生命情報解析向けインタプリタを搭載した秘密計算用クラウド
- BI-SGX: Analyzing, only better. -

1　背景

近年，生命情報解析の分野におけるクラウド利用が活発となってきている．例えば，各種医療機関がアップロードしたデータを用いての計算や，複数の研究者がデータを持ち寄って機械学習等を行うマルチパーティ計算等，その応用方法は多岐に渡る．

こうした生命情報を用いてのクラウド計算を行う際には，データの取り扱いについて細心の注意を払う必要がある．何故ならば，ゲノムデータを始めとした生命情報は，そのデータから個人の様々な情報を特定できてしまう非常にセンシティブな情報である為だ．しかしながら，クラウドサービスというものは必ずしも安全であるとは限らず，寧ろリスクに満ち溢れている．悪性のクラウドプロバイダによる盗聴や外部からのクラウドマシンの脆弱性を突いた攻撃など，クラウドシステムの考えうるリスクの種類は枚挙に暇がない．

この為，生命情報のような機密性の高い情報を用いてクラウド上で演算を行う際には，安全である事が客観的に保証できる手段を用いて処理を行う必要がある．これを実現する為の，センシティブデータを保護しながら計算を行い，何らかの知見を得るような技術を「秘密計算」と呼ぶ．

数ある秘密計算技術の中でも，より現実的なパフォーマンスでの秘密計算を実現する技術として注目されているのが信頼可能な実行環境（Trusted Execution Environment）と呼ばれる技術である．TEE とは，ハードウェア等のセキュリティ機能の支援によって厳重に保護されている実行環境の事を指す．この TEE の内，Intel 社によって開発された技術に Intel Software Guard Extension（SGX）が存在する．SGX では，対応する Intel 製の CPU が搭載している専用の拡張機能の支援により，RAM 上に暗号的に厳重に保護された小区画を生成する事で，TEE の実現に成功している．準同型暗号などとは異なり，高速な AES 暗号をベースとした技術である為，SGX は現実的な実行コストにて秘密計算と同様の仕組みを実現する事が可能となる．

しかし，SGX にも SGX 独特の様々な欠点が存在する．SGX の欠点は，いずれも SGX を用いたプログラムの開発者に著しい負担を強いるような制限や仕様に由来するものに溢れており，これは生命情報解析を行う研究者にとって，生命情報解析の本筋とは関係のない些末な部分に大変な労力を割かせる結果となってしまう．この欠点により，折角実用的なパフォーマンスを実現可能である技術が存在するにも関わらず，生命情報解析を行う研究者から敬遠されてしまう原因になりかねないのが実情である．

2　目的

本プロジェクトの目的は，SGX の実用的なパフォーマンスを活用しつつも，研究者の負担を最小限に留めながら秘密計算を実行可能なクラウドプラットフォームを実現する事である．また，生命情報を保有するデータ所有者が安心して利用する事

の出来るセキュアクラウドストレージとしての機能も提供し，かつそのデータを安全に用いて秘密計算をする機能を実現する．その他にも，プロトコル上は正しいが機密情報を抜き出せてしまうような実行定義を阻止し，出力情報が匿名化されている事を担保する「アウトプットプライバシの保護」や，SGX が不得手とするサイドチャネル攻撃への対抗手段についても講じている．

3　開発の内容

本プロジェクトでは，Intel SGX が RAM 上に生成する保護領域「Enclave」上でインタプリタを駆動させる事で，SGX 公式 SDK の極めて不可解かつ煩雑な仕様に悩まされる事なく解析内容を記述し解析できる秘密計算用クラウドシステム「BI-SGX」(Bioinformatic Interpreter on SGX-based Secure Computing Cloud）を開発した．BI-SGX のシステム全体の概要図を図 1 に示す．

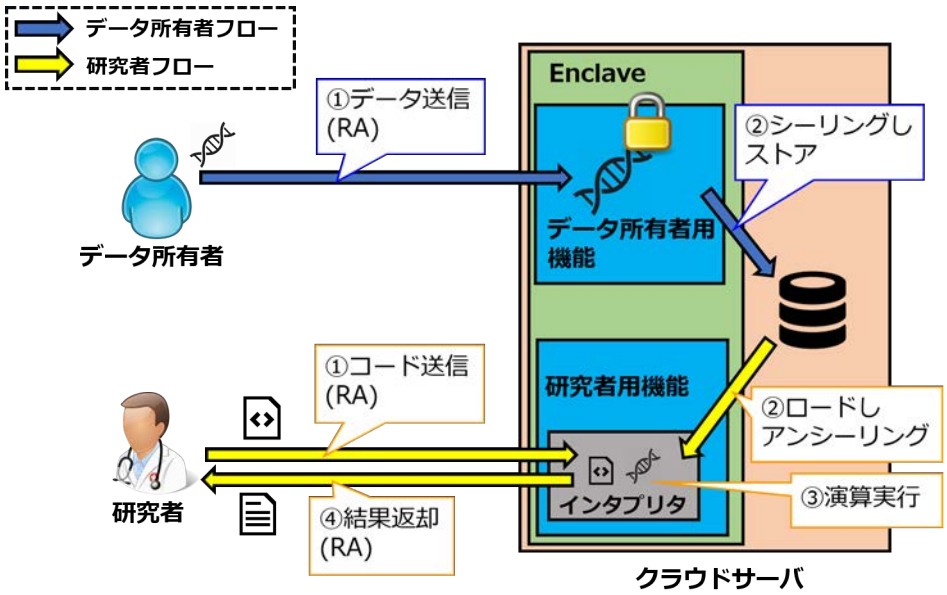

図 1: BI-SGX のシステム全体の概要図

BI-SGX は，ゲノムデータ等の機密情報を持つデータ所有者向けのストレージ機能と，そのストレージ上のデータを用いて様々な解析を行う事の出来る秘密計算機能を提供している．データ所有者のアップロードしたデータは，Enclave 内でのみ復号され，かつ SGX のシーリングと呼ばれる 128bit AES/GCM ベースの強固な暗号化によって保護された状態でストアされる．

一方で，研究者は BI-SGX が提供する独自の言語「Qliphoth」を用いる事で，簡単にクラウド上で実行したい処理の定義を記述する事が出来る．Qliphoth によるアリル頻度分析の秘密計算を要求するコードをプログラム 1 に示す．

プログラム 1: Qliphoth の組み込み関数を利用したプログラム例

```
1  func main()
2  var a
3          a = alleleFreq("21", 10417440, "JPT", "none")
4  end
```

Qliphoth を使用する事により，研究者は SGXSDK を用いた場合に比べて劇的に小さい負担で実行定義を記述可能になる．SGXSDK で開発を行った BI-SGX は，本体のコード量が合計約 40000 行という膨大なコード量にまで膨れ上がっている．しかしながら，Qliphoth を利用すれば，このように莫大な負担を強いられる SGXSDK を使わずに済み，たった 4 行で後述の GWAS のような比較的大規模な処理についての秘密計算を実行させる事が出来る．

また，BI-SGX は Qliphoth はプログラミング言語として基本的な機能を一通り備えているだけでなく，言語仕様レベルでアウトプットプライバシを侵害するような処理を根本的に排除している．アウトプットプライバシを侵害する処理とは，例えば平均を計算すると謳いながら，単一のデータに対する平均値，即ちデータそのものを取り出す，というような,「プロトコル上は正しいが結果的に得られる値がプライバシを侵害している」ような処理を指す．Qliphoth では，SQL 等とは異なり，データの集合であるデータセットよりも細かい粒度で条件指定を行う事が根本的に不可能である為，アウトプットプライバシを侵害するような処理を排除する事に成功している．

更に，BI-SGX は SGX アプリケーションが概して苦手とするサイドチャネル攻撃に対する防護も実現している．サイドチャネル攻撃の中でも極めて SGX に対して有効な攻撃に，制御チャネル攻撃（Controlled-Channel Attack）が存在する．これは，ページフォルトを悪用して条件分岐先の変数や関数へのアクセスを観測し，その結果から分岐条件となった変数の値を推定するような攻撃である．しかし，BI-SGX では Qliphoth のスクリプトコードは Enclave により完全に保護されており，かつインタプリタ本体は字句解析器がスクリプトコードから取得する「トークン」という，謂わばその要素の種別に相当する粒度でしか条件分岐を行わない．よって，制御チャネル攻撃をもってしても秘密情報を BI-SGX から抽出する事が不可能となっている．

4 従来の技術との相違

BI-SGX は，SGX の提供する高い安全性と計算コストの低さを利用するにあたって，本来使用しなければならない非常に煩雑な SGXSDK を使わずに済むという点が最も大きい．例えば，BI-SGX の本体は合計 40000 行にも及ぶ,（少なくとも個人が単独で開発するシステムとしては）大規模なシステムとなっている．しかし，利用者はこのような膨大な記述を行う必要はない．BI-SGX を実現するにあたり，本プロジェクトにおいて独自に開発した Qliphoth 言語を用いれば，例え GWAS 処理であろうとたった 1 行で解析の実行と結果の表示が出来てしまう．単純比較をすると，それこそ数万倍の負担削減を実現できるのである．BI-SGX は様々な処理を提供するシステムであるから，単純比較は野暮であるかも知れないが，それでも劇的に負担を

削減できている事に相違はない．

ここで，BI-SGX と先行研究である PRINCESS，Graphene-SGX，SGXElide，そして SGX-BigMatrix との比較を行う．これらの比較表を表 1 に示す．

	PRINCESS	Graphene-SGX	SGXElide	SGX-BigMatrix	BI-SGX
アウトプットプライバシ	✓	*	×	✓	✓
低利用難易度	△	✓	×	✓	✓
コードの保護	×	×	✓	×	✓
実行定義変更の柔軟性	×	*	✓	✓	✓
EPC 消費量	✓	△	✓	✓	✓

表 1: 先行研究と BI-SGX の比較

但し「*」と記された要素は，同一システム内で両立する事が不可能である事を示す．先行研究のシステムがどれも何らかの欠点を抱えているのに対して，BI-SGX は，いずれの項目に対しても有効なソリューションを提供する事が出来ている．アウトプットプライバシに関しては，Qliphoth 言語の言語仕様レベルで制御している為，厳密に保護する事が出来る．また，Qliphoth 言語は簡潔な文法を提供している為，開発者にとっての実装に際する負担も大幅に軽減されている．コードの保護及び実行定義の変更における柔軟性に関しては，スクリプトコードをデータとして送信し，Enclave 内で復号・展開する事によって実現できている．更に，インタプリタ本体について，いくつか存在するインタプリタの実装方法の中でも特にコンパクトな構成方式を採用している為，EPC 消費量も限りなく最小化する事に成功している．

次に，秘密計算の既存手法である完全準同型暗号との比較した場合の BI-SGX の有用性についても示す．Qliphoth では，データ所有者がアップロードしたデータを用いて編集距離を秘密計算する組み込み関数を提供している．この関数では，64 文字同士の 50000 ペアについて実行してもわずか 63 ミリ秒で処理を完了する事が可能である．一方で， Cheon et al., 2015 の論文「Homomorphic Encryption of Edit Distance」が示す実験結果では，8 文字同士の文字列の編集距離を計算する為に，並列化無しで 5 時間 13 分という極めて重い実行時間を記録している．BI-SGX における編集距離計算用組み込み関数の実験と Cheon et al., 2015 の論文における実験結果の比較を図 2 に示す．

	セキュリティレベル	**データ数**	**実行時間(並列化なし)**
完全準同型暗号[3]	80bit	8文字×8文字	**5時間13分** (1.88×10^7ミリ秒)
提案手法	128bit	64文字×64文字×50000	**63.572ミリ秒**

図 2: BI-SGX と完全準同型暗号の編集距離計算における比較表

BI-SGX は，完全準同型暗号に比べてセキュリティレベルで大きく勝っており，かつ編集距離を計算する文字列の合計サイズも 25600 倍と遥かに大きいにも関わらず，実行時間では完全準同型暗号に比べて約 3 億倍の短さという圧倒的に高い実行効率を実現している．

以上の事から，BI-SGX が秘密計算を行う既存手法と比較して実用的な秘密計算環境を提供できていると考えられる．

5　期待される効果

生命情報解析の分野においては，扱うデータの機密性が極めて高い事，そして十分に実用的な秘密計算技術が存在しなかった事により，各組織が各自で出たデータを内輪のみで扱うしか無かった．つまり，各組織が得たデータを，外部から隔絶された謂わば ”ファラデー箱 ” 的な空間で厳重に取り扱う事によって，生体情報を守りながら解析を行うのである．しかし，BI-SGX という実用的な秘密計算用クラウドシステムが実現した事により，そのようなデータの囲い込みを打破し，世界中の研究機関が各自のデータを持ち寄りグローバルスケールの知見を得る事が出来る，生命情報解析の分野におけるブレイクスルーとなる事が出来る．

6　普及の見通し

BI-SGX は，Intel SGX 対応のマシンを使用し，Linux OS をインストールして，SGX ドライバや SGXSDK 等の SGX 環境をインストールすれば，簡単に導入及び使用が可能になる．これはパブリッククラウドについても例外ではなく，例えば Microsoft Azure や IBM Cloud 等で BI-SGX を駆動させる事が可能である．よって，まずは生命情報解析の分野の研究者や生体情報を有する機関を始めとして普及していき，パブリッククラウド上で BI-SGX を駆動させる事でより大規模な秘密計算クラスタを形成する事も可能になる．また更には，BI-SGX は生命情報解析以外の分野にも十分に応用可能であるから，ゆくゆくは SGXSDK の煩雑な仕様に耐えねばならない現状を覆し，BI-SGX のフレームワークが SGX システムにおけるデファクトスタンダードになるポテンシャルすら秘めていると考えられる．

7　クリエータ名（所属）

櫻井　碧（早稲田大学大学院基幹理工学研究科情報理工・情報通信専攻）

(参考) 関連 URL

- BI-SGX の GitHub リポジトリ：`https://github.com/hello31337/BI-SGX`