TEE を用いたセキュアかつ高性能なデータベースシステムの開発
- CASSA: 機密性と性能を両立するデータ基盤 -

1　背景

クラウドの利用は増加の一途を辿っている。Google、Microsoft、Amazon など、様々な企業がインターネットを介して計算資源を貸し出すクラウドサービスを提供している。それらは外部からの攻撃に対する様々な策を講じているため、安全であると考えられている。しかし、それにはクラウドサービスを提供するクラウド事業者が信頼できるという前提がある。クラウドでは、外部からの不正アクセスに対する防御措置に加えて、クラウド事業者自身が不正アクセスを行わないかを考慮する必要がある。なぜなら、クラウド事業者は計算資源の管理者権限を有しているため、メモリ上のデータを読み取ることができるからである。

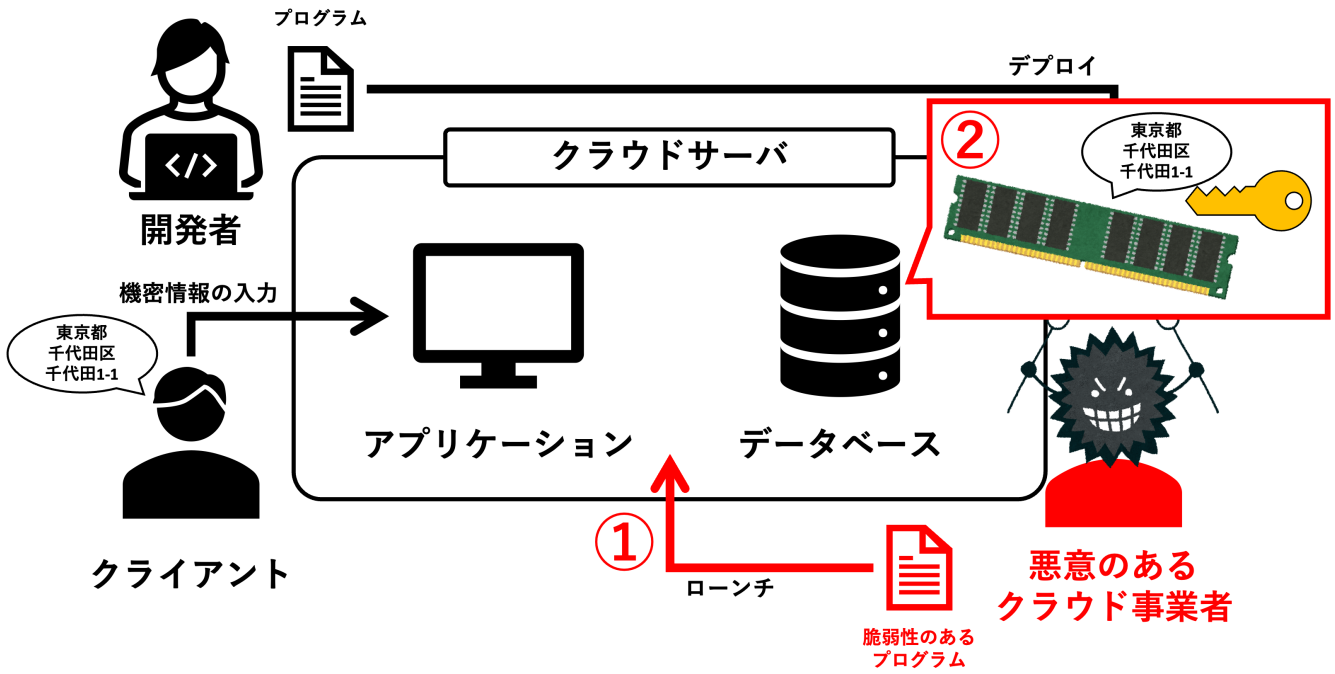

図 1: クラウド事業者に悪意がある場合

クラウド事業者により可能な攻撃手法を、図 1 に示す。クラウド事業者に悪意がある場合、主に 2 つのシナリオが考えらえる。

1 つ目のシナリオは、クラウド事業者が開発者から受け取ったプログラムを意図的に改ざんすることである。図 1 の①は、開発者が提供したプログラムを、クラウド事業者が保有する計算資源上で実行することで、サービスを公開している。この過程でクラウド事業者は、開発者が提供したプログラムを意図的に改ざんしたり、既知の脆弱性を修正するためのパッチを適用しないなどの不正行為を行う可能性がある。この改ざんにより、システム内にセキュリティホールを作成し、それを悪用して情報を奪取したり、漏えいさせる可能性がある。クラウド事業者が、脆弱性のパッチを適用せずに古い状態を保持することでも、同様のリスクが生じる。

2 つ目のシナリオは、管理者権限の悪用である。図 1 の②では、クラウド事業者が有する計算資源上でアプリケーションを実行し、クライアントにサービスを公開している。クライアントはサービスと通信を行い、必要に応じて情報を提供する。アプリケーションの実行中は、プログラムとデータがメモリ上に展開され、CPU が適宜データを処理する。クライアントが提供したデータもこのプロセスに含まれる。データは通常、転送や保存時に暗号化されているため、機密性は保証されている。しかし、データが処理されている際には状況が異なる。一般的な CPU はデータを暗号化したまま演算できないため、データを処理するときは一時的に復号し

たデータをメモリ上に配置し、CPU で演算を行った後、その結果をメモリに書き戻し、再度暗号化してストレージに保存するというプロセスが必要である。このプロセス中、メモリ上には復号されたデータが存在するため、クラウド事業者は管理者権限を悪用することで、メモリ上のデータを読み取り、機密データを盗み出すことが可能である。クラウドでは上記の理由から、そのような機密情報の機密性は保証されていない。

2　目的

　クラウドの恩恵を受けるためには、クラウド事業者に依存しない形で機密性の保証が必要である。これは、ソフトウェアレベルの保護だけでは不十分である。データは処理する際、復号されなければならないため、管理者権限を悪用しメモリ上のデータを読み取ることが可能なクラウド事業者に対しては、ソフトウェアレベルの保護は効力を発揮しない。本プロジェクトでは、クラウド事業者が信頼できない条件下でもクラウド上の機密性を保証できるシステムを構築する。

3　開発の内容

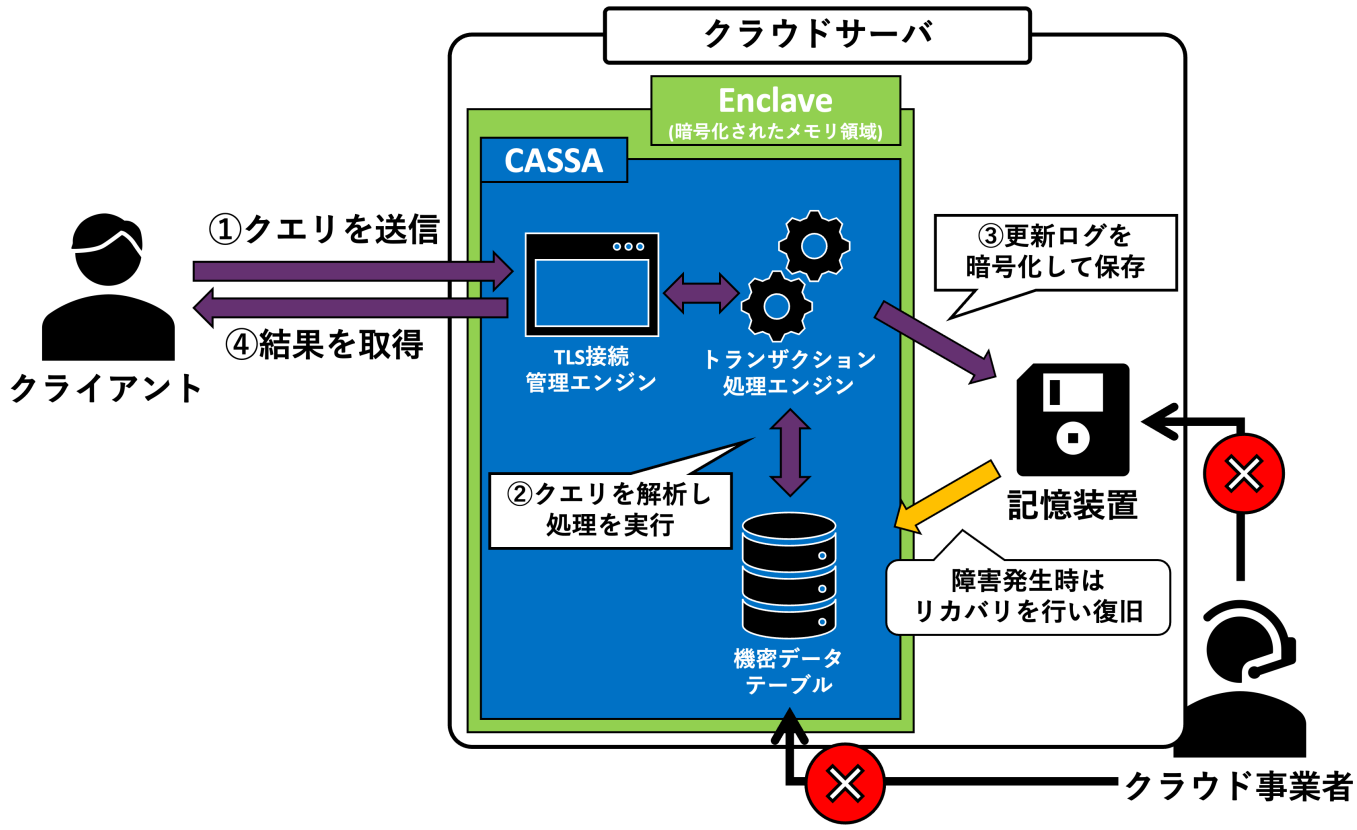

図 2: CASSA アーキテクチャ

　本プロジェクトで開発した CASSA（Cloud-Adapted Secure Silo Architecture）は、クラウド事業者が信頼できない状況下でも、機密性と性能を両立するデータ基盤である。CASSA のアーキテクチャを図 2 に示す。

　このシステムは、Intel SGX の 1 つである「Scalable-SGX」を用いて隔離実行環境（Enclave）のメモリ制限を緩和し、トランザクション処理プロトコルには Silo を、索引には Masstree を採用している。これにより、データの処理性能と機密性を向上させる。さらに、Intel SGX の機能である「Remote Attestation」を活用し、意図した通信相手が Enclave 内で正しいプログラムを実行していることを保証しつつ、TLS セッションを確立することで、通信相手と通信経路の機密性と真正性を保証する。システムの耐故障性付与のためのログに関しても、暗号化して保存することで、クラウド事業者による不正な観測や操作を防ぐ。信頼できないストレージ

上のログデータは欠損するリスクがあり、完全性保証のため、CASSA はログ改ざん検知プロトコルを導入している。これにより、ログの欠損や改ざんを検知可能にし、システムの完全性と真正性を保証する。

クライアントは、TLS セッションを通じて、CASSA にクエリを送信する（図 2 の①）。CASSA はクエリを解析し、Silo と Masstree を用いてデータの検索、更新を行う（図 2 の②）。クエリによるデータベースの改変内容は、必要に応じてログを作成し、暗号化してストレージに書き込みを行う（図 2 の③）。ログ書き込み後、クライアントに結果を返送する（図 2 の④）。全体として常にデータの機密性が保証されているのが本システムの特徴である。CASSA のコンポーネントは以下の通りである。

- Silo 並行性制御システム

  Silo は、メニーコアと大容量メモリを活用できるように設計されたトランザクション処理システムである。CASSA における、トランザクション処理を担当する。
- Masstree

  Masstree は、B+ 木の順序維持性、トライ木のプレフィックスマッチングによる迅速な検索を組み合わせた並行索引木である。CASSA における、高速データアクセスを担当する。
- CASSA リカバリシステム

  リカバリシステムは、システム障害や予期せぬシャットダウンからの迅速な復旧を可能にする機能である。CASSA における、データの完全性と一貫性の保証を担当する。
- ECDSA Attestation

  ECDSA Attestation は、Intel SGX が提供する Remote Attestation 方式の 1 つで、Intel SGX が有効なプラットフォームで、適切にインスタンス化された Enclave 内で、既知のセキュリティ構成のシステム上で実行されていることを遠隔から検証する機能である。CASSA における、通信相手の真正性検証とセキュアな TLS セッション確立を担当する。
- クエリパーサ

  クエリパーサはクライアントからのクエリ内容を解析し、トランザクション処理システムが扱える形式へ変換する機構である。CASSA における、データ互換性の付与を担当する。
- TLS 接続管理エンジン

  TLS 接続管理エンジンは、複数のクライアントが同時にアクセスする際に確立された TLS セッションを効率的に管理する機能である。CASSA におけるセキュアな通信の維持と管理を担当する。

4 従来の技術との相違

- EnclaveDB

  EnclaveDB は、データとクエリの機密性と完全性を保証するデータベース管理システムである。SQL Server を拡張し、Enclave 内での機密データの処理及び管理を行うことで、OS やハイパーバイザからの不正アクセスや改ざんを防ぐ。EnclaveDB は逐次ロギングを採用しており、並列ロギングを採用していない点が CASSA と異なる。
- ShieldStore

  ShieldStore は、Intel SGX を活用し、インメモリキーバリューストアを保護するために設計されたシステムである。SheildStore は旧式 SGX に基づくためメモリ制限が 128MB である一方、CASSA のメモリ制限は CPU1 基当たり最大 512GB である。
- Always Encrypted

  Always Encrypted は、SQL Server データベースに保存された機密情報の保護を目的とした機能である。この機能は、データに対する操作を Enclave 内で行うことにより機密性を保証するが、取得したデータを用いた分析などを行うことは考慮されていない。

・MONOMI

MONOMI は暗号化されたデータを復号化せずに分析可能なシステムである。MONOMI は準同型暗号を用いる一方、CASSA は Enclave を用いる点が異なる。

・AMD SEV、Intel TDX

AMD SEV と Intel TDX は、仮想マシン全体を暗号化して保護する TEE 技術である。これにより、仮想化された環境下でのデータと実行コードの機密性は保証されるが、OS を含む VM 全体が保護の対象となるため、OS を信頼する必要がある。CASSA は Intel SGX を用いるため、OS を信頼する必要はない。

・ARM TrustZone

ARM TrustZone は、特定のメモリ領域を隔離し、保護することに特化した TEE 技術である。TrustZone はメモリ隔離に注力しており、隔離実行環境のメモリ暗号化をしない。CASSA は、Intel SGX を活用することにより、隔離実行環境のメモリ暗号化を実現する。

5　期待される効果

CASSA は、Intel SGX の不親切な仕様や煩雑な部分を抽象化している。CASSA を使えばそのような困難に直面することなく、データの機密性と真正性が保証されたデータ基盤を構築可能である。また、CASSA は高い処理能力を実現し、クラウド上での高速なデータ処理を可能にする。これにより、仮にクラウド環境を十全に信用できない状況であっても、機密データの安全な管理と高速なデータ処理を行うことができるため、機密データを扱う任意の分野での応用が期待される。

6　普及の見通し

このデータ基盤は、機密性と性能を両立するため、金融、医療等、機密データを扱う任意の分野に応用可能である。例えば、医療データの分析では、準同型暗号を用いた秘密計算が実施されているが、計算オーバーヘッドによる性能低下が懸念されている。CASSA が提供するセキュアなデータ管理環境は、現実的な時間内での複雑なデータ処理を実現する。

7　クリエータ名（所属）

- 福山 将英（慶應義塾大学 環境情報学部 環境情報学科）

（参考）関連 URL

- ソースコード：https://github.com/Noxy3301/CASSA