2ZD-05

# コンテナオーケストレーション技術を用いた秘密分散ベースでのマルチパーティ計算システムの開発

橋本辰浩† 金宰郁†

松蔭大学 観光メディア文化学部 メディア情報文化学科†

## １．はじめに

昨今セキュリティの強化を目的として秘密分散に関する研究が行われてきた。近年ではこれを商用サービスとする企業も出てきたが、その顧客の多くは機微データを扱う金融機関などの大企業に限定されていて、コスト面のみならず担当者にはセキュリティに関する高い技術力が要求されることもあり導入が難しいケースも存在すると考える。そこで、本研究では秘密分散をベースとしたマルチパーティ計算システムの開発を行う。本稿では概要とシステムの構成や課題などについて記す。

## ２．本システムに期待できる効果について

これまでにも本研究と類似したシステムは存在しているが、煩雑な環境構築を行わなければならないものが多い。そこで本システムは、計算実行部だけでなく全てを Docker コンテナとして構築し、システムのビルドを簡便に行えるようにすることで容易に導入が行えるのではないかと考えている。またコンテナの運用は Kubernetes といったコンテナオーケストレーション技術を用いることでスケーリングやバックアップの運用が行いやすくなる。

## ３．開発環境について

実装にあたって使用するプログラミング言語には Rust を採用した。Rust を用いることによってメモリリークなどのセキュリティに影響を及ぼす原因となりうるものを最小限に抑えることを目的としている。実行速度の観点からみても比較的高速なことも利点となり得ると考えている。そのほか、Docker と Kubernetes の実行環境として Amazon Web Service（AWS）を活用する。

## ４．本システムの各部構成について

本システムの全体的な概念図について図１に示す。

以下に各部の役割や用いる技術について説明する。

（１）秘密分散処理・復元実行部

今回は Shamir による$(k,n)$しきい値秘密分散法を用いることとした[1]。本方式は「秘密分散処理」と「復元処理」の操作があり、「秘密分散処理」に関して、秘密情報$m$を入力として受け取り、$n$個のシェア$s_1, s_2, \ldots, s_n$を出力する。「復元処理」について、$n$個のシェアのうち任意の$k$個のシェアを集めることによって、$m$を復元し出力する。そのため、$k$個未満のシェアが流出したとしても復元が不可能であり、秘密情報を解読されないという特徴を持っている。本システムにおいては秘密分散処理専用コンテナを用意し、秘密情報を Web API を通して入力することによって実現し、出力されたシェアを次のマルチパーティ計算実行部の入力とする。マルチパーティ計算後に復元処理を行うコンテナも提供し、クライアントに計算結果を返す役割も担う。

（２）マルチパーティ計算実行部

マルチパーティ計算は、計算に関わる情報を持っている参加者間でシェアのみを共有し加算や乗算などの計算を行う。秘密情報そのものを利用する必要のない方式であり、計算結果のみ復元を行い解読することが可能である。計算手法として主に Yao による Garbled Circuit Protocol[2] などが知られている。当コンテナは秘密分散実行部から出力されたシェアを入力として受け取り保存する。計算を行う際は Docker Compose を利用してコンテナ間通信を実現する。計算された結果を再度秘密分散復元処理のコンテナに入力として与えるところまでを担う。

（３）その他

本システムでは全ての処理を AWS 上にあるコンテナ間にて行う。その際の通信に関しては Docker のネットワークを構築し、その中でシェアの共有や計算を行う。マルチパーティ計算実行部内の複数コンテナの管理などは Kubernetes によって行う。また AWS 上のセキュリティ関連リソースを用いることによって、Web API への不正なアクセスや攻撃などを防ぐことができ、セキュアなシステムを実現できる。

Development of multi-party computing system based on secret sharing using container orchestration technology

†Tatsuhiro HASHIMOTO, Jaewook KIM ・Shoin university

## ５．本システム各部の課題

（１）秘密分散実行部

本システムで用いる Shamir の$(k,n)$しきい値秘密分散法以外の主な秘密分散法は、「加法型秘密分散法」「複製型秘密分散法」が存在する[3]が、これらは計算量や扱うデータの大きさを考慮しながら用いることによって本来の効果があり、これらの実装も行っていきたい。また秘密分散処理専用コンテナについて、秘密情報を扱うケースによってはオンプレミス（AWS のリソースに秘密情報を置かない状態）との連携で利用することが現状最良であるが、システムの安定性やクライアントサイドでの問題が生じた際のバックアップ手段が必要という課題が挙げられる。

（２）マルチパーティ計算実行部

マルチパーティ計算の具体的な計算手法についてYaoによる方法を紹介したが、他の手法も検討していく必要がある。また、本システムでは統計処理などの高度な演算を行えるようにすることなども課題として挙げられる。

（３）その他

AWS 上のリソースについて障害が発生したときには、別のプラットフォーム（Google Cloud Platform など）を使用できるようにしておく必要があるが、現時点ではマルチパーティ計算への参加者が AWS 上のコンテナ同士であることを想定しているため、別のプラットフォームとの連携についての検討が必要である。また、コンテナやネットワークに関する脆弱性が原因で秘密情報が流出することなどもリスクとして存在し、セキュアな処理を保つためにも対策を考える必要がある。

## ６．まとめ

本稿では、利用者が比較的簡単に導入・使用することのできるコンテナオーケストレション技術を用いた秘密分散ベースのマルチパーティ計算システムの開発について述べた。

今後の課題としては、本システムの実行性能に関する評価や計算処理や通信などの部分での安全性についての考察とし、秘密分散およびマルチパーティ計算にて用いるアルゴリズムに関しては計算量と安全性のバランスを考えつつ、他の方式の利用も検討していきたい。また、統計処理や認証といったマルチパーティ計算の応用を行う機能の提供や、その他の構成についても、よりセキュアな技術を用いていきたいと考えている。

## 参考文献

[1] A.Shamir: How to Share a Secret, Communications of the ACM,Vol.22,No.11, pp.612-613,Nov.1979.
[2] A.C.Yao: How to Generate and Exchange Secrets, Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science, pp.162-167,1986.
[3] 大原一真: 秘密分散法を用いた秘密計算, システム制御情報学会誌「システム/制御/情報」プライバシー保護データマイニング特集号, Vol. 63, No. 2, pp. 71-76, 2019.

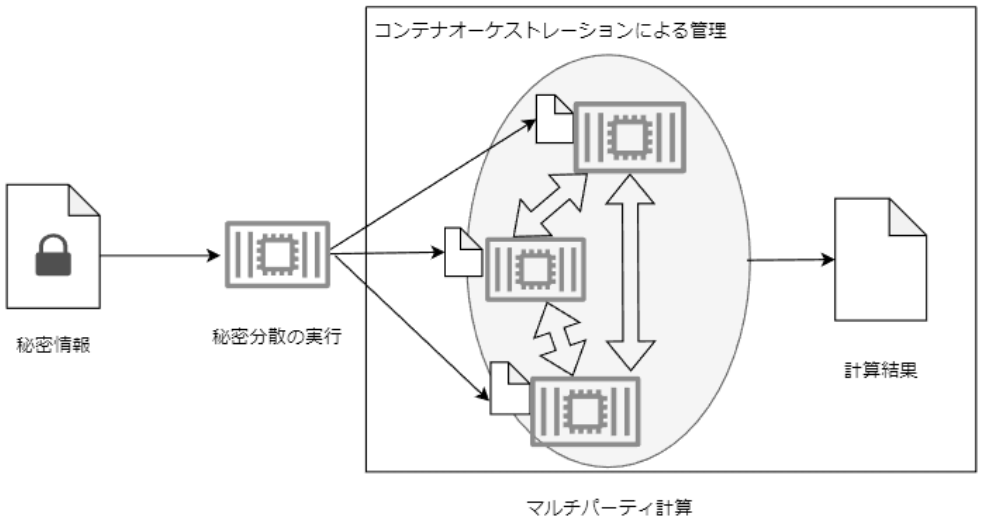


**図１　提案するシステムの概念図**