider の組み合わせとして以下の 3 種類が提案されている。

- Risk Analysis Rule Provider が Computation Engine Provider を兼ねる
- Network System Information Provider が Computation Engine Provider を兼ねる
- Network System Information Provider 及び Risk Analysis Rule Provider, Computation Engine Provider それぞれを別の Provider が担当

Liu らの提案手法では Risk Analysis Rule Provider が Computation Engine Provider を兼ねる場合において、MPC platform に FairplayMP[11]、リスク分析手法に MulVAL を使用したプライバシ保護型リスク分析システムを実装しており、実行時間やコンパイル時間の評価が行われている。システムモデルは図 3 となる。図 3 中の分析依頼者が Network Information Provider を担い、組織 A, B, C が Risk Analysisu Rule Provider と Computation Engine Provider を担当している。

Liu らのシステムモデルの一連の処理の流れは以下となる。

1. ネットワーク情報の提供
   分析依頼者が自組織のネットワーク情報をリスク分析を行う複数の専門組織（組織 A, B, C）に対して秘密分散して提供
2. マルチパーティ計算の実施
   専門組織（組織 A, B, C）は受け取った秘密分散されたネットワーク情報と組織間で共通の分析ルール情報を用いてマルチパーティ計算によるリスク分析を実施
3. 分析結果の返却
   マルチパーティ計算によって得られた結果を分析依頼者に対して返却

しかし、Liu らが実装したプロトタイプシステムでは、Input Party の数の制約とコンパイル操作のパフォーマンスの 2 つの問題点があった。作本ら [2] はそれらの問題点を解決する以下の 3 つのアプローチを提案した。

1. FairplayMP の改良
2. MulVAL 以外のリスク分析手法の利用
3. 新しい MPC platform の使用や開発

作本らはこのアプローチの中から新しい MPC platform の使用（FairplayMP 以外の MPC プラットフォームの利用）を採用し、実際にシステムを試作し評価を行った。そこでは、MPC platform に SEPIA[3]、リスク分析手法に MulVAL を採用し、MulVAL の Interaction Rule を SEPIA のプログラム内に記述することでプロトタイプシステムを実装が行われた。試作システムの評価の結果、Liu らの手法における 2 つの問題点を解決し、プライバシ保護型リスク分析システムの性能向上を果たした。

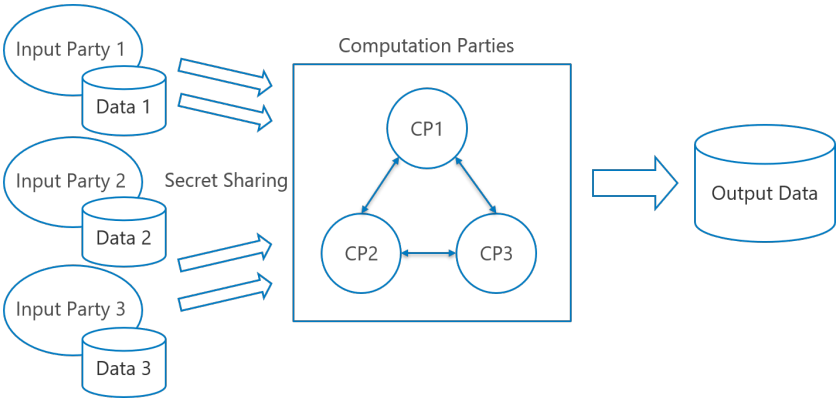

図 4: MPC の実行例

4　ゲーム理論を適用した MPC 参加者利得の検討

MPC の研究全般に言えることであるが、これまでの研究では MPC に参加する各組織がいかに利益を得るかといった社会的側面の議論が十分にはされてこなかった。そこで本章では、社会的側面の議論の 1 つとして、MPC に参加する各組織が得る利益に着目し、複数の制約条件の中でいかに利益を分配するかの議論を行う。

議論を行うモデルを考慮するにあたって、複数の意思決定をする主体が、その意思決定に関して相互作用する状況を研究する理論であるゲーム理論の考えを取り入れた。

4.1　利得要素

3.1 節で述べたように。MPC の処理性能の評価軸には、オンライン計算時間、オフライン計算時間などがある。処理性能の評価と示すようにこれらは MPC 実施に伴い必要となる処理である。本研究では、MPC の処理に関する利得要素として以下の 4 つに焦点を当てることとする。

- 計算量
  MPC 評価軸のオンライン計算時間に関係
  MPC 実施にあたり各組織が提供するマシンの処理量
- 事前計算量
  MPC 評価軸のオフライン計算時間に関係
  必要に応じて行う事前計算にあたり各組織が提供するマシンの処理量
- 提供計算リソース
  MPC 実施にあたり各組織が提供する計算リソース
- 通信帯域
  MPC 実施にあたり各組織が提供する通信帯域

以降 4.2 節から 4.5 節で各利得要素の詳細について述べる。

4.2 計算量に対しての利得

図 4 にあるように MPC の実施には計算資源（Computation Parties）が必要である。計算資源は異なる複数の組織が提供することになり、MPC に参加する組織は計算資源提供による計算量の見返りとして利得を受け取ることが考えられる。そこで、各組織が MPC に参加するにあたって提供する計算量に応じて利得を受け取る方式を検討する。

計算資源を提供するにあたって参加組織に発生する利得を計算利得（CP : Computation Profit）とし以下の式で定義する。ただし、各組織の利得合計を 1 とし、n は MPC の参加組織数とする。

$$CP(X) = 1/n \tag{1}$$

MPC では各組織が行う処理の量が等しいため、組織 X が受け取る計算の量に対しての利得 $CP(X)$ は参加する組織数の逆数を取るものとなる。

組織 A, B, C が存在したときの各組織の計算利得は以下となる。

- $CP(A) = 1/3$
- $CP(B) = 1/3$
- $CP(C) = 1/3$

4.3 事前計算量に対しての利得

2.1.2 項で述べたように所望のマルチパーティ計算実行前に事前計算を行う Damgard らの方式が存在する。この方式は、MAC 付き 3 つ組を利用し、能動的な攻撃者による n パーティのうち $n-1$ 人までの買収に対して安全なマルチパーティ計算である。しかし、事前計算とあるようにマルチパーティ計算実行前、つまり、事前に計算を行う必要がある。Damgard らの方式の事前計算では、マルチパーティ計算を行う組織が somewhat 準同型暗号を使い、MAC 付き 3 つ組と呼ばれる MPC 実行時の入力とは無関係なランダムな値を事前に大量に準備する。MAC 付き 3 つ組の準備には時間がかかるので事前に計算しておくこと MPC のオンライン計算時間の短縮にもつながる。マルチパーティ計算とは別に処理が必要になるので、事前計算に対しても利得を与える必要があると考えた。また、濱田らの研究 [12][13] では、事前計算を手助けする情報を提供する信頼できるサーバをの存在を仮定することで、Damgard らの方式を効率化する方法を提案している。

濱田らの方式ではサーバに対しての利得も考える必要があるが、Damgard らの方式ではマルチパーティ計算を行う組織だけに対しての利得だけを考慮すれば済む。本研究では、簡単のため Damgard らの方式に対しての利得を考えることとする。Damgard らの方式の事前計算を実施するのは、マルチパーティ計算を行う組織そのものであり主従関係は存在しない。そのため事前計算にあたっての処理の量は組織間で等しいと考えられる。

事前計算を行うにあたって参加組織に発生する利得を事前計算利得（PP : Preprocessing Profit）とし以下の式で定義する。ただし、各組織の利得合計を 1 とし、n は MPC の参加組織数とする。

$$PP(X) = 1/n \tag{2}$$

組織 A, B, C が存在したときの各組織の事前計算利得は以下となる。

- $PP(A) = 1/3$

- $PP(B) = 1/3$
- $PP(C) = 1/3$

このように Damgard らの方式による事前計算量に対しての利得と、4.2 節の計算量に対しての利得は配分割合は同じものであった。つまり、事前計算の有無に関わらず処理量に対しての利得は各組織間で等分であることが分かった。

4.4 提供計算リソースに対しての利得

次に提供計算リソースに基づいた利得分配を検討する。Ben-David らは「提供計算リソースが少ない（weak）マシンと多い（strong）のマシンが混在するとき、マルチパーティ計算の処理時間は提供計算リソースが少ないマシンに引っ張られる」と言及している [11]。実際、各組織の処理内容が同一であれば、各組織での処理終了は提供計算リソースに依存し、ある 1 つの組織が提供計算リソースを多く提供し、残りの組織がそれと比較して提供計算リソースを少なく提供した場合は、全体としては提供計算リソースの少ないマシンの処理に依存した実行時間となる。このような場合において各組織が得る利益が組織間で同一である場合、各組織が多くの計算リソースを提供する積極的な動機はない。多くの計算リソースの提供を検討したとしても、協力してマルチパーティ計算を行う他の組織に 1 つでも少ない計算リソースを提供することがあれば実行時間に対しての利得がないため、提供する計算リソースを少ないものに変更することも十分に考えられる。そして相対的に多くの計算リソースを提供することになる他の組織が少ない計算リソースに変更する連鎖が起こる。最終的には情報統合分析に参加する全ての組織が少ない計算リソースを提供することとなってしまう。

この問題を解決するために参加する組織の提供計算リソースが異なることを考慮に入れた利得の配分方法を提案する。具体的には、情報統合分析を行う各組織が提供する計算リソースを測定し、提供した計算リソースの比に対して利得を配分する。

ここで、組織 X ごとの計算リソース（PR: Provide Resource）を $PR(X)$ と表すこととする。$PR(X)$ の値としては、1 秒あたりの処理性能である FLOPS や、メモリ量、あるいはそれらの統合情報など、様々な方法が考えられる。ここではその詳細はスコープ外とするが、各組織の $PR(X)$ は加算可能として検討する。

提供計算リソースの比率に対しての利得を提供計算リソース比利得（PRP: Provide Resource Proit）とし、$PR(X)$ を用いて表すと以下のようになる。

$$PRP(X) = PR(X) / \sum_{O} PR(O) \tag{3}$$

例えば、参加組織 A, B, C の提供計算リソースが次の場合

- $PR(A) = 1$
- $PR(B) = 3$
- $PR(C) = 1$

提供計算リソースに対しての利得の全体を 1 としたとき提供計算リソース比利得は以下になる。

- $PRP(A) = 1/5$

表 1: MPC 処理時間計測に用いた機器

機器名	CPU	RAM	OS	Java
Surface Pro 4	Intel®Core™i7-6650U CPU @ 2.20GHz	8.00GB	Windows 10	jre 1.8.0_161
Let's note CF-SX2	Intel®Core™i7-3540M CPU @ 3.00GHz	16.00GB	Windows 10	jre 1.8.0_181
LIFEBOOK SH54/G	Intel®Core™i3-2350M CPU @ 2.30GHz	4.00 GB	Windows 10	jre 1.8.0_144

図 5: MPC 処理時間計測時の機器構成

- $PRP(B) = 3/5$
- $PRP(C) = 1/5$

各組織が提供した計算リソースに依存した報酬の支払いにより、相対的に計算リソースを多く提供する組織が高い利得を受け取ることになる。これにより、各組織が提供する計算リソースの増加が期待できる。また、各組織が提供計算リソースに対しての利得を増やすためにどの程度計算リソースを提供するかといったゲーム性も生まれる。

4.4.1 提供計算リソースが実行時間に影響を与える実験結果

本項では提供計算リソースとして計算マシンの性能を扱い、提供計算リソースが MPC の実行時間に影響を与えるかを確認する。実験は表 1 の機器を使って図 5 の機器構成で行った。図 5 中の PC とは、表 1 のいずれかの機器である。また、LAN には WLX202[10]を使用した。

今回の実験では SEPIA のサンプルプログラムである tutorial.jar の実行時間を計測した。実行には 6 つの Java プログラム（Input Peer[11] 3 つと Private Peer[12] 3 つ）を立ち上げる必要がある。初期設定だと 6 つのプログラムを実行するのは 1 台の機器であるが、今回はマシン性能の違いによる実行時間の変化に加え複数機器での実行時間を確認するため、複数台での実行ができるよう config ファイルを変更した。Input Peer（ip）と Privacy Peer（pp）の組み合わせを 5 パターンごと、機器ごとに各 100 回ずつ計測を行った。各パターン、各機器ごとの平均実行時間は表 2 と 3 の結果となった。

まず表 2 から見ていく、パターン 1 と 5 から分かるように単体でプログラムを動かした場合、提供計算リソースが多いマシン（Surface Pro 4）のほうが実行時間が短くなっている。また、複数台でプログラムを実行するパターン 2, 3, 4 のいずれも、相対的に提供計算リソースが少ないマシン（ここでは Let's note CF-SX2）単体で動かした場合よりも実行時間が長くなっている。このことから通信帯域も実行時間に関係していることが分かる

次に表 3 を見る。こちらも表 2 と同様に単体でプログラムを動かした場合、提供計算リソースの多いマシン（Surface Pro 4）のほうが実行時間が短いこと、複数台でプログラムを実行した場合は相対的に提供計算リソースが少ないマシン（ここでは LIFEBOOK SH54/G）単体で動かした場合よりも実行時間が長くなっていることが分かった。

[10]YAMAHA 無線 LAN アクセスポイント WLX202
[11]Input Peer: マルチパーティ計算に使用する入力情報を提供する組織
[12]Private Peer: 入力情報を受け取りマルチパーティ計算を行う組織

表 2: MPC 処理時間計測結果 1

パターン	Surface Pro 4（PC1）	Let's note CF-SX2（PC2）	実行時間（秒）
1	ip1, ip2, ip3, pp1, pp2, pp3	なし	10.177
2	ip1, ip2, pp1, pp2, pp3	ip3	10.318
3	ip1, ip2, ip3, pp1, pp2	pp3	10.574
4	ip1, ip2, pp1, pp2	ip3, pp3	10.591
5	なし	ip1, ip2, ip3, pp1, pp2, pp3	10.257

表 3: MPC 処理時間計測結果 2

パターン	Surface Pro 4（PC1）	LIFEBOOK SH54/G（PC2）	実行時間（秒）
1	ip1, ip2, ip3, pp1, pp2, pp3	なし	10.176
2	ip1, ip2, pp1, pp2, pp3	ip3	10.515
3	ip1, ip2, ip3, pp1, pp2	pp3	10.611
4	ip1, ip2, pp1, pp2	ip3, pp3	10.730
5	なし	ip1, ip2, ip3, pp1, pp2, pp3	10.480

これらの結果から MPC の実行時間には確かにマシンの提供計算リソースが関わることが分かった。そのため、4.4 節で提案した提供計算リソースに対しての利得は 1 つの妥当な利得要素だと考えられる。

4.5 通信帯域に対しての利得

図 6 にあるように MPC の実行には各組織が提供する計算マシンをつなぐ通信帯域が必要になる。4.4.1 節では通信帯域が実行時間に影響があることを示した。本節では通信帯域に対して与える利得の提案を行う。

提供計算リソースの考えと同様にある 1 つの組織が幅の広い通信帯域を提供し、残りの組織がそれと比較して幅の狭い通信帯域を提供した場合は、全体としては幅の狭い通信帯域に依存した実行時間となる。このような場合において各組織が得る利益が組織間で同一である場合、各組織が幅の広い通信帯域を提供する積極的な動機はない。幅の広い通信帯域の提供を検討したとしても、協力してマルチパーティ計算を行う他の組織に 1 つでも幅の狭い通信帯域を提供することがあれば実行時間に対しての利得がないため、提供する通信帯域を幅の狭いものと変更することも十分に考えられる。そして相対的に幅の広い通信帯域を提供することになる他の組織が幅の狭い通信帯域に変更する連鎖が起こる。最終的には情報統合分析に参加する全ての組織が幅の狭い通信帯域を使うこととなってしまう。

この問題を解決するために参加する組織が利用する通信帯域が異なることを考慮に入れた利得の配分方法を提案する。具体的には、情報統合分析を行う各組織が提供する帯域幅を測定し、帯域幅の比に対して利得を配分する。各組織が利用する帯域のモデルは 4.5.2 項で示す。

組織 X ごとの帯域幅（BW: Band Width）を $BW(X)$ と表すこととする。このとき、各組織の $BW(X)$ は加算可能として検討する。

通信帯域の幅の比に対する利得を帯域幅比利得（BWP: Band Width Proit）とし、$BW(X)$ を

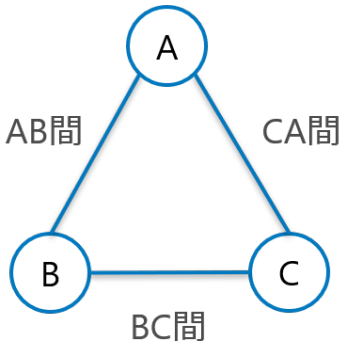

図 6: MPC の通信形態

用いて表すと以下のようになる。

$$BWP(X) = BW(X) / \sum_{O} BW(O) \tag{4}$$

例えば、参加組織 A, B, C が利用する帯域幅が次の場合

- $BW(A) = 1$
- $BW(B) = 2$
- $BW(C) = 1$

帯域幅に対して利得の全体を 1 としたとき帯域幅比利得は以下になる。

- $BWP(A) = 1/4$
- $BWP(B) = 2/4$
- $BWP(C) = 1/4$

各組織が提供した通信帯域に依存した報酬の支払いにより、相対的に大きい通信帯域を提供する組織が高い利得を受け取ることになる。これにより、各組織が提供する通信帯域の増加が期待できる。また、各組織が通信帯域に対しての利得を増やすためにどの程度通信帯域を提供するかといったゲーム性も生まれる。

4.5.1 通信帯域が実行時間に影響を与える実験結果

本項では、通信帯域が MPC の実行時間に影響を与えるかを確認する。実験は 4.4.1 項の実験と同様に表 1 の機器を使って図 5 の機器構成で行った。

今回の実験では SEPIA のサンプルプログラムである tutorial.jar の実行時間を Surface Pro 4 側のアウトバウンド方向の帯域を 1kbps に制限[13]したうえで計測した。Input Peer（ip）と Privacy Peer（pp）の組み合わせを 5 パターンごと、機器ごとに各 100 回ずつ計測を行った。各パターン、各機器ごとの平均実行時間は表 4 と表 5 の結果となった。

[13]アウトバウンドの帯域制限を Windows 標準機能で行う https://qiita.com/DogFortune/items/70e66f9e5c4c04429565 を参考

表 4: MPC 処理時間計測結果 帯域制限なし

パターン	Surface Pro 4	LIFEBOOK SH54/G	実行時間（秒）
1	ip1, ip2, ip3, pp1, pp2, pp3	なし	10.176
2	ip1, ip2, pp1, pp2, pp3	ip3	10.515
3	ip1, ip2, ip3, pp1, pp2	pp3	10.611
4	ip1, ip2, pp1, pp2	ip3, pp3	10.730
5	なし	ip1, ip2, ip3, pp1, pp2, pp3	10.480

表 5: MPC 処理時間計測結果 帯域制限あり

パターン	Surface Pro 4	LIFEBOOK SH54/G	実行時間（秒）
1	ip1, ip2, ip3, pp1, pp2, pp3	なし	10.182
2	ip1, ip2, pp1, pp2, pp3	ip3	10.514
3	ip1, ip2, ip3, pp1, pp2	pp3	10.657
4	ip1, ip2, pp1, pp2	ip3, pp3	10.810
5	なし	ip1, ip2, ip3, pp1, pp2, pp3	NO DATA

表 4 と表 5 を比較してみる。パターン 1 では Surface Pro 4 のみでの MPC の実行であり、Surface Pro 4 のアウトバウンド方向に帯域制限をかけているので表 5 のほうが遅いものとなっている。パターン 2 は LIFEBOOK SH54/G が 1 つの ip を実行しているが pp から ip への通信回数は ip から pp と比べて少ないので実行時間の違いはほとんど見られなかった。パターン 3 と 4 は LIFEBOOK SH54/G において pp が実行されており pp 間は多くの通信が行われるため実行時間は表 5 のほうが遅くなっている。パターン 5 は、LIFEBOOK SH54/G だけで MPC の実行が完結していること、LIFEBOOK SH54/G には帯域制限を設けていないことから実行時間の計測は行わなかった。

いずれの実行時間も Surface Pro 4 のアウトバウンド方向に 1kbps という帯域制限をかけたにもかかわらず、大きな実行時間の差は見られなかった。これは、SEPIA のサンプルプログラムである tutoria.jar の通信コストが少なかったからだと考えられる。

しかし、この結果から MPC の実行時間には確かにマシン間の通信帯域が関わることが分かった。そのため、4.5 節で提案した通信帯域に対しての利得は 1 つの妥当な利得要素だと考えられる。

4.5.2 通信帯域のモデル

4.5 節において通信帯域に対しての利得を提案した。それは各組織が提供した帯域幅（BW）を測定して利得を配分するものであった。しかし、図 6 において通信経路は AB 間, BC 間, CA 間の 3 つであり、各組織が提供した帯域幅を求めるのは容易ではない。そこで基幹回線を Internet とした図 7 の通信形態モデルを考える。図 7 中の通信経路は、A と Internet 間, B と Internet 間, C と Internet 間であり、Internet と各組織間の Internet 側、及び、Internet 内は十分に高速であるとすると、A の帯域, B の帯域, C の帯域と読み替えることができる。

十分に高速な Internet を基幹回線と想定することで各組織が提供する帯域幅を求めることが容易になる。

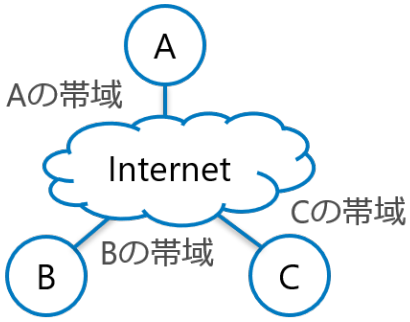

図 7: 基幹回線を考慮した MPC の通信形態

表 6: フェーズごとの MPC 利得要素の有無

利得要素	利得名称	オンラインフェーズ	オフラインフェーズ
計算量（4.2 節）	計算利得（CP）	○	×
事前計算（4.3 節）	事前計算利得（PP）	×	○
提供計算リソース（4.4 節）	提供計算リソース比利得（PRP）	○	○
通信帯域（4.5 節）	帯域幅比利得（BW）	○	○

4.6 MPC 実施に必要な利得要素

4.1 節において MPC 実施に関連した利得要素を紹介し、4.2 節から 4.5 節で各利得要素の詳細を述べた。これまでに考慮した利得要素を荒木らの研究で述べられている MPC の 2 つのフェーズ（オンラインフェーズとオフラインフェーズ）に当てはめると表 6 となる。

オンラインフェーズは MPC 実施そのものであるため利得を考慮するにあたって必須のフェーズである。それに対して、オフラインフェーズ、つまり、事前計算を行うフェーズは能動的な攻撃者に対して安全とするときに必要な MAC 付き 3 つ組を準備するフェーズである。4.3 節で述べたように、MAC 付き 3 つ組の準備には時間がかかるが、MPC 実行時の入力とは無関係なランダムな値であるためオンラインフェーズの実行時間を短縮するために事前に行われる。

しかし 4.3 節において、事前計算の有無に関わらず処理量（計算量と事前計算量）に対しての利得は各組織で等分であることを述べた。計算量は MPC 実施のオンラインフェーズに必須の利得要素であるので、処理量に対する利得は計算利得だけを考えればよいことになる。

したがって、事前計算がある場合とない場合それぞれに必要な利得要素は同一と考えることができ、考慮すべき利得要素は表 7 になる。次節では MPC 実施における利得配分の具体例を紹介する。

4.7 MPC 実施に対しての利得

4.2 節では計算量に対しての利得を、4.4 節では提供計算リソースに対しての利得を提案した。今回、情報統合分析における利得の発生要素を計算量と提供計算リソースだけと仮定し、それらを組み合わせた利得配分手法を提案する。計算量と提供計算リソースの 2 つの利得発生要素を組み合わせたものを計算合計利得 (CBP: Computation Balance Profit) と呼ぶこととし、その配分手法を以下のように 2 つ提案する。

- 計算量に対しての利得と提供計算リソースに対しての利得を等分に与える場合 (CBP_1)

表 7: MPC 実施に必要な利得要素

利得要素	利得名称
計算量（4.2 節）	計算利得（CP）
提供計算リソース（4.4 節）	提供計算リソース比利得（PRP）
通信帯域（4.5 節）	帯域幅比利得（BW）

- 計算量に対しての利得と提供計算リソースに対しての利得の比率を各組織の投票により決定する場合 (CBP_2)

CBP_1 の計算量に対しての利得と提供計算リソースに対しての利得を個別に与える手法は、計算量に対しての利得と提供計算リソースに対しての利得の幅を同じものとし、それらの合計の平均を各組織に配分する単純な手法である。

$$CBP_1(X) = (CP(X) + PRP(X))/2 \tag{5}$$

これまでに求めた $CP(X)$ と $PRP(X)$ の具体的な値を使って $CBP_1(X)$ を求めると以下の値となる。

- $CBP_1(A) = 4/15$
- $CBP_1(B) = 7/15$
- $CBP_1(C) = 4/15$

しかし、式 5 のもと利得の配分を行うと少ない提供計算リソースしか提供できない組織が情報統合分析に参加することが難しくなる。そこで、計算量に対しての利得と提供計算リソースに対しての利得の比率を各組織の投票により決定する手法を CBP_2 として提案する。計算量に対しての利得と提供計算リソースに対しての利得を個別に与えるのではなく各組織がどちらに重きを置くかを各組織がパラメータ ρ_X により宣言するものとする。そして各組織の宣言値をもとに配分のパラメータ ρ を定める。これによって少ない提供計算リソースしか提供できない組織の情報統合分析への参加を促す。$CBP_2(X)$ は $CP(X)$ と $PRP(X)$、さらに配分パラメータ ρ を用いて以下で表される。

$$CBP_2(X) = \rho CP(X) + (1-\rho)PRP(X) \tag{6}$$

ここで、配分パラメータ ρ は各組織が宣言する ρ_X をもとに以下のように求める。

$$\rho = \frac{1}{n}\sum_X \rho_X \tag{7}$$

ここで、n は組織数を表し、ρ_X の範囲は $0 \leq \rho_X \leq 1$ とする。

これまでに求めた $CP(X)$ と $PRP(X)$ の具体的な値を使って $CBP_2(X)$ を求めると以下の値となる。

- $CBP_2(A) = 13/45, \rho_A = 1$
- $CBP_2(B) = 19/45, \rho_B = 0$
- $CBP_2(C) = 13/45, \rho_C = 1$

表 8: MPC 実施により発生する利得

参加組織	A, B, C
計算利得	$CP(A) = 1/3, CP(B) = 1/3, CP(C) = 1/3$
提供計算リソース比利得	$PRP(A) = 1/5, PRP(B) = 3/5, PRP(C) = 1/5$
計算合計利得	$CBP(X) = \rho CP(X) + (1 - \rho)PRP(X)$

4.8 MPC 実施に対しての利得のゲーム理論的な分析

4.2 節で計算量に対しての利得、4.4 節で提供計算リソースに対しての利得、4.7 節でそれぞれの利得の比率の決定方法について紹介し、具体的な利得の値も確認した。本節では、式 6 における ρ の決まり方をゲーム理論を考慮にいれて説明する。これまでの内容を整理すると表 8 になる。

計算合計利得における ρ の決定は各組織が ρ_X を宣言するものであり、各組織ごとの行動は以下のようになる。このとき、各組織の計算利得と提供計算リソース比利得は公開されているものとする。

- 組織 A の戦略: 少ない提供計算リソースを提供し、提供計算リソースに対しての利得の割合（1/5）が計算量に対しての利得の割合（1/3）より小さいため $\rho_A = 1$ を宣言
- 組織 B の戦略: 多くの提供計算リソースを提供し、提供計算リソースに対しての利得の割合（3/5）が計算量に対しての利得の割合（1/3）より大きいため $\rho_B = 0$ を宣言
- 組織 C の戦略: 少ない提供計算リソースを提供し、提供計算リソースに対しての利得の割合（1/5）が計算量に対しての利得の割合（1/3）より小さいため $\rho_C = 1$ を宣言

このときの各組織の利得の配分は以下となる。

- $CBP_2(A) = 13/45, \rho_A = 1$
- $CBP_2(B) = 19/45, \rho_B = 0$
- $CBP_2(C) = 13/45, \rho_C = 1$

最終的に各組織は導き出された $CBP_2(X)$ と一般の MPC に参加するにあたってあらかじめ想定していた利得を比較し、$CBP_2(X)$ が大きければ参加し、$CBP_2(X)$ が小さければ参加しないことになる。

本モデルでは ρ の値を各組織が宣言した ρ_X の平均から決定した。そのため、このモデルでは少ない計算リソースを提供する組織が増えた場合 ρ の値が非常に大きくなってしまう。だが、荒木らの論文 [7] において秘密分散ベースの 3 者マルチパーティ計算が十分実用的であると記述されている。情報統合分析は、3 者による MPC の実用的なサービス形態の一種であったのでパーティ数が 3 であれば実用的であるといえる。そのため ρ の値を各組織が宣言した ρ_X の平均から決定するものとした。

4.8.1 各組織が取る戦略の詳細

4.8 節での各組織が取る ρ_X 戦略は各組織ごとに提供計算リソースに対しての利得と計算量に対しての利得を比べて値の大きいほうの比率を上げるものであった。本項ではその詳細を見てみる。

ρ_X は、$0 \leq \rho_X \leq 1$ の範囲であった。今回 ρ_X は $\{0, 0.5, 1\}$ のいずれかの値を取るとし、式 6 を用い各組織の戦略 ρ_X ごとの各組織の利得 $CBP_2(X)$ を求めると表 9 になる。

表 9: 各組織の戦略ごとの各組織の利得

行番号	ρ_A	ρ_B	ρ_C	$CBP_2(A)$	$CBP_2(B)$	$CBP_2(C)$
1	0.0	0.0	0.0	0.200	0.600	0.200
2	0.0	0.0	0.5	0.222	0.556	0.222
3	0.0	0.0	1.0	0.244	0.511	0.244
4	0.0	0.5	0.0	0.222	0.556	0.222
5	0.0	0.5	0.5	0.244	0.511	0.244
6	0.0	0.5	1.0	0.267	0.467	0.267
7	0.0	1.0	0.0	0.244	0.511	0.244
8	0.0	1.0	0.5	0.267	0.467	0.267
9	0.0	1.0	1.0	0.289	0.422	0.289
10	0.5	0.0	0.0	0.222	0.556	0.222
11	0.5	0.0	0.5	0.244	0.511	0.244
12	0.5	0.0	1.0	0.267	0.467	0.267
13	0.5	0.5	0.0	0.244	0.511	0.244
14	0.5	0.5	0.5	0.267	0.467	0.267
15	0.5	0.5	1.0	0.289	0.422	0.289
16	0.5	1.0	0.0	0.267	0.467	0.267
17	0.5	1.0	0.5	0.289	0.422	0.289
18	0.5	1.0	1.0	0.311	0.378	0.311
19	1.0	0.0	0.0	0.244	0.511	0.244
20	1.0	0.0	0.5	0.267	0.467	0.267
21	1.0	0.0	1.0	0.289	0.422	0.289
22	1.0	0.5	0.0	0.267	0.467	0.267
23	1.0	0.5	0.5	0.289	0.422	0.289
24	1.0	0.5	1.0	0.311	0.378	0.311
25	1.0	1.0	0.0	0.289	0.422	0.289
26	1.0	1.0	0.5	0.311	0.378	0.311
27	1.0	1.0	1.0	0.333	0.333	0.333

表 9 を用い、組織 A が取る戦略を見る。ゲーム理論では相手の戦略ごとに自身の戦略を立てていく。

- $(\rho_B, \rho_C) = (0, 0)$ のとき
  行番号 1, 10, 19 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択

- $(\rho_B, \rho_C) = (0, 0.5)$ のとき
  行番号 2, 11, 20 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択
- $(\rho_B, \rho_C) = (0, 1)$ のとき
  行番号 3, 12, 21 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択
- $(\rho_B, \rho_C) = (0.5, 0)$ のとき
  行番号 4, 13, 22 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択
- $(\rho_B, \rho_C) = (0.5, 0.5)$ のとき
  行番号 5, 14, 23 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択
- $(\rho_B, \rho_C) = (0.5, 1)$ のとき
  行番号 6, 15, 24 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択
- $(\rho_B, \rho_C) = (1, 0)$ のとき
  行番号 7, 16, 25 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択
- $(\rho_B, \rho_C) = (1, 0.5)$ のとき
  行番号 8, 17, 26 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択
- $(\rho_B, \rho_C) = (1, 1)$ のとき
  行番号 9, 18, 27 の $CBP_2(A)$ を比較して最も高くなる戦略 $\rho_A = 1$ を選択

この結果、相手のどんな戦略の組に対しても組織 A は $\rho_A = 1$ を選択することになる。このような戦略は支配戦略[14]と呼ばれる。同様に組織 B が取る戦略を求めると $\rho_B = 0$、組織 C が取る戦略を求めると $\rho_C = 1$ となり、どちらも支配戦略となっていることが分かる。すべてのプレイヤーに支配戦略が存在するとき、その支配戦略の組み合わせを支配戦略均衡と呼ぶ。支配戦略均衡はゲームの解であるので、$(\rho_A, \rho_B, \rho_C) = (0, 1, 0)$ が表 8 の値のもとでのゲームの解となり、これは 4.8 節の結果と一致する。

4.9 MPC 参加者が提供するリソースの動的変更

4.7 節で MPC 実施に対しての利得、4.8 節で MPC 実施に対しての利得のゲーム理論的な分析を述べた。MPC 実施に対しての利得を各組織が提供するリソースから算出するものであった。しかし、算出された値を見てから自組織が受け取る利得の割合を高めたいと思う組織の存在も考えられる。そこで、本節では MPC 参加者が提供するリソースを動的に変更する場合を考察する。

表 8 のもと各祖組織の利得を求めると以下の通りであった。

- $CBP_2(A) = 13/45, \rho_A = 1$
- $CBP_2(B) = 19/45, \rho_B = 0$
- $CBP_2(C) = 13/45, \rho_C = 1$

この算出された利得を見て組織 A は提供計算リソースの性能を高め自組織の利得を増やす (例えば、$PR(A) = 6$) 行動をとるとする。このとき、各組織の提供計算リソース比利得は以下になり

[14] あるプレイヤーのある戦略が、他のプレイヤーのすべての戦略に対して、他のどんな戦略よりも高い利得を与えるときの戦略

- $PRP(A) = 6/10$
- $PRP(B) = 3/10$
- $PRP(C) = 1/10$

各組織が取る ρ_X 戦略は以下のように変わることになる。

- 組織 A の戦略: 提供計算リソースに対しての利得の割合 (6/10) が計算量に対しての利得の割合 (1/3) より大きいため $\rho_A = 0$ を宣言
- 組織 B の戦略: 提供計算リソースに対しての利得の割合 (3/10) が計算量に対しての利得の割合 (1/3) より小さいため $\rho_B = 1$ を宣言
- 組織 C の戦略: 提供計算リソースに対しての利得の割合 (1/10) が計算量に対しての利得の割合 (1/3) より小さいため $\rho_C = 1$ を宣言

その結果各組織が受け取る利得の割合を式 6 から算出すると以下となる。

- $CBP_2(A) = 38/90, \rho_A = 0$
- $CBP_2(B) = 29/90, \rho_B = 1$
- $CBP_2(C) = 23/90, \rho_C = 1$

このように他の組織の提供するリソースを見て自組織の利得の割合を高めるといった行動も現実には考えられる。このような行動はゲーム理論でいう展開形ゲーム[15]で分析することができる。前節までは各組織が提供するリソースや ρ_X の宣言は一度きりであったが、展開形ゲームを採用することで複数回の宣言が行われる場合の分析もできるようになり、より柔軟な利得配分の分析が可能になる。

4.10 協力ゲームによる利得配分

4.1 節から 4.9 節までの内容は、ゲーム理論の非協力ゲームの考えのもと考察を行った。個人を MPC に参加する組織、行動の選択肢として提供する計算資源の性能と式 7 の ρ_X の宣言の行動を与え、各組織における利得を非協力ゲームのもと分析するものであった。しかし、参加組織が 3 者以上の場合は協力ゲームとして分析を行うこともできる。

そこで本節では、MPC に参加する 3 つの組織の組織間提携を考える。協力ゲームでは、各提携が組まれたときの利益は各組織の行動に関係なく既に与えられており、その下でどのような提携が組まれるか、または提携を考えたときに自身の組織に配分される利益はどのような結果となるかが焦点となる。協力ゲームの理論である提携形ゲーム（特性関数形ゲーム）は、以下で構成される。

- プレイヤーの集合 N
- 提携 S : N の非空の部分集合
- 特性関数 $v(S)$: 提携 S により発生する利得を定めた関数

[15] あるタイミングで何人かのプレイヤーが同時に行動したり、順番に行動したり、同時の行動と交互の行動が混合して行われるような複雑なゲーム

表 10: シャープレイ値による解

全体提携の形成過程	限界貢献度		
	A	B	C
A ← B ← C	0	0	1
A ← C ← B	0	1	0
B ← A ← C	0	0	1
B ← C ← A	1	0	0
C ← A ← B	0	1	0
C ← B ← A	1	0	0
シャープレイ値	1/3	1/3	1/3

MPC では専門組織 3 つの参加が適しているという議論をもとに、$v(S)$ は 3 つの組織が協力するときだけに発生するように設定した。協力による発生する利得の全体を 1 としたときの、MPC における特性関数形ゲームは以下となる。

- プレイヤーの集合 N : $N = \{A, B, C\}$
- 提携 S :
  $S = \{\{A\}, \{B\}, \{C\}, \{A, B\}, \{B, C\}, \{C, A\}, \{A, B, C\}\}$
- 特性関数 $v(S)$:
  $v(\{A\}) = v(\{B\}) = v(\{C\}) = v(\{A, B\}) = v(\{B, C\}) = v(\{C, A\}) = 0, v(\{A, B, C\}) = 1$

特性関数形ゲームの解としては、コア、仁、シャープレイ値があるが、本項ではシャープレイ値により特性関数形ゲームの解を求める。

シャープレイ値は、全体の提携に対してのプレイヤーの順序をすべて考えて、各順序での限界貢献度の平均値をとったものである。今回の設定でのシャープレイ値での解の導出過程は表 10 となる。任意の $i \in N$ と任意の部分集合 $S \subseteq N \backslash \{i\}$ に対して $v(S \cup \{i\}) - v(S)$ を S に対するプレイヤー i の限界貢献度と呼ぶ。その結果、各組織の利得は 1/3 となることが示された。これは特性関数が $\{A, B, C\}$ の時のみ利得の発生が定義されているためであり、この特性関数の変化により各組織の利得は変化する。その変化を実社会において、あるいはユースケースにおいてどう定めるかについても検討の余地があるだろう。

$\{A, B, C\}$ だけが 1 となるために自明な結果であるが、今後の MPC 方式の発展によってはこの前提が崩れる可能性がある。その場合、この結果は自明ではなくなる。しかしその場合も分析は本節で示したようにシャープレイ値に沿って行われて結果が得られる。

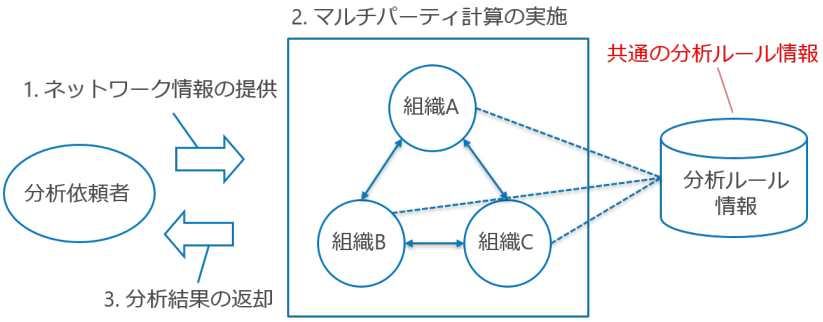

図 8: 先行研究（Liu ら, 作本ら）のシステムモデル

5　プライバシ保護型リスク分析におけるシステムモデル

5.1　Liu ら, 作本らの研究でのシステムの問題点

3.3 節で紹介した Liu ら [1] や作本ら [2] の先行研究でのプライバシ保護型リスク分析のプロトタイプシステムは、Risk Analysis Rule Provider が Computation Engine Provider を兼ねる場合において実装されている。システムモデルは図 8 となる。

先行研究（Liu ら, 作本ら）のシステムモデルの一連の処理の流れは以下となる。

1. ネットワーク情報の提供
   分析依頼者が自組織のネットワーク情報をリスク分析を行う複数の専門組織（組織 A, B, C）に対して秘密分散して提供

2. マルチパーティ計算の実施
   専門組織（組織 A, B, C）は受け取った秘密分散されたネットワーク情報と組織間で共通の分析ルール情報を用いてマルチパーティ計算によるリスク分析を実施

3. 分析結果の返却
   マルチパーティ計算によって得られた結果を分析依頼者に対して返却

先行研究（Liu ら, 作本ら）のシステムモデルでは、複数の専門組織（組織 A, B, C）間で分析ルール情報を共有している。複数の専門組織は実社会においては独立して運営されるセキュリティ専門組織であると考えられるため、分析ルール情報の共有はビジネス上考えにくい。分析ルール情報をお互いが開示することなく協力可能なシステムモデルが求められる。

5.2　組織ごとに分析ルール情報を持つ場合のシステムモデル

5.1 節で述べたように、複数の専門組織は実社会においては独立して運営されるセキュリティ専門組織であると考えられる、そのため分析ルール情報の共有は考えにくい。この問題点を考慮し本節では分析ルール情報をそれぞれの専門組織が持つ新たなシステムモデルを提案する。

本研究で提案するシステムモデルは図 9 である。システムモデルの一連の処理の流れは以下となる。

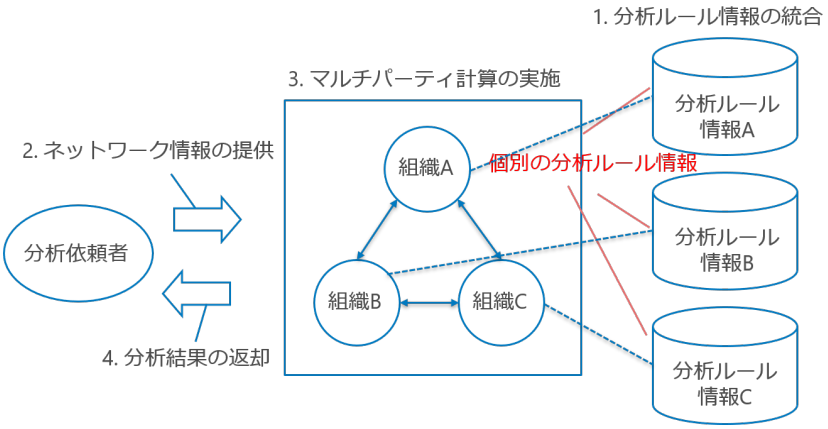

図 9: 本研究で提案するシステムモデル

1. 専門組織間の分析ルール情報の統合
   分析ルール情報を専門組織ごとに持っているため、リスク分析を行うにあたって分析ルール情報を統合

2. ネットワーク情報の提供
   分析依頼者が自組織のネットワーク情報をリスク分析を行う複数の専門組織（組織 A, B, C）に対して秘密分散して提供

3. マルチパーティ計算の実施
   専門組織（組織 A, B, C）は受け取った秘密分散されたネットワーク情報と組織間で共通の分析ルール情報を用いてマルチパーティ計算によるリスク分析を実施

4. 分析結果の返却
   マルチパーティ計算によって得られた結果を分析依頼者に対して返却

先行研究と本研究のシステムモデルでの異なる処理は、専門組織ごとに分析ルール情報を持っているため分析ルール情報を統合する点である。したがって、本研究で提案するシステムモデルを実社会で実現するにあたっては、専門組織間の分析ルール情報統合の課題を解決する必要がある。

5.3 分析ルール情報の統合

提案するシステムモデルでは分析ルール情報を専門組織ごとに持っているため、リスク分析を行うにあたって分析ルール情報を統合する必要がある。しかし、分析ルール情報はそれぞれの専門組織が持つ独自の情報であり、専門組織は分析ルール情報を公開したくないと考えるのが一般的である。そのため、分析ルール情報を平文で統合することは考えにくい。そこで分析ルール情報を秘匿化したまま統合が可能かの調査を行った。Many らは、MPC とブルームフィルタを組み合わせることで秘匿集合演算が可能なことを示した [14]。しかし、2.3 節で示したようにブルームフィルタ表現をリスク分析における分析ルール情報としては扱うことは困難である。分析ルール情報の全世界共通の採番などがされていない限りどのような分析ルール情報が統合されブルームフィルタ表現として格納されているのかを確認できないためである。

表 11: 実行環境

プロセッサ	Intel®Core^{TM}i7-6650U CPU @ 2.20GHz
RAM	8.00 GB
Java	jdk 1.8.0_161

表 12: 利用ルール数と分析ルール情報統合の実行時間（秒）

Rules Sum	25	30	60	90	120	150
A Rules	8	10	20	30	40	50
B Rules	8	10	20	30	40	50
C Rules	9	10	20	30	40	50
実行時間	16.98	16.87	16.96	16.79	16.90	16.89

数論的な視点での要素を秘匿化したままの集合演算はさまざまなアプローチで研究がされているが、リスク分析のルール群を要素としてその和集合や積集合の秘匿な取得はより柔軟な手法が求められる。本研究では、ルール統合の手法自体は議論せず、後述するように試作実験では分析ルール情報に全世界共通の採番がされていることを前提に開発を行った。

5.3.1 採番した分析ルール情報の統合

作本らの研究 [2] では、プライバシ保護型リスク分析システムの実装と評価が行われている。しかし、先行研究（Liu ら, 作本ら）のシステムモデルでは複数の専門組織間（組織 A, B, C）で分析ルール情報が共有されている点が問題であった。作本らの研究のシステムでは分析ルール情報をハードコーディングしている。本研究では、ハードコーディングされている分析ルール情報（MulVAL の Interaction Rules）を採番し、各専門組織は採番された番号を宣言する。宣言された番号を 5.3 節の方法で統合したものを使用可能な分析ルール情報とする。採番した分析ルール情報は MulVAL のオープンソース実装に含まれている 25 種類の Interaction Rules である。

5.3.2 分析ルール情報統合のパフォーマンス評価

分析ルール情報の統合にかかる実行時間（単位:秒）を表 11 の環境のもと、分析ルール情報の合計数ごとに 100 回ずつ計測したところ表 12 の結果が得られた。Rules Sum は統合された分析ルール情報の合計数、X Rules は組織 X が使用を宣言した分析ルール情報の数である。この実行結果から採番された分析ルール情報の統合にかかる時間は分析ルール情報の数が増加してもほぼ変化がないことが分かった。

また、プライバシ保護型リスク分析の実施における実行時間についても表 11 の環境のもと同条件で 100 回ずつ計測した。表 13 はその平均実行時間 (単位:秒) である。比較対象として、先行研究である作本らの研究のシステムを挙げた。提案システムは、先行研究のシステムを分析ルール情報の統合結果を参照し、分析ルール情報の使用権限を確認しながら実行するように変更したものである。実装においては使用権限のフラグを確認するように変更しただけであるため、実行時間が大きく変化することはないことが結果からも見て取れる。

表 13: プライバシ保護型リスク分析の実行時間（秒）

	作本ら [2] のシステム	提案システム
実行時間	19.88	19.73

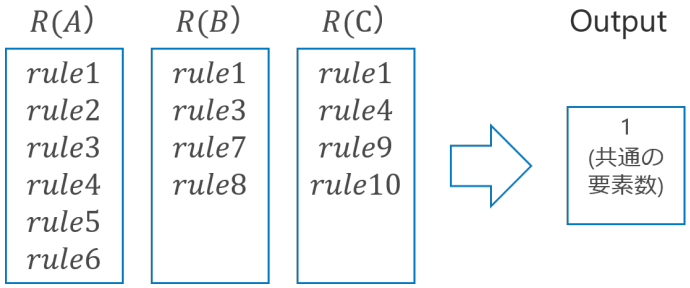

図 10: Private Set Intersection Cardinality の実行例

5.3.3 各組織が提供した分析ルール情報量を求める手法

本研究で提案するシステムモデルでは分析ルール情報を専門組織ごとに持っているため、リスク分析を行うにあたって分析ルール情報を統合する必要があった。その際、各専門組織がどれぐらいの量、分析ルール情報を提供したかの情報が 6 章で紹介するプライバシ保護型リスク分析で発生する利得を考える上では必要になる。

分析ルール情報は各組織に独自のルールも存在するため、お互いに情報を秘匿化したまま共通する分析ルール情報の数を計算する必要がある。本項では SEPIA の Private Set Intersection Cardinality[14] を用いることで各組織（組織 A, B, C）がお互いに分析ルール情報を漏らさずに各組織が独自に持つ分析ルール情報量を求める手法を提案する。

例として、各組織が持つ分析ルール情報が以下の時を考える。

- $R(A) = \{rule1, rule2, rule3, rule4, rule5, rule6\}$
- $R(B) = \{rule1, rule3, rule7, rule8\}$
- $R(C) = \{rule1, rule4, rule9, rule10\}$

このとき、各組織だけが持つ分析ルール情報は以下のように表すことができる。

- $|R(A)| = |\{rule2, rule5, rule6\}| = 3$
- $|R(B)| = |\{rule7, rule8\}| = 2$
- $|R(C)| = |\{rule9, rule10\}| = 2$

しかし、各組織だけが持つ分析ルール情報の量を求める際には、各組織独自の分析ルール情報を漏らさずに行わなければならない。そこで、SEPIA の Private Set Intersection Cardinality を使い入力する分析ルール情報を秘匿化したまま各組織で共通する要素数を求める。Private Set Intersection Cardinality の実行例は図 10 である。

図 10 では、3 つの組織があり、各組織の入力情報において共通する要素数が出力されている。Private Set Intersection Cardinality を使うことで各組織だけが持つ分析ルール情報の量を以下のようにして求めることができる。

- $|R(A)| = |A - (A \cap B + A \cap C) + A \cap B \cap C|$
- $|R(B)| = |B - (B \cap C + B \cap A) + A \cap B \cap C|$
- $|R(C)| = |C - (C \cap A + C \cap B) + A \cap B \cap C|$

A だけが持つ分析ルール情報の量を求めるためには、次の流れになる。まず自組織 A が持つ分析ルール情報を数える。次に、組織 A と B が共通で持つ分析ルール情報の数を上述した Private Set Intersection Cardinality を用いて求める、同様に組織 A と C 間、及び、組織 A, B, C 間で共通する分析ルール情報の数を求める。最終的にそれらの値を使い組織 A だけが持つ分析ルール情報の量を求めることができる。

5.4 プライバシ保護型リスク分析実施に対しての利得の検討

4 章では MPC における一般的な利得についての議論を行った。本章で議論しているプライバシ保護型リスク分析の利得を考える際には、計算量と提供計算リソースのほかに、分析ルール情報の質や量といった指標も利得に影響すると考えるべきである。そこでその検討を新たに 6 章で行うこととする。

6 ゲーム理論を適用したプライバシ保護型リスク分析参加者利得の検討

5 章にてプライバシ保護型リスク分析における新たなシステムモデルを提案した。本章では、リスク分析に必要な分析ルール情報と 4 章で紹介した MPC へのゲーム理論を適用した利得を考慮し、プライバシ保護型リスク分析の実施により発生する利得を検討する。

6.1 参加組織の提供リソース

プライバシ保護型リスク分析に参加するのは複数の専門組織である。この複数の専門組織には組織ごとに分析手法や得意分野などの提供リソースがあると考えられる。本研究ではマルチパーティ計算に参加する組織の提供リソースとして以下の 2 つを挙げる。

- 分析ルール情報
  専門組織ごとに持つ独自の分析ルール情報
- 計算資源
  MPC を行う計算資源（4 章で紹介）

これらの提供リソースを考慮したプライバシ保護型リスク分析に参加する専門組織の行動としては以下の 3 つが考えられる。

- 分析ルール情報と計算資源の両方を提供
- 分析ルール情報は提供しないが、計算資源は提供
- 計算資源は提供しないが、分析ルール情報を提供

本章では 4 章と同様にゲーム理論の考えを導入してプライバシ保護型リスク分析へのゲーム理論を適用した利得の検討を行う。

6.2 分析ルール情報量に対しての利得

本研究で提案するシステムモデルでは専門組織ごとに独自の分析ルール情報を持つ。組織ごとに提供する分析ルール情報の量は異なるため、各組織は提供した分析ルール情報の量に応じて利得を受け取ることになる。組織 X が提供する分析ルール情報を $R(X)$, 分析ルール情報量を $|R(X)|$ とする。

分析ルール情報の量の割合に応じて利得を配分する式を KP: Knowledge Profit とする。$KP(X)$ は相対的な値であり、参加組織全体が持つ分析ルール情報量のうち組織 X が持つ分析ルール情報量の割合を示すものである。このとき各組織が得る利得は全体を 1 として式 8 となる。

$$KP(X) = |R(X)| / \sum_{O} |R(O)| \tag{8}$$

例えば、参加組織 A, B, C がそれぞれ提供する分析ルール情報が次の場合

- $R(A) = \{rule1, rule2, rule3, rule4, rule5\}$

- $R(B) = \phi$
- $R(C) = \phi$

各組織が提供した分析ルール情報量は 5.3.3 項の手法を用いて以下となる。

- $|R(A)| = |\{rule1, rule2, rule3, rule4, rule5\}| = 5$
- $|R(B)| = |\phi| = 0$
- $|R(C)| = |\phi| = 0$

そのため、分析ルール情報を提供するにあたって発生する各組織の利得は以下となる。

- $KP(A) = 1$
- $KP(B) = 0$
- $KP(C) = 0$

しかし、式 8 では、分析には無意味な分析ルール情報（以下、ダミー情報）も利得配分に関係することになる。ダミー情報を含んだ利得配分を防ぐには、分析に使える分析ルール情報は何か、つまり、使える分析ルール情報の判別が必要になる。本研究では、これらの内容は範囲外とするが、現時点での考察を 7.2 節で述べる。

6.3 計算資源に対しての利得

本研究で提案するシステムモデルでは専門組織はマルチパーティ計算を実施するための計算資源を提供する。それぞれの組織は、マルチパーティ計算に参加するにあたって提供した計算資源に応じて利得を受け取る。計算資源により発生する利得は 4 章で紹介した。本節では、式 6 を用い、各組織におけるマシンの提供計算リソースが等しい $\rho = 1$ の場合で考える。このとき式 6 は以下の式として表せる。

$$CBP_2(X) = CP(X) = 1/n \tag{9}$$

そのため、計算資源を提供するにあたって発生する各組織の利得は以下となる。

- $CBP_2(A) = CP(A) = 1/3$
- $CBP_2(B) = CP(B) = 1/3$
- $CBP_2(C) = CP(C) = 1/3$

6.4 プライバシ保護型リスク分析実施に対しての利得

6.2 節では分析ルール情報量に対しての利得を、6.3 節では計算資源に対しての利得を提案した。さらにプライバシ保護型リスク分析における利得の発生要素を分析ルール情報と計算資源だけと仮定し、それらを組み合わせた利得配分手法を提案する。分析ルール情報と計算資源の 2 つの利得発生要素を組み合わせたものを合計利得 (BP: Balance Profit) と呼ぶこととし、その配分手法を以下のように 2 つ提案する。

- 分析ルール情報に対しての利得と計算資源に対しての利得を個別に与える場合（BP_1）
- 分析ルール情報に対しての利得と計算資源に対しての利得の比率を各組織の投票により決定する場合（BP_2）

BP_1 の分析ルール情報に対しての利得と計算資源に対しての利得を個別に与える手法は、分析ルール情報に対しての利得と計算資源に対しての利得の幅を同じものとし、それらの合計の平均を各組織に配分する単純な手法である。

$$BP_1(X) = (KP(X) + CBP(X))/2 \tag{10}$$

これまでに求めた $KP(X)$ と $CBP_2(X)$ の値を使って $BP_1(X)$ を求めると以下の値となる。

- $BP_1(A) = 4/6$
- $BP_1(B) = 1/6$
- $BP_1(C) = 1/6$

しかし、式 10 のもと利得を配分すると、分析ルール情報は提供しないが計算資源は提供する組織や、計算資源は提供しないが分析ルール情報を提供する組織がプライバシ保護型リスク分析に参加することが難しくなる。そこで、分析ルール情報に対しての利得と計算資源に対しての利得の比率を各組織の投票により決定する場合を提案する。分析ルール情報に対しての利得と計算資源に対しての利得を個別に与えるのではなく各組織がどちらに重きを置くかを各組織がパラメータ ρ_X により宣言するものとする。これによって分析ルール情報か計算資源のどちらかしか提供しない組織のプライバシ保護型リスク分析への参加を促す。ρ の決定方法は式 7 と同じとする。

$$BP_2(X) = \rho KP(X) + (1-\rho)CBP(X) \tag{11}$$

これまでに求めた $KP(X)$ と $CBP_2(X)$ の具体的な値を使って $BP_2(X)$ を求めると以下の値となる。

- $BP_2(A) = 5/9, \rho_A = 1$
- $BP_2(B) = 2/9, \rho_B = 0$
- $BP_2(C) = 2/9, \rho_C = 0$

6.5 プライバシ保護型リスク分析実施に対しての利得のゲーム理論的な分析

前節までに分析ルール情報量に対しての利得、計算資源に対しての利得、それぞれの利得の比率の決定方法について紹介し、具体的な利得の値も確認した。本節では、式 11 における ρ の決まり方をゲーム理論を考慮にいれて説明する。これまでの内容を整理すると表 14 になる。

合計利得における ρ の決定は各組織が ρ_X を宣言するものであり、各組織ごとの行動は以下のようになる。このとき、各組織の分析ルール情報利得と計算利得は公開されているものとする。

- 組織 A の戦略: 分析ルール情報量に対しての利得の割合（1）が計算資源に対しての利得の割合（1/3）より大きいため $\rho_A = 1$ を宣言

表 14: プライバシ保護型リスク分析により発生する利得

参加組織	A, B, C
分析ルール情報利得	$KP(A) = 1, KP(B) = 0, KP(C) = 0$
計算合計利得	$CBP(A) = CBP(B) = CBP(C) = 1/3$
合計利得	$BP(X) = \rho KP(X) + (1 - \rho)CP(X)$

- 組織 B の戦略: 分析ルール情報量に対しての利得の割合（0）が計算資源に対しての利得の割合（1/3）より小さいため $\rho_B = 0$ を宣言
- 組織 C の戦略: 分析ルール情報量に対しての利得の割合（0）が計算資源に対しての利得の割合（1/3）より小さいため $\rho_C = 0$ を宣言

このときの各組織の利得の配分は以下となる。

- $BP_2(A) = 5/9, \rho_A = 1$
- $BP_2(B) = 2/9, \rho_B = 0$
- $BP_2(C) = 2/9, \rho_C = 0$

最終的に各組織は導き出された $BP_2(X)$ とプライバシ保護型リスク分析に参加するにあたってあらかじめ想定していた利得を比較し、$BP_2(X)$ が大きければ参加し、$BP_2(X)$ が小さければ参加しないことになる。

本モデルでは ρ の値を各組織が宣言した ρ_X の平均から決定した。そのため、このモデルでは計算資源だけを提供する組織が増えた場合 ρ の値が非常に小さくなってしまう。だが、荒木らの論文 [7] において秘密分散ベースの 3 者マルチパーティ計算が十分実用的であると記述されている。情報統合分析は、3 者による MPC の実用的なサービス形態の一種であったのでパーティ数が 3 であれば実用的であるといえる。そのため ρ の値を各組織が宣言した ρ_X の平均から決定するものとした。

7　今後の課題

7.1　分析ルール情報そのものの統合

5.2 節にて分析ルール情報をそれぞれの専門組織が持つ新たなシステムモデルを提案し、5.3 節で各専門組織が持つ簡易的な分析ルール情報の統合を示した。簡易的な分析ルール情報の統合とは、分析ルール情報に全世界共通の採番がされているとの前提のもと各組織が採番された番号を宣言し、宣言された番号を統合するものであった。

しかし、分析ルール情報に対しての全世界共通の採番は考えにくい、そのため分析ルール情報そのものを秘匿化したまま統合する必要がある。このとき、統合した情報がリスク分析の入力情報として使えることが必須となる。

7.2　分析ルール情報の質

6.2 節にて分析ルール情報の提供に対しての利得を述べた。それは、各組織ごとに提供する分析ルール情報の量が異なるため、各組織が提供した分析ルール情報の量に応じた利得を与えるものであった（式 8)。

しかし式 8 の利得配分では、分析に使えない情報（以下、ダミー情報）に対しても利得が与えられることになる。これによりダミー情報の量が多ければ多いほど提供した組織が受け取る利得の割合が大きくなってしまうという問題が発生する。

そのため、分析ルール情報の質を考える必要がある。分析ルール情報の質とは、リスク分析に使える情報とダミー情報を判断する指標である。分析ルール情報の質を明確化できれば、提供した分析ルール情報の量に対して利得を与えるのではなく、提供した分析ルール情報の質に対して利得を与えることができると考える。

7.3　事前計算がある MPC の実行

本研究で使用した MPC platform は 2.1.1 項で説明した SEPIA であった。SEPIA では、事前計算は行われない。そのため、本研究では実際に事前計算がある場合の MPC の実行は行えていない。4.3 節で述べた事前計算量に対しての利得は事前計算がある場合の MPC の論文を調査し考察したものである。

事前計算がある MPC の実行を実際に行うことで知見を得られ、事前計算に対する利得の考察が深まると考える。

7.4　利得配分方法

7.4.1　利得要素の重要度

表 8 にて MPC 実施により発生利得、表 14 にてプライバシ保護型リスク分析実施により発生する利得をまとめている。

表 14 の分析ルール情報利得と計算合計利得の 2 つの利得要素がある。分析ルール情報利得は各組織が持つ独自の分析ルール情報のノウハウである。一方、計算合計利得は計算資源さえあればよい。現在、計算資源はアマゾンウェブサービス (AWS) などで比較的簡単に準備することができる。

これらから利得要素の重要度を考慮した配分方式も必要になってくると考えられる。

7.4.2 利得情報の非公開

本研究では MPC に参加する組織が各利得要素をどれだけ提供したかを組織間で公開するものとし議論を進めてきた。しかし、情報を公開しないで参加したいという組織も考えることができる。

そのため、利得情報を非公開とし利得を配分する方式も検討する必要があると考える。

7.5 通信帯域の計測方法

4.5.1 項にて通信帯域が実行時間に影響を与える実験を実機を用いて行った。そこでは、帯域制限をしたうえで SEPIA のサンプルプログラム（tutorial.jar）の実行時間を計測したが大きな差は見られなかった。これは tutorial.jar の通信コストが少なかったからだと考えられる。

このことから、通信帯域が実行時間に与える影響を見るために、通信コストが大きいプログラムを実行する必要があると考える。

7.6 分析結果のフィードバック

図 8 や図 9 にあるようにプライバシ保護型リスク分析では分析結果のフィードバックが行われる。分析ルール情報が共有されている先行研究のシステムモデルでは MulVAL のルールの description をフィードバックをしている。一方で、本研究で提案した分析ルール情報を個別に持つ場合のフィードバックはまだ考えられていない。本研究で実装したプログラムは全世界共通の分析ルール番号情報のもと分析ルール情報を統合をしており、それは全世界共通なので分析者側全員が知っているため、先行研究のシステムモデルと同じフィードバックをしているからである。

分析ルール番号情報の統合ではない場合、どの組織も分析ルール情報を漏らしたくないのでどのような情報をフィードバックするかを考える必要がある。

8　まとめ

本論文では、MPC の実社会での実施方法の検討の 1 つとして、MPC を行う各組織が得る利益に着目し、その利益分配の方法についてゲーム理論を適用したものを提案した。それは、MPC に参加する複数の組織の特性を考慮し、各組織の行動をゲーム理論を用いた分析をすることで社会的側面の議論が可能となるものであった。本研究の提案手法により、MPC 実施に際して各組織が提供する計算量や事前計算量、提供計算リソース、通信帯域といった利得要素をもとにした利益の分配を検討することが可能になる。

さらに、MPC のユースケースとして情報統合分析に焦点を当て、複数の組織が協力して MPC を実施するモデルの提案を行った。そして、情報統合分析の応用的なモデルとして、プライバシ保護型リスク分析へのゲーム理論を適用した利得の配分方法を提案した。本研究の提案手法により、計算量や事前計算量、提供計算リソース、通信帯域、各組織が持つ分析ルール情報といった利得要素をもとにプライバシ保護型リスク分析実施による利益の分配を検討することが可能になる。そしてプライバシ保護型リスク分析というユースケースに焦点を当てて議論をすることで、その利益分配の合理性を議論することを可能にした。

参考文献

[1] Yu Liu, Nasato Goto, Akira Kanaoka, Eiji Okamoto. Privacy Preserved Rule-Based Risk Analysis through Secure Multi-party Computation. Asia Joint Conference on Information Security, 2015.

[2] Soushirou Sakumoto, Akira Kanaoka. Improvement of Privacy Preserved Rule-Based Risk Analysis via Secure Multi-Party Computation. Asia Joint Conference on Information Security, 2017.

[3] Martin Burkhart, Mario Strasser, Dilip Many, Xenofontas Dimitropoulos. SEPIA: Privacy-Preserving Aggregation of Multi-Domain Network Events and Statistics . USENIX Security, 2010.

[4] Adi Shamir. How to share a secret. Communications of the ACM, 1979.

[5] Ivan Damgard, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, Nigel P. Smart. Practical covertly secure MPC for dishonest majorityor: breaking the SPDZ limits. European Symposium on Research in Computer Security, 2013.

[6] Burton H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. ACM, 1970.

[7] 荒木俊則, 五十嵐大, 高橋克巳, 竹之内隆夫, ティブシメディ, 花岡悟一郎, 古川潤. 秘密計算の実用可能性, 暗号と情報セキュリティシンポジウム（SCIS）, 2018.

[8] Katz, Jonathan. Bridging game theory and cryptography: Recent results and future directions. Theory of Cryptography Conference, 2008.

[9] S. Dov Gordon, Jonathan Katz. Rational Secret Sharing, Revisited. International Conference on Security and Cryptography for Networks, 2006.

[10] Xinming Ou, Sudhakar Govindavajhala, Andrew W. Appel. MulVAL: A Logic-based Network Security Analyzer. USENIX Security, 2005.

[11] Assaf Ben-David, Noam Nisan, Benny Pinkas. FairplayMP - A System for Secure Multi-Party Computation. ACM, 2008.

[12] 濱田浩気, 菊池亮. 事前計算が効率的で不正者が多くても安全なマルチパーティ計算. コンピュータセキュリティシンポジウム（CSS）, 2015.

[13] 濱田浩気, 木村映善, 菊池亮, 千田浩司, 岡本和也, 真鍋史朗, 黒田知宏, 松村泰志, 武田理宏, 三原直樹. 秘密計算による分散医療統計システムの実装評価. コンピュータセキュリティ研究会（CSEC）, 2016

[14] Dilip Many, Martin Burkhart, Xenofontas. Fast Private Set Operations with SEPIA. ETZ G93, 2012.