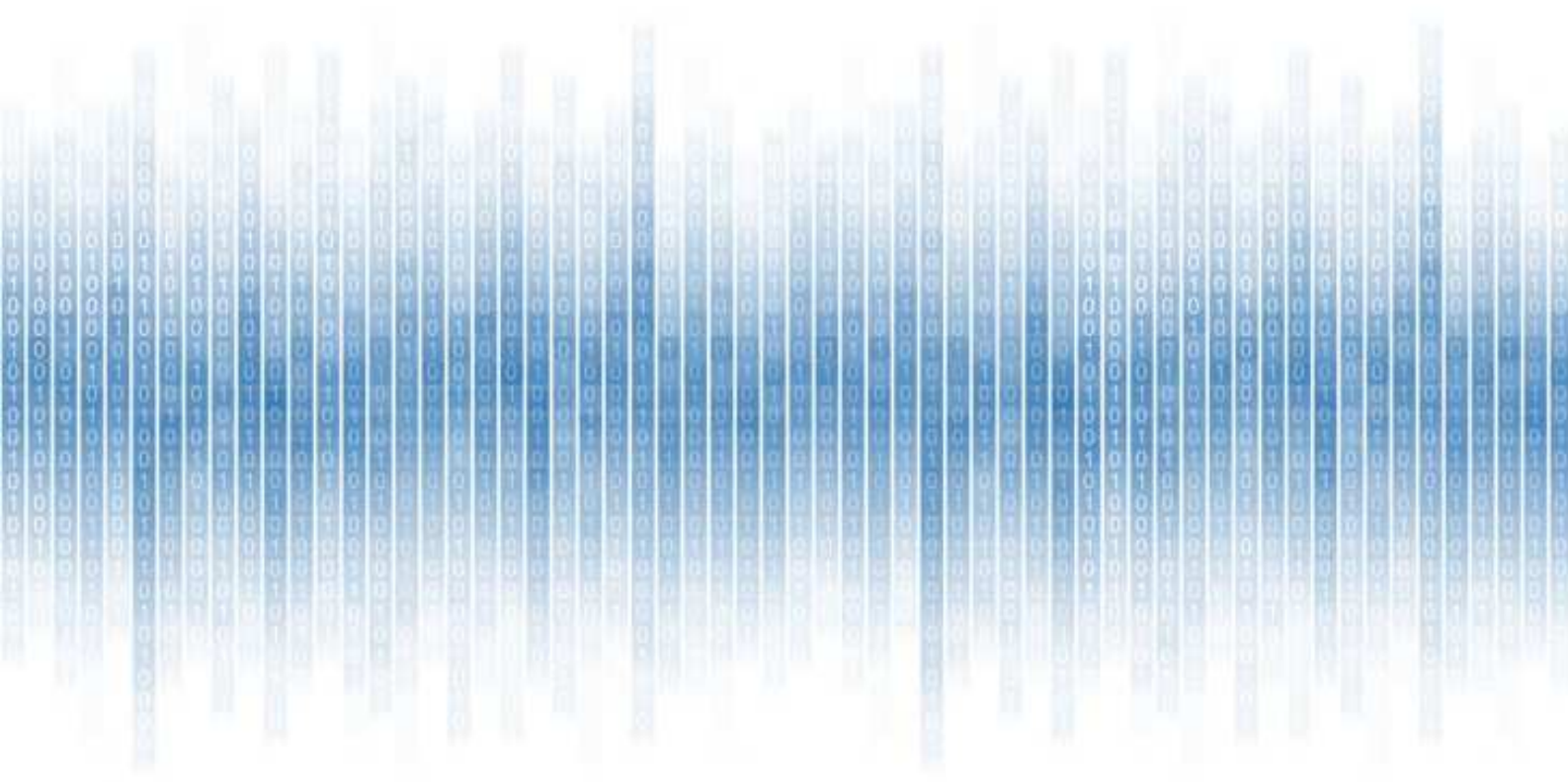


クラウドセキュリティアライアンス

ビッグデータワーキンググループ

ビッグデータのセキュリティ／プライバシーにおける十大脅威 拡張版（日本語訳）

2014 年 5 月

Acknowledgements

CSA Big Data Working Group Co-Chairs

Lead: Sreeranga Rajan, Fujitsu
Co-Chair: Wilco van Ginkel, Verizon
Co-Chair: Neel Sundaresan, eBay

Contributors

Anant Bardhan, CTS
Yu Chen, SUNY Binghamton
Adam Fuchs, Sqrrl
Aditya Kapre
Adrian Lane, Securosis
Rongxing Lu, University of Waterloo
Pratyusa Manadhata, HP Labs
Jesus Molina, Fujitsu
Alvaro A. Cárdenas Mora, University of Texas Dallas
Praveen Murthy, Fujitsu
Arnab Roy, Fujitsu
Shiju Sathyadevan, Amrita University
Nrupak Shah, Dimension Data

CSA Global Staff

Alex Ginsburg, Copyeditor
Luciano JR Santos, Global Research Director
Evan Scoboria, Webmaster
Kendall Scoboria, Graphic Designer
John Yeoh, Research Analyst

日本語訳の提供について

「ビッグデータのセキュリティ／プライバシーにおける十大脅威 拡張版」は、Cloud Security Alliance Big Data Working Group よりリリースされている「Expanded Top Ten Big Data Security and Privacy Challenges」（2013 年 4 月）の日本語訳です。このドキュメントは、ビッグデータセキュリティに関心のあるクラウドユーザーの教育・啓発を目的として、原文をそのまま翻訳したものであり、日本独自の法令や基準に関する記述は含まれておりません。

なお、日本クラウドセキュリティアライアンスに関する情報は、以下の URL より参照可能ですので、ご覧下さい。

http://www.cloudsecurityalliance.jp/

このドキュメントは、以下の日本クラウドセキュリティアライアンスの有志により作成されています。

日本クラウドセキュリティアライアンス・ビッグデータユーザーワーキンググループ

リーダー：笹原 英司 （特定非営利活動法人ヘルスケアクラウド研究会　医薬学博士）

阿倍 克英 （特定非営利活動法人ヘルスケアクラウド研究会）

里中 慧 （特定非営利活動法人ヘルスケアクラウド研究会）

協力

イー・ガーディアン株式会社

目次

はじめに

ビッグデータという用語は、企業や政府機関が、人間や我々の周辺から収集している大容量のデジタル情報を指している。生成されるデータの容量は、2 年毎に 2 倍となり、2012 年には 250 京バイトで 2020 年には 4,000 京バイトとなる見込みである[56]。セキュリティ／プライバシーの問題は、ビッグデータの容量、多様性、速度によって増幅される。大規模なクラウドインフラストラクチャ、データソースやフォーマットの多様性、データ収集の流動的な性質、大容量のクラウド間の移動が全て、固有のセキュリティの脆弱性を作り出す。

単に、新たなセキュリティの脅威を生み出す大容量のデータが存在するだけではない。ビッグデータは、数十年に渡り、多くの組織によって収集・活用されてきた。今やあらゆる規模の組織がビッグデータにアクセスし、導入する手段を有しているので、現在のビッグデータ利用は新たなものになっている。従来、ビッグデータは、政府機関、大企業など、大容量のデータをホスティングしてマイニングするために必要なインフラストラクチャを構築・保有する余裕のある、非常に大規模な組織に限られていた。これらのインフラストラクチャは、概して独自開発で、一般的なネットワークから孤立していた。今日、ビッグデータは、パブリッククラウドのインフラストラクチャを介して、大規模および小規模の組織に、安価かつ容易にアクセスすることが可能である。Hadoop のようなソフトウェアインフラストラクチャによって、開発者は、数千のコンピューティングノードを活用してデータの並行計算処理を実行することができる。パブリッククラウドプロバイダーから必要に応じて計算処理能力を購入する機能と結びつけることによって、このような開発が、ビッグデータマイニング手法の採用を大いに加速させている。結果として、データを保存し、計算処理するために、コモディティ化したハードウェアと、コモディティ化した OS やコモディティ化したソフトウェアインフラストラクチャの異種の構成によって特徴付けられるパブリッククラウド環境下で、ビッグデータの結合から新たなセキュリティの脅威が生まれてきた。

ビッグデータが、ストリーミングのクラウド技術を介して拡大するにつれて、ファイアウォールに囲まれて半ば孤立したネットワーク上で、小規模の静的データをセキュアにするために構築された伝統的なセキュリティのメカニズムでは不十分である。例えば、アノマリ検知の分析では、大量の異常値を生成するかもしれない。同様に、既存のクラウドインフラストラクチャで来歴を追加する方法は明らかでない。ストリーミングデータは、セキュリティ／プライバシーソリューションから超高速のレスポンス時間を必要とする。

本稿は、実務家に従って、ビッグデータにおけるセキュリティ／プライバシーの十大脅威に焦点を当てることを目的としている。そのために、本ワーキンググループは、ビッグデータにおける重要な脅威に到達するために、3 段階のプロセスを活用した。

1. 本ワーキンググループは、重大なセキュリティ / プライバシーの問題についてのドラフトの最初のリストを作成するために、クラウドセキュリティアライアンス（CSA）の会員にインタビューを実施し、セキュリティ専門家向けの業界誌にサーベイを行った。

2. 本ワーキンググループは、公開されている研究資料の調査を行った。

3. 本ワーキンググループは、提案されたソリューションが問題のシナリオをカバーしない場合に、その問題を脅威として特徴付けた。

この 3 段階のプロセスに基づいて、本ワーキンググループは、ビッグデータのセキュリティ／プライバシーに対する十大脅威をとりまとめた：

1. 分散プログラミングフレームワークにおけるセキュアな計算処理

2. ノンリレーショナルデータデータストアに対するセキュリティのベストプラクティス

3. セキュアなデータ保存とトランザクションのログ

4. エンドポイントの入力の検証／フィルタリング

5. リアルタイムのセキュリティ／コンプライアンスモニタリング

6. 拡張性があり構成可能なプライバシー保護データマイニング／分析

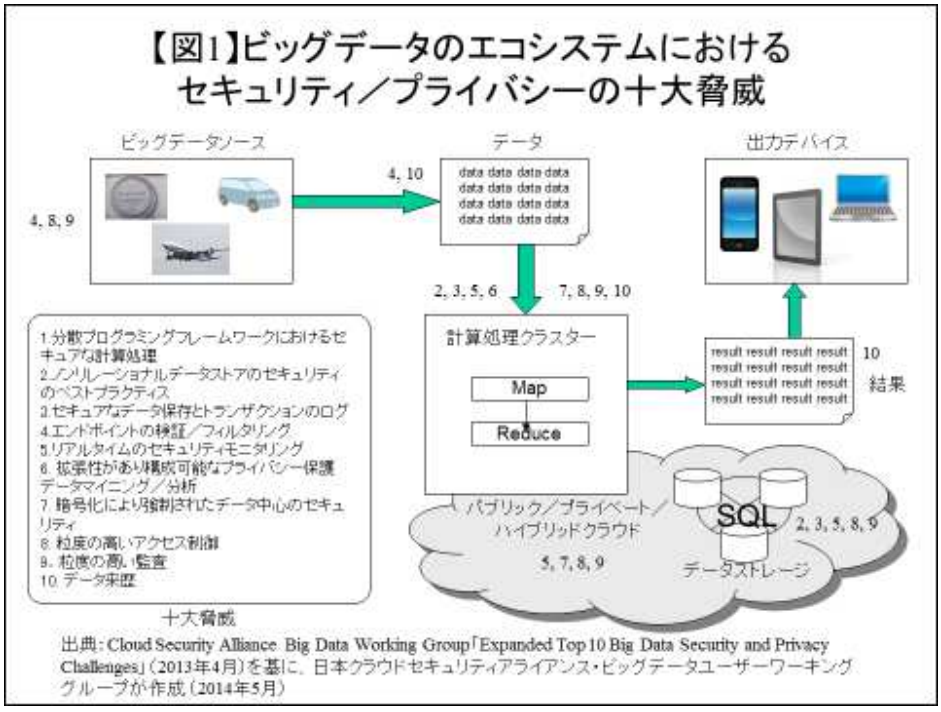
7. 暗号化により強制されたデータ中心のセキュリティ

8. 粒度の高いアクセス制御

9. 粒度の高い監査

10. データ来歴

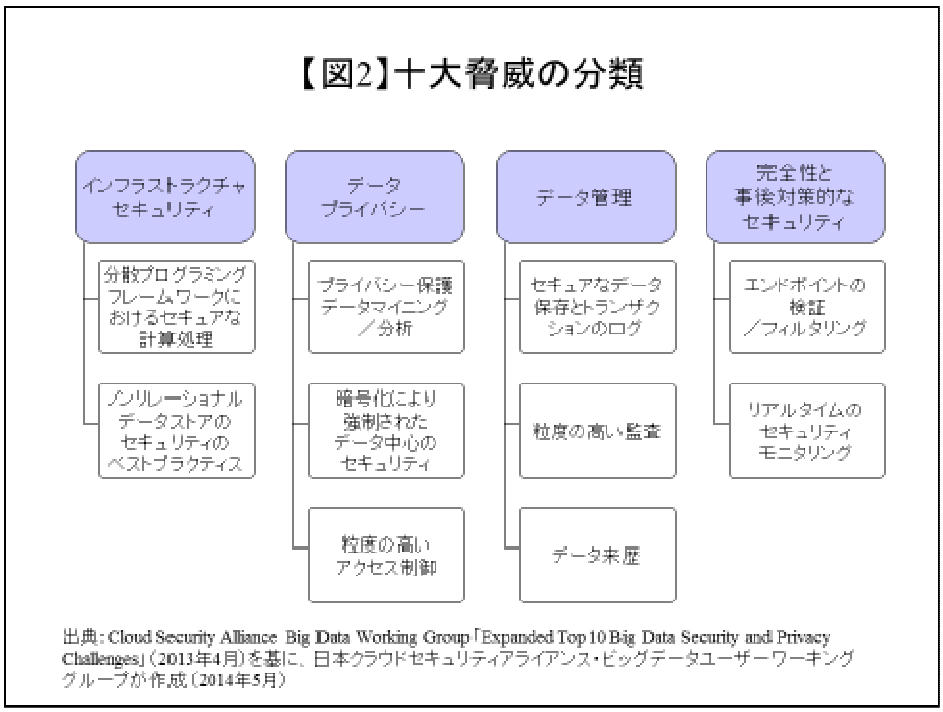
図 1 は、ビッグデータのエコシステムにおける十大脅威を示している。



出典：Cloud Security Alliance Big Data Working Group「Expanded Top 10 Big Data Security and Privacy Challenges」（2013年4月）を基に、日本クラウドセキュリティアライアンス・ビッグデータユーザーワーキンググループが作成（2014年5月）

図 2 に示す通り、脅威は、ビッグデータのエコシステムにおける 4 つの観点に整理することが可能である：

1. インフラストラクチャセキュリティ
2. データプライバシー
3. データ管理
4. 完全性と事後対策的なセキュリティ



ビッグデータシステムのインフラストラクチャをセキュアにするためには、分散化した計算処理とデータストアをセキュアにすることが必要である。データ自体をセキュアにするためには、情報の配布がプライバシーを保護するものである必要があり、機微なデータが、暗号化や粒度の高い制御の利用を介して保護される必要がある。膨大な容量のデータを管理するためには、拡張性があり、分散化されたソリューションによって、データストアをセキュアにすると共に、効果的な監査やデータ来歴を可能にする必要がある。最後に、多様なエンドポイントから出現したストリーミングデータについては、完全性をチェックし、リアルタイム分析を実行してインフラストラクチャの健全性を保証するために利用できるようにしておく必要がある。

セキュリティ／プライバシーの脅威を解決するためには、通常、3 つの明確な問題点を処理することが求められる：

1. モデリング：サイバー攻撃もしくはデータ漏えいのシナリオの大半をカバーする脅威モデルを構築する。
2. 分析：脅威モデルに基づいて、扱いやすいソリューションを見つける。
3. 導入：既存のインフラストラクチャにソリューションを導入する。

本稿では、個々の脅威を簡単に説明し、脆弱化する恐れのあるビッグデータの利用について吟味して、個々の脅威のモデリング、分析、導入に従い、既存の知識を要約する。

1.0 分散プログラミングフレームワークにおけるセキュアな計算処理

分散プログラミングフレームワークでは、大容量データを計算して保存するために並列処理を利用する。例えば MapReduce フレームワークは、入力ファイルを複数のチャンク（かたまり）に分割する。MapReduce の最初のフェーズでは、個々のチャンクの Mapper がデータを読み込み、一定の計算処理を行って、鍵／値のペアのリストを出力する。次のフェーズでは、Reducer が個々の鍵に附属する値を結びつけて、結果を出力する。主な攻撃防止手段としては、Mapper のセキュア化と、信頼できない Mapper に存在するデータのセキュア化の 2 種類がある。

1.1 ユースケース

信頼できない Mapper が変更され、要求に応じて覗き見したり、MapReduce スクリプトを変えたり、結果を変えたりすることが起こり得る。最も難しい問題は、不正確な結果を返す Mapper を検知することであり、代わりに不正確な集約結果を生成することになる。大規模なデータセットの場合、判別することは不可能に近く、特に科学／金融計算においては重大な損害を生む結果となる。

小売業者の消費者データは、ターゲット広告や顧客セグメンテーションのためにマーケティング代理店が分析することがよくある。これらの作業には、大規模のデータセット上での高度な並列処理が含まれており、特に Hadoop のような MapReduce フレームワークに適している。しかしながら、データの Mapper に、意図的若しくは意図的でない漏えいが含まれる可能性がある。例えば、Mapper が、プライベートな記録を分析し、特別な値を外に出して、ユーザーのプライバシーを侵害する可能性がある。

1.2 モデリング

Mapper の脅威モデルには、三つの主要なシナリオがある。

1. **Worker ノードの誤作動による計算処理** - 分散計算処理で Mapper に割り当てられた Worker が、不正確な構成や障害ノードにより誤作動を起こす可能性がある。誤作動を起こした Worker は Mapper から不正確な結果を返すことがあり、集約結果の完全性を損なう可能性がある。また、このような Worker が修正されて、ユーザーの機密データを漏えいしたり、ユーザーの行動やプライバシーマイニングの選択項目をプロファイリングしたりする可能性がある。

2. **インフラストラクチャ攻撃** – 危険にさらされた Worker ノードは、他の Worker と、再生を目的とする Master、中間者、MapReduce の計算処理に対する DoS 攻撃との間の通信を傍受する可能性がある。

3. **偽りのデータノード** – 偽りのデータノードがクラスタに追加されて、その後、複製されたデータを受信するか、変更された MapReduce コードを配布する可能性がある。正当なノードのスナップショットを生成し、変更されたコピーを再導入する能力は、クラウド／仮想化環境上の直接攻撃であり、検知することが困難である。

1.3 分析

上述の脅威モデルに基づいて、Mapper の信頼性の保証と、信頼できない Mapper におけるデータのセキュア化の 2 つの分析軸がある[1]。

Mapper の信頼性を保証するためには、信頼の設定と、強制アクセス制御（MAC）の 2 つの技法がある。

1. 信頼の設定には、内部的な信頼の設定と、それに続く時系列的な信頼のアップデートの 2 つの段階がある。Worker が Master に対して接続要求を送信すると、Master は Worker を認証する。期待されたプロパティを有する、認証された Worker のみが、Mapper のタスクに割り当てられる。内部的な認証に続いて、各 Worker のセキュリティのプロパティが、あらかじめ定義されたセキュリティポリシーに適合するかどうか時系列にチェックされる。

2. MAC が、あらかじめ定義されたセキュリティポリシーによって認証されたファイルへのアクセスを保証する。MAC は入力の Mapper に対する完全性を保証するが、Mapper 出力からのデータ漏えいを防止するものではない。

Mapper 出力からの情報漏えいを防止するためには、集約計算処理の出力を介したプライバシー違反を防止するデータ匿名化技術が要求される。データ匿名化の数学的に厳格な定義は、差動的なプライバシーの概念であり、ランダムなノイズを、計算処理の出力結果に追加することによって達成される。しかしながら、特定の技術がプライバシー保護に寄与することを証明するのは難しい。

1.4 導入

MAC は、Airavat [1]において、MapReduce フレームワークや分散ファイルシステム、基盤となる OS としての SELinux が付属する Java 仮想マシンを修正することによって導入される。SELinux において、MAC は、信頼できないコードがシステムリソースを介して情報を漏えいしないことを保証する。しかしながら、信頼できない Mapper によって生成される出力鍵に基づく計算処理のプライバシーを保証することはできない。出力を介した情報漏えいの防止は、最近開発された、関数の感度に基づく差動的なプライバシーの匿名化フレームワークに依存する。Mapper の文脈関係において、関数の感度は、入力が Mapper の出力に及ぼす影響の度合いのことである。任意の信頼できないコードの感度を推定することは困難である。

幅広く実践的に導入するために上述のソリューションに取り組む際には、2 つの問題がある：

1. MAC の負荷によるパフォーマンスの損失
2. 保証の提供における差動的なプライバシーの限界

2.0 ノンリレーショナルデータストアに対するセキュリティのベストプラクティス

NoSQL によって普及したノンリレーショナルデータストアは、セキュリティインフラストラクチャに関しては、まだ進化の途上にある [2]。例えば、NoSQL インジェクション向けの堅牢なソリューションは未成熟である。個々の NoSQL DB は、分析の世界から提示された異なる課題に取り組むよう構築されており、設計段階においてセキュリティが取り扱われることはなかった。NoSQL データベースを利用する開発者は、通常、ミドルウェアにセキュリティを組み込んできた。NoSQL データベースは、データベースの中で明確にそれを強制するためのサポートを提供していない。しかしながら、NoSQL データベースにおけるクラスタの観点は、このようなセキュリティプラクティスの堅牢性に対する追加的な課題を示している。

2.1 ユースケース

大規模な非構造化データセットを取り扱う企業は、伝統的なリレーショナルデータベース（RDB）を NoSQL データベースに移植することによって恩恵を得る可能性がある。NoSQL データベースは、予測分析や時系列分析のために、大容量の静的／動的データを収容して処理する。NoSQL データベースで広く利用されている脅威モデル化技術を利用した詳細な脅威分析から得られた脅威ツリーは、伝統的な RDB と比較して、NoSQL データベースが非常に薄いセキュリティのレイヤしか持っていないこ

とを示している。一般的に、NoSQL データベースのセキュリティの思想は、外部の強制メカニズムに依存している。セキュリティインシデントを減らすために、企業は、ミドルウェアのセキュリティポリシーを見直して、エンジンに項目を追加すると同時に、運用の機能で妥協することなしに RDB に対抗できるように NoSQL データベース自身を強化する必要がある。構造化／非構造化データ上で手軽に分析を実行できる NoSQL データベースの性能は、OLTP と OLAP を（最新バージョンの RDBMS の最大限の範囲で）処理できる RDB の能力と比較しても遜色ない。しかしながら、NoSQL データベース内にあるセキュリティの抜け穴が、卓越した分析性能で妥協することなく塞がれることは重要である。

伝統的なサービスプロバイダーがアナリティクス・アズ・ア・サービス（AaaS）として提供しているクラウドベースのソリューションは、中間データ処理のために利用される NoSQL データベースと共に、動的／静的データの双方を処理できるツールの組み合わせを利用して構築された分析フレームワークに基づいている。このようなシナリオでは、複数のユーザーがフレームワークを共有し、動的／静的データの双方を、分析フレームワーク経由で適切なコネクタに送り込む。これらのデータセットは、結果が個々のユーザーに送られる前に、中間処理のための NoSQL データベースにおいて保持される必要がある。現行の NoSQL セキュリティメカニズムにおいて、フレームワークの内部 NoSQL データベースを共有する、異なったクラウドユーザーに関連付けて機微なデータを分離することは仮想的に不可能である。

2.2 モデリング

パフォーマンスと拡張性という NoSQL の 2 つの優れた特徴によって可能となる、アーキテクチャの柔軟性が、同様に最大のセキュリティリスクを引き起こす[3]。NoSQL は、大規模データセットを処理する観点から、セキュリティの強調が制限された状態で設計された[4]。これによって、NoSQL における重要なセキュリティ上の不備が発生してきたが、本稿ではごく一部しか示されていない。セキュリティ標準の欠如により、ベンダーはボトムアップで NoSQL ソリューションを開発し、その場凌ぎでセキュリティ問題を処理してきた。NoSQL データベースの脅威モデルには、6 つの大きなシナリオがある：

1. 処理の完全性 – NoSQL の最も目に見える問題は、処理の完全性を保証するためのソフトなアプローチである。複雑な整合性制約をアーキテクチャに導入したら、よりよいパフォーマンスと拡張性を達成するという NoSQL 本来の目的を実現できないであろう。アーキテクチャトレードオフ分析（ATAM）のような手法は、アーキテクチャに係る意思決定（例えば、パフォーマンス対セキュリティ）におけるトレードオフを特に取り扱うものである。この分析手法を利用して、整合性制約のレベルを評価することが可能であり、パフォーマンスに重大な影響を及ぼすことなく、中核となるアーキテクチャのカーネルに整合性制約を注入することができる。

2. 緩やかな認証メカニズム – 総じて NoSQL は、弱い認証手法と弱いパスワード保存メカニズムを利用している。これにより、NoSQL は反射攻撃やパスワード総当り攻撃にさらされて、情報漏えい

の結果をもたらす。NoSQL は、HTTP Basic 若しくは Digest ベースの認証を利用しており、反射攻撃や中間者攻撃を招きやすい。もう 1 つのよく選択される通信プロトコルである REST もまた HTTP に基づいており、クロスサイトスクリプティング、クロスサイトリクエストフォージェリ、インジェクション攻撃などを招きやすい。結局のところ、NoSQL は、認証を強制するサードパーティのプラグイン可能なモジュールの統合をサポートしていない。REST 原則に従うコネクションの定義を操作することにより、基盤にあるデータベースのハンドルや構成パラメータにアクセスすることが可能であり、それによってファイルシステムにアクセスできる。既存の NoSQL データベースの中には、ローカルノードレベルでの認証を提供しているものもあるが、全てのクラスタノードを通して認証を強制することはできない。

3. 不十分な承認のメカニズム – 承認の手法は、NoSQL ソリューションによって異なる。一般的なソリューションの大半は、低いレイヤで承認を強制するよりも、高いレイヤで承認を適用する。特に、承認は、コレクションのレベルよりもデータベース単位のレベルで強制される。アーキテクチャに組み込まれたロールベースアクセス制御（RBAC）メカニズムが存在しないのは、RBAC メカニズムでユーザーのロールやセキュリティグループを定義することができないためである。

4. インジェクション攻撃に対する感受性 – 導入しやすいインジェクション手法により、悪意のある行動のためのファイルへのバックドアアクセスが可能になる。NoSQL アーキテクチャは、疎結合状態で軽量のプロトコルとメカニズムを導入しているため、JSON インジェクション、配列インジェクション、ビューインジェクション、REST インジェクション、GQL インジェクション、スキーマインジェクションなど、様々なインジェクション攻撃の影響を受けやすい。例えば攻撃者は、スキーマインジェクションを活用して、攻撃者が選択したデータを有するデータベース上に何千列も注入することができる。このような攻撃の影響は、破損したデータを有するデータベースから DoS 攻撃へと及び、データベース全体が利用できない結果となる。

5. 整合性の欠如 – 分散モードで、　CAP 定理（整合性、可用性、分割耐性）の 3 つの要素全てを同時に強制できなくなると、攪拌された結果の信頼性が損なわれる。結果として、個々の参画しているノードが最新のイメージを有するノードと完全に同期できない可能性があるため、ユーザーは、いかなる時でも一貫した結果が保証されないことになる。クラスタノードを通してデータを複製するよう任された現行のハッシュ化アルゴリズムは、単一ノードの失敗の場合には破損して、クラスタノード間の負荷の不均一化を招く結果となる。

6. 内部関係者による攻撃 – 許容性のあるセキュリティメカニズムは、内部関係者による攻撃を達成するために利用可能である。これらの攻撃は、貧弱なロギングやログ分析手法、その他初歩的なセキュリティメカニズムのために、気付かれない状態となり得る。重要なデータが、セキュリティの薄いレイヤ下に収容されると、データ所有者による制御の維持を保証することが難しい。

2.3 分析

データの完全性は、アプリケーション若しくはミドルウェアのレイヤを介して強制される必要がある。パスワードは、停止や転送の間に消去されるべきではないが、その代わり、セキュアなハッシュ化アルゴリズムを利用して、暗号化／ハッシュ化されるべきである。同様に、データベースに保存されたデータは、消去されるべきではない。既に貧弱な認証／承認手法が導入されている点を考慮すると、関連するパフォーマンスの影響に関わらず、停止中はデータを暗号化した状態に保つことが不可欠である。ハードウェアアプライアンスベースの暗号化／復号化やバルクファイルベースの暗号化は加速しており、暗号化のパフォーマンスへの影響に対する懸念は緩和されるであろう。もちろん、ハードウェアベースの暗号化は、しばしばベンダーロックインにつながるので、それ自身に対する批判がないわけではない。結果として、ファイルシステムにアクセスする悪意のあるユーザーは、直接ファイルシステムから機微なデータを抜き取ることが可能であろう。転送の間、機密性を維持するためには、SSL／TLS を利用して、クライアントとサーバー間および参加するクラスタノード同士のコネクションを確立することがよいプラクティスである。データ転送の間、このようなメカニズムを採用して、相互認証鍵を交換し、信頼を確立すれば、データの機密性が保証されるであろう。NoSQL アーキテクチャは、状況に応じて全てのレベルでセキュリティを強制する能力を有するプラグイン可能な認証モジュールをサポートすべきである。

クラスタ同士の通信についても、信頼された通信チャネルを確立する前に、個々のノードが他の参加するノードの信頼性レベルを検証できるように、よりよく制御されるべきである。インテリジェントハッシュ化アルゴリズムを利用することによって、ノード障害の際にも、データがノード間で複製されることを保証することができる。全ての NoSQL 製品／ソリューションが、信頼された環境上で稼動させて、信頼されたマシンだけがデータベースのポートにアクセスできることを保証することを推奨している。

適正なログインメカニズム、実行中のログ分析、ログ分析への相関の統合／適用は、潜在的な攻撃を洗い出すことを可能にするであろう。ファジング手法（不正な、予期されない、または無作為の入力を提供する）の適用は、HTTP によってクライアントとの通信の確立を図る NoSQL における潜在的な脆弱性を洗い出す重要な手段となり得る。適正なデータのタグ付け手法は、データをソースから結んでいる間にインテリジェントアルゴリズムを介して強制されるタイムスタンプと共に、権限のないデータ修正に対する防御となるであろう。これらの手法はまた、保存されたデータの信頼性を維持するであろう。

2.4 導入

脅威モデル／分析より、Web 若しくは同様のインタフェースの、薄くて簡単に侵入できるセキュリティのレイヤの中に機微なデータを包み込むことによって NoSQL の穴をふさぐことが、基盤にあるデータのセキュリティを保証するのに不十分であることは明らかである。ミドルウェアのセキュアなラッパーの下に NoSQL を隠す、若しくは Hadoop のようなフレームワークを利用して NoSQL にアクセス

することによって、NoSQL の境界周辺に仮想のセキュアなレイヤを生成することができる。コレクションまたは列レベルにおけるオブジェクトレベルのセキュリティについては、薄いデータベースのレイヤを維持しながら、ミドルウェアを介して誘導することができる。このような手法によって、データへの直接のアクセスがないことや、ミドルウェアまたはフレームワークのレイヤ内に構成された制御に基づいてのみデータが外に出ることが保証されるであろう。クラウド時代の分散型 Hadoop は、強力なケルベロス認証をサポートする。このようなメカニズムにより：

1. 悪意のあるユーザーの偽装の防止
2. 全ての遠隔手続き呼出（RPC）におけるユーザー認証の強制
3. HadoopMaster ノード、クラスタノード、ジョブトラッカー上でグループ解決を実行することによる、グループメンバーシップのユーザー操作の防止
4. ジョブを投入したユーザーのアカウント下で Map タスクが実行される時の適当な分離の保証

が可能になる。

下にある NoSQL 層をカプセル化するためにミドルウェアのレイヤを導入することは、セキュリティ導入のもう一つの選択肢である。ミドルウェアソフトウェアの大半は、認証、承認、アクセス制御のサポートをあらかじめ備えている。Java の場合、Java 認証／承認サービス（JAAS）と SpringSource、Spring セキュリティフレームワークが、認証、承認、アクセス制御のために導入されている。このようなアーキテクチャが、スキーマ、オブジェクト、そして／またはデータへのいかなる変更も妥当であることを保証し、その上で、NoSQL の能力を維持しながら、よりよい制御を実行する[6]。

パフォーマンスや需要に合わせる能力、システム全体のセキュリティを維持するためには、NoSQL をフレームワークに統合する必要があり、それによってセキュリティのコンポーネントをフレームワークに受け渡して負荷を軽減する。このようなフレームワークは、ポリシーベースのセキュリティレイヤがより低い下位の層（カーネルのレイヤ）に焼き付けられるように、基盤の OS と密に結合されるべきである。これによって、基盤にあるデータへのアクセスを制限し、データベースのレイヤの薄さを保持し、NoSQL の分析能力を維持するために、欠損したロールベースのアクセス制御（RBAC）を強制することが保証されるであろう。結局、このようなメカニズムが、データの所有者が自分のデータを上手に制御することを保証し、それによって内部者による攻撃を防止／暴露する。これによって、中間の状態で保持されたデータのセキュリティが強化され、共有のフレームワークアーキテクチャの中にある NoSQL データベース上で処理されて、単一または複数のクラウドサービスモデルを介したサービスとして分析が提供される。

脆弱な NoSQL データの代替として、暗号化はより良い保護策を提供する。Hadoop は、OS、プラットフォームやストレージ形態の区別なく、確固たる保護を提供するために、ファイルのレイヤの暗号化を導入している。暗号化を提供する能力を備えた製品の利用が可能になるにつれて、動的データの取り扱いやインメモリでの処理に対する需要が増大している。暗号化ソリューションは、多数ある既知のデータセキュリティ問題を明らかにする費用対効果に秀でた方法と思われる。

3.0 セキュアなデータ保存とトランザクションのログ

データとトランザクションのログは、多層のストレージメディアに保存される。手動で各層間をデータ移動させると、IT マネージャーにどのデータがいつ移動されたかを直接コントロールさせることになる。しかしながら、データセットの容量は指数関数的に増加し続けており、拡張性と可用性のためにビッグデータストレージ管理の自動階層化が求められる。自動階層化ソリューションは、どこにデータが保存されるか、どれがセキュアなデータ保存の新たな脅威となるかを追跡することはない。新たな機能として、権限のないアクセスを遮断し、常時、可用性を維持することが必須となる。

3.1 ユースケース

ある製造企業は、様々な部門からのデータを統合したいと考えている。このデータの中にはほとんど引き出されないものがある一方、同じデータプールを継続的に利用する部門もある。自動階層化ストレージシステムは、ほとんど利用されないデータをより下位の（安い）層に格納することによって、製造企業の経費を節約するであろう。しかしながら、このデータには、一般的ではないが重要な情報を含む研究開発結果が含まれている可能性もある。下位層ではしばしば低いセキュリティが提供されることがあるので、企業は、慎重に階層化戦略を研究すべきである。加えて、例えばテキストログのようなメタデータは、保護する必要のあるもう 1 つの軸を導入する。ログポイゾニング攻撃は、潜在的にデータの整合性の欠如を招くため、ユーザーの間で論争になっている。

3.2 モデリング

ネットワークに基づく分散型の自動階層化ストレージシステムは、透明性のあるサービス、優れた拡張性と弾力性など、先進的な機能を持つ有望なソリューションである。しかしながら、物理的所有物の欠如、信頼できないストレージサービス、あるいは整合性のないセキュリティポリシーにより、自動階層化ストレージは、新たな脆弱性を生み出す。自動階層化ストレージの脅威モデルには、7 つの主要なシナリオが含まれる。

1. 機密性と完全性 – これら機微な情報を盗んだり、ユーザーデータに損害を与えたりするための試みに加えて、ストレージサービスプロバイダーが信頼できない第三者であることも想定される。ストレージシステムにおける階層間のデータ転送は、サービスプロバイダーがユーザーの行動とデータセットを連動させることを可能にする手掛かりを提供する。暗号文を破る能力がなくても、特定のプロパティをさらけ出すことは可能である。

2. 来歴 – 極端に大きい容量のため、可用性と完全性を検証するために、データセット全体をダウンロードすることは実行不可能である。推定に基づいて正確で、計算処理や通信の負荷が低くて済むような証明を提供するためには、軽量のスキームが望ましい。

3. 可用性 – 自動階層化はまた、一貫した可用性の保証という問題をサービスプロバイダーに課する。低位層におけるより脆弱なセキュリティがサービス拒否（DoS）攻撃の危険にさらすだけでなく、低位層と高位層の間のパフォーマンスのギャップもまた、高速再保存や災害復旧時のバックアップウィンドウを拡張する。

4. 整合性 – 今や、層の間をデータが流れ、複数のユーザーによって共有されているのが一般的である。異なる場所に保存された複数の複製同士で整合性を維持することは重要である。慎重に処理する必要がある 2 つの課題は、書き込みのシリアライズ化と複数の書き手と複数の読み手（MWMR）の問題である。

5. 共謀攻撃 – データ所有者が自動階層化ストレージシステムで暗号文を保存し、鍵とアクセス許可をユーザーに配布する時、個々のユーザーは、データセットの特定の部分にアクセスできる権限が付与される。また、サービスプロバイダーは、暗号鍵の要素なしでデータを翻訳することができない。しかしながら、サービスプロバイダーがユーザーと共謀して鍵とデータを交換すれば、付与されていないデータセットを入手するであろう。

6. ロールバック攻撃 – 複数ユーザー環境において、サービスプロバイダーは、ユーザーにロールバック攻撃を仕掛けることができる。データセットの最新版がストレージ上で更新されると、サービスプロバイダーは、古いバージョンを送付してユーザーを騙すことができる。データが最新であることをユーザーが保証するのを助けるためには一定の証拠が必要であり、ユーザーは不統一であることを検知する能力を持つことが必要である。これはまた「ユーザーの鮮度」と呼ばれる。

7. 論争 – 記録保存の欠如は、ユーザーとストレージサービスプロバイダー間の論争をもたらす。データの損失や改ざんが発覚すると、責任を判断するために転送ログ／記録が重要となる。例えば、悪意のあるユーザーがデータをストレージシステムに外部委託する。後で、ユーザーがデータ損失を報告し、請求した損害に対する補償を求める。この場合、上手に保持されたログが効果的に不正を防止する。

3.3 分析

最近数十年間、情報の保証とサイバーインフラストラクチャセキュリティの分野が急速に発展してきた。今日、上述のセキュリティ問題を処理するための洗練された技術がある。機密性と完全性は、堅牢な暗号化技術とメッセージダイジェストで実現することができる。署名されたメッセージダイジェストの交換を、潜在的な論争を処理するために利用することができる[18]。ユーザーの鮮度と書き込みのシリアライズ化は、定期監査 [24]とチェインハッシュ [23]または一貫した認証ディクショナリ（PAD）[11] によって解決することができる。セキュアな信頼されていないデータレポジトリ

（SUNDR）は、フォーク整合性攻撃や書き込みのシリアライズ化を検知するために利用することができる。

線形と並行という 2 つの「lock-free」プロトコルが、単独の書き手と複数の読み手（SWMR）の問題を処理するために提案されてきた[21]。しかしながら、SWMR はケース依存型の問題であり、本稿の範囲を超えている。ブロードキャスト暗号化 [14]とキーローテーション [20] は、拡張性を改善するために利用することができる。研究者は、来歴の問題を処理する技術を提案してきた[22]。データの可用性は、取得可能性の証明または委任可能なデータ所有の手法により、高い確率で改善することができる[10][19]。

共謀攻撃に関しては、ユーザーが秘密鍵を交換しない限り、ポリシーベースの暗号化システム（PBES）が、共謀のない環境を成功裡に保証することができる[13]。もし、ユーザーが復号化されたコンテンツを交換することなく進んで秘密鍵を交換すれば、媒介された復号化システムが共謀攻撃を防ぐことができる。もし、ユーザーが復号化されたコンテンツを進んで交換すれば、デジタル著作権管理が共謀攻撃を防ぐことができる。2 つの否認不可プロトコルが、論争の問題を処理するために、最近提案されてきた [16][17]。

大規模の自動階層化ストレージシステムにおける各個人のセキュリティ問題のための技術が存在する一方で、それらをシームレスで全人的なソリューションに統合する、系統的な方法はない。異なる層の間の不統一なセキュリティポリシーは、層内のデータ転送をセキュア化するのに、追加的な問題を引き起こす。セキュリティ、ユーザビリティ、複雑性、費用の間のトレードオフを均衡させるために、さらなる検討が必要である。

3.4 導入

技術の不均質性、多様なセキュリティポリシーと費用の制約によって、多層ストレージシステムにおけるセキュリティ戦略が多様化している。データの機密性、完全性、可用性など、一般的なセキュリティ要件を満たすために、多くのストラクチャを採用することができる一方で、3 つの特別な点に注意する必要がある。

1. 動的なデータ操作 – 修正、複製、削除、挿入などの操作がより頻繁に起きるので、自動階層化ストレージシステムのデータセットは動的である。PDP スキーム [12]の拡張された動的バージョンは、対象鍵暗号のみに依存しているため、高い効率性を実現する。しかしながらこのスキームは、クエリの数が制限されているため、動的データ操作を完全にサポートすることができない。動的で委任可能なデータ所有（DPDP）の公式フレームワーク [15] は、サーバー計算処理における負荷の増加分の費用で、検知の確率を改善すると考えられる。簡易的な復元可能性証明（POR）[26]の拡張バージョンは、クラウドストレージにおける公的な証明可能性とデータのダイナミック性の双方をサポートしており、ネットワークベースの自動階層化ストレージシステムの特殊なケースとして考えることができる。

2. プライバシー保護 – 証明の手順を第三者の監査人（TPA）に外部委託する傾向があり、証明のプロトコルは、公的に検証可能であることが期待される。一見すると、プライバシー保護は、公的な検証可能性の要件と矛盾するように思われる。プライバシーを保護する公的監査のスキームは、ワンらにより[25]、クラウドストレージ向けに提案された。提案されたスキームは、無作為化マスキングに統合された準同型線形認証に基づき、TPA が異なる階層のサーバーに保存されたデータセットを監査する際にデータプライバシーを保護することができる。

3. セキュアな暗号化データの取り扱い – プライバシー保護監査に加えて、今日、計算処理業務を外部委託するためには、復号化することなく暗号文上で操作を実行する能力が要求される。完全準同型暗号化のスキームによって、より複雑な機能がサポートされ、これらの操作が可能になる[54]。「暗号化クラウドストレージ」[55] における最近の業績は、信頼されないインフラストラクチャ上におけるセキュアな IaaS ストレージ構築を可能にすることによって、もう 1 つのクラウドプラットフォームソリューションを提供する。

4.0 エンドポイントの入力の検証／フィルタリング

企業環境におけるビッグデータのユースケースの多くで、エンドポイントデバイスなど様々なソースからのデータ収集が要求される。例えば、セキュリティ情報イベント管理システム（SIEM）は、企業ネットワーク上にある数百万のハードウェアデバイスやソフトウェアアプリケーションからイベントログを収集する可能性がある。データ収集プロセスにおける重要な課題として、入力の検証がある。どのようにしてデータを信頼することができるのか？ どのようにして入力データのソースが悪意のないことを検証でき、どのようにして収集物から悪意のある入力をフィルタリングすることができるのか？ 検証とフィルタリングは、特に BYOD（Bring Your Own Device）モデルなど、信頼できない入力ソースにより引き起こされる手強い課題である。

4.1 ユースケース

気象センサーから収集されたデータや、iPhone アプリケーションから送信されたフィードバックの投票には、同じような検証上の問題が存在する。動機付けられた相手が、「不正を働く」仮想的なセンサーを生成したり、結果を偽装するために iPhone の ID になりすましたりすることが可能かもしれない。これが、収集したデータの容量によって一層複雑化して、読み込みデータ数／投票が数百万を超える可能性がある。このような業務を効率的に実行するためには、大規模なデータセットの入力を検証するアルゴリズムを生成する必要がある。

4.2 モデリング

入力検証のための脅威モデルには、4 つの主要なシナリオがある：

1. 相手方は、データを収集するデバイスを改ざんしたり、悪意のある入力を中央データ収集システムに提供するために、デバイス上で稼動するデータ収集アプリケーションを改ざんしたりする可能性がある。例えば、iPhone フィードバック投票の場合、相手方は iPhone（例．IPhone のソフトウェアプラットフォーム）を危険にさらす、若しくはユーザーのフィードバックを収集する iPhone アプリケーションを危険にさらす可能性がある。

2. 相手方は、なりすましのアイデンティティ（例．iPhone ID）を複数生成し、それから偽りのアイデンティティより悪意のある入力を提供することによって、データ収集システムに対し、ID クローニング攻撃（例．Sybil 攻撃）を実行する可能性がある。Sybil 攻撃の問題は、BYOD のシナリオにおいて一層深刻である。企業のユーザーは、自分自身のデバイスを持ち込んで、企業内ネットワークで使用することが認められているので、相手方がそのデバイスを使用して信頼されたデバイスのアイデンティティを偽り、その上で悪意のある入力を中央のデータ収集システムに提供する可能性がある。

3. 複雑なシナリオとしては、相手方が検知データの入力ソースを操作することができる場合が含まれる。例えば、温度センサーを危険にさらす代わりに、相手方は、検知した場所の温度をわざと変更して、悪意のある入力を温度収集プロセスに導入することが可能かもしれない。同様に、iPhone または iPhone 上で稼動する GPS ベースの測位アプリケーションを危険にさらす代わりに、相手方は、GPS 衛星シミュレーターを使用して、GPS 信号自体を危険にさらすかもしれない[7]。

4. 相手方は、害のないソースから中央収集システムへ転送中のデータを危険にさらすかもしれない（例．中間者攻撃または反射攻撃の実行による）。この問題については、第 7 章で詳細に議論する。

4.3 分析

上述の脅威モデルを仮定すると、入力の検証問題のためのソリューションは、(a) 相手方が悪意のある入力を生成して中央の収集システムに送ることを防止するソリューションと、(b)相手方が成功裡に悪意のあるデータを入力した場合に中央システムで悪意のある入力を検知／フィルタリングするソリューションの 2 つのカテゴリーに分類される。

相手方が悪意のある入力を送るのを防止するためには、改ざん防止ソフトウェアが必要であり、Sybil 攻撃に対して防御する。改ざん防止セキュア化ソフトウェアの設計／導入に関する研究は、学界および産業界の双方でとても長い歴史がある。設計／導入のためのツール、技術、ベストプラクティスの多くは、ソフトウェアから脆弱性を特定して取り除くために開発されてきた。しかしながら、脆弱性のない複雑なソフトウェアの開発はほぼ不可能である。さらに、PC ベースのソフトウェアプラットフォーム／アプリケーションのセキュリティは幅広く研究されてきたが、モバイルデバイスとアプリケ

ーションセキュリティは研究が活発な領域となっている。結果として、特定の相手方は、モバイルデバイスおよびその上で稼動するアプリケーションを危険にさらすことが可能になると、我々は想定している。ギルバートらは最近、生データから抽出したデータだけでなく、生のセンサーデータについても完全性を保証するために、信頼されたプラットフォームモジュール（TPMs）を使用することを提案した[8]。しかしながら、TPM は、モバイルデバイスで共通に見られるものではない。さらに、TPM が出現しても、相手方は、センサー入力（例．GPS 信号）を操作することができる。

ID クローニング攻撃や Sybil 攻撃に対する防御スキームが、ピアツーピアシステム、推薦者システム、自動車ネットワーク、無線センサーネットワークなど、様々な領域で提案されてきた[9] 。これらのスキームの多くは、Sybil 攻撃を防止するために、信頼された証明書と信頼されたデバイスを使用することを提案している。しかしながら、数百万の主体が設定された大企業において証明書を管理することは難しい。他のスキームの多くは、複数の偽のアイデンティティが、独立した正規のアイデンティティからの見込みよりも少ないリソースを所有しているかどうかの決定など、リソース検証の概念の変動を提案してきた。リソース検証は、Sybil 攻撃を防止する代わりに、思いとどまらせることによって、Sybil 攻撃に対する最低限の防御を提供する。

中央収集システムにおいて悪意のある入力を検知／フィルタリングするために、ビッグデータの強みを活用することができる。現実世界のデータ収集システムは数百万のソースから大容量のデータを収集するので、相手方からの入力は外れ値として見えるかもしれない。従って、既存の統計的類似性検知技術や線形検知技術が、悪意のある入力を検知／フィルタリングするために導入されるかもしれない。

4.4 導入

入力の検証／フィルタリングに、確実な方法はない。結果として、我々は実務に導入するためのハイブリッドな手法を推奨する。第 1 に、ビッグデータ収集システムの設計者は、セキュアなデータ収集プラットフォームとアプリケーションを開発するために、最高レベルの注意を払うべきである。特に、信頼されないデバイス上でアプリケーションが稼動する BYOD のシナリオを考慮すべきである。第 2 に、設計者は、システムに対するもっともらしい Sybil 攻撃や ID 成りすまし攻撃を特定し、その上で、攻撃を軽減するために費用対効果のある方法を特定すべきである。第 3 に、設計者は、決められた相手方が悪意のある入力を中央収集システムに送ることができる点を認識すべきである。呼応して、設計者は、相手方から悪意のある入力を検知／フィルタリングするためのアルゴリズムを開発すべきである。

5.0 リアルタイムのセキュリティモニタリング

ビッグデータとセキュリティは、ビッグデータインフラストラクチャを保護する点だけでなく、他システムのセキュリティを改善するのに役立つビッグデータ分析を加速させる点でも重なり合う。

最も難しいビッグデータ分析の問題の１つがリアルタイムのセキュリティモニタリングであり、(a)ビッグデータインフラストラクチャ自体のモニタリングと、(b)ビッグデータ分析のための同一インフラストラクチャ利用の２つの重要な観点がある。(a)の例は、ビッグデータインフラストラクチャを構成する全てのノードのパフォーマンスと健全性のモニタリングである。(b)の例は、モニタリングツールを使用して不正請求を調べる医療機関、または同様のビッグデータツールを使用してより良いリアルタイムの警告／コンプライアンスモニタリングを実現するクラウドプロバイダーである。これらの改善は、偽陽性の数の減少と／または真陽性の質の向上をもたらす可能性がある。本稿では、双方の見方に注目する。

セキュリティデバイスによって数多くの警告が生成されると、リアルタイムのセキュリティモニタリングが問題になる。これらの警告は（相互関係の有無に関わらず）大量の数の偽陽性につながり、人間の分析能力の限界によってしばしば見過ごされる。この問題は、データの流れの容量や速度によって、ビッグデータと共に増大する可能性がある。しかしながら、ビッグデータ技術は、異なるタイプのデータを迅速に処理／分析する機会をもたらす可能性がある。これらの技術は、例えば、拡張性のあるセキュリティ分析に基づいてリアルタイムのアノマリ検知を提供するために利用することが可能である。

5.1 ユースケース

用途によって異なる可能性があるが、産業や政府機関の大半はリアルタイムセキュリティ分析から恩恵を受ける。共通の利用には、「誰が、どのデータに、どのソースから、いつアクセスしているのか」「攻撃を受けているのか」「A という行動のせいで、コンプライアンス基準 C に違反していないか」などの質問に答える技術の活用が含まれる。

これらの分析領域は新しいものではないが、改善されたデータ収集／分析によって、より迅速で適切な意思決定が可能になる（例．偽陽性の減少）。これらの改善された分析領域に加えて、新たな用途が定義されたり、既存のビッグデータに相対する用途を我々が再定義したりすることが可能である。

例えば、健康医療産業の場合、潜在的に納税者向けに高額のお金を節約したり、請求の支払がより正確になったり、請求に関連する不正を削減したりするなどして、ビッグデータの大きな恩恵を享受する。同時に、保存された記録は非常に機微であり、患者のプライバシーに関わる規制を遵守しなければならない。その結果、医療データは慎重に保護されなければならない。意図した若しくは意図しな

い個人情報の異常な取得をリアルタイムに検知することによって、医療機関は、損害を迅速に修復し、さらなる誤使用を防止することが可能となる。

5.2 モデリング

セキュリティモニタリングでは、ビッグデータインフラストラクチャ、或いはプラットフォームが本質的にセキュアであることが求められる。ビッグデータインフラストラクチャに対する脅威には、アプリケーションまたはノードへの不正を招く管理者のアクセスや（web）アプリケーションの脅威、回線の傍受が含まれる。このようなインフラストラクチャは、大抵異なるコンポーネントのエコシステムであり、(a)各コンポーネントのセキュリティと(b)これらコンポーネントのセキュリティ統合を考慮しなければならない。例えば、パブリッククラウドで Hadoop クラスタを稼動させる時、考慮しなければならない点がある：

1. パブリッククラウドのセキュリティであり、それ自体はコンピューティング、ストレージ、ネットワークのコンポーネントから構成されるコンポーネントのエコシステムである。
2. Hadoop クラスタのセキュリティ、ノードのセキュリティ、ノードの相互接続、そしてノードに保存されたデータのセキュリティ。
3. モニタリングアプリケーション自体のセキュリティであり、適用可能な相互関係のルールが含まれ、セキュアなコーディングの原則およびベストプラクティスに準拠しなければならない。
4. データが由来する入力ソース（例．デバイス、センサー）のセキュリティ。

もう一つの主要な脅威モデルは、それらを特定するために利用されるビッグデータ分析ツールを回避しようとする相手方の周辺で展開される。攻撃者は、検知されることを防止するために、回避攻撃[51]を起こしたり、ビッグデータ分析のアルゴリズムを訓練するために利用されるデータセットの信頼性の低下を狙ったデータポイゾニング攻撃 [52]を起こしたりすることも可能である。

これらのセキュリティの脅威とは別に、法規制のような他の障壁が重要になっている。モニターされるデータがどこに存在するか如何で、特定の法律が適用される可能性がある。データによってはセキュリティモニタリングができない可能性があり、また特定の形式（例．匿名化）のみで利用可能なことがあるので、これによって障壁が作られるかもしれない。

5.3 分析

ビッグデータ分析は、クラスタへの異常な接続をモニタリングしたり、ログのイベントをマイニングして疑わしい行動を特定したりするために利用することができる。

マイニング／分析アルゴリズムを導入する人々は、潜在的な回避またはポイゾニング攻撃を軽減するために、計算処理を行う統計に関して対立する問題に注意すべきである。加えて、ビッグデータを分析に利用することに対する法的／倫理的見方については、依然として多くの論争がある。ビッグデータのモニタリングは、考慮すべき様々な要因（例．技術的、法的、倫理的）の組み合わせである。次章では、データ管理を改善するために、我々がビッグデータ分析システムに組み入れることができるプライバシー保護メカニズムに焦点を当てる。

5.4 導入

セキュリティのベストプラクティス導入は目前の状況に依存する。本稿を書いている時点で、Hadoopにはあらかじめ組み込まれたセキュリティモニタリング／分析ツールは存在しない。しかしながら、モニタリング／分析ツールは、様々なHadoopプロバイダー／ベンダーによって開発／発表されている。もう１つのソリューションは、Hadoopの要求（例．ファイアウォールのDatabase Activity Monitoringプロキシ）をモニタリングするフロントエンドシステムの導入である。アプリケーションセキュリティは、アプリケーション自体とあらかじめセキュリティ制御が組み込まれているか否か（例．OWASPガイドラインへの準拠）に依存する。リアルタイムモニタリングに関しては、リアルタイムモニタリング向けのソリューションやフレームワーク（例えばNISTのセキュリティ設定共通化手順（SCAP））が、ビッグデータの領域で徐々に導入されている。Hadoopにおけるこれらのツールに関しては、バッチをベースとするもののみであり、時系列または傾向分析には役立つが、リアルタイムモニタリング向けではないという批判が１つある。この障壁を克服する試みの例として、Storm（storm-project.net）とApache Kafkaがある。その他のリアルタイムストリーミングアプリケーションについては、Hadoopに組み込まれたものが今市場に参入しているところである。

6.0 拡張性があり構成可能なプライバシー保護データマイニング／分析

ボイドとクロフォードが述べたように[27]、ビッグデータは、潜在的にプライバシーの侵害、侵略的なマーケティング、市民の自由の制限、国家や企業によるコントロールの増大を可能にする。

最近、企業のマーケティングを目的としたデータ分析の活用方法に関する分析により、どのようにして、当人の父親が知る前に十代の若者が妊娠したことを小売事業者が確認することができるかが事例として示された[28]。同様に、ユーザーのプライバシーを保持するために、分析用データの匿名化だけでは十分でない。例えばAOLは、学術目的で匿名化された検索ログを公表したが、その検索者によって簡単にユーザーが特定された[29]。Netflixは、同社の映像スコアをIMDBのスコアで修正することで匿名化したデータセットのユーザーが特定されてしまった時、同様の問題に直面した。

このようなことから、意図しないプライバシーの公開を防止するためのガイドラインや推奨を策定することが重要である。

6.1 ユースケース

大規模組織によって収集されたユーザーデータは、内部の分析者、場合によっては外部委託先やビジネスパートナーによって継続的にマイニング／分析される。悪意のある内部関係者や信頼できないパートナーが、これらのデータセットを悪用して、顧客からプライベートな情報を抜き出すことは可能である。

同様に、諜報機関は膨大な量のデータの収集を必要とする。データソースは多岐に渡り、チャットルーム、個人のブログやネットワークルーターが含まれる可能性がある。しかしながら、収集されたデータの大半は元来悪意のないものであり、保存したり　匿名化を維持したりする必要はない。堅牢で拡張性のあるプライバシー保護マイニングアルゴリズムによって、適切な情報を収集し、ユーザーの安全性を高める機会が増える。

6.2 モデリング

複数の脅威モデルが、ビッグデータストアにおけるユーザーのセキュリティを危険にさらすことができる。悪意のある攻撃者は、データをホスティングする企業にある脆弱性を悪用することによって（例えば、Lulzsec ハッカーグループが、HBGary Federal を含む複数のサイトからデータを取得した場合）、データストアを危険にさらすことができる。ユーザープライバシーの脅威モデルには、3 つの主要なシナリオがある：

1. ビッグデータストアをホスティングする企業の内部者はアクセスのレベルを悪用して、プライバシーポリシーを破ることができる。このシナリオの例としては、Google のチャットのやり取りをモニタリングすることによって、ティーンエイジャーをストーカーした Google の従業員のケースがある[30]。

2. データを所有する当事者がデータ分析を外部委託している場合、信頼できないパートナーが、データへのアクセスを悪用して、ユーザーからプライベートな情報を割り出すことができるかもしれない。通常、クラウドインフラストラクチャ（データが保存／処理されるところ）はデータ所有者によって制御されていないので、このケースをクラウドにおけるビッグデータ利用に適用することができる。

3. 研究目的のデータ共有はもう 1 つの重要な用途である。しかしながら、この章のはじめに指摘したように、再識別化の理由から、公開されたデータが完全に匿名化されていることを保証するのは難しい。EPIC の再識別化についての定義は、匿名化された個人データが真の所有者と照合されるプロセスとなっている。再識別化に関する多くの例が、EPIC の Web サイトで閲覧できる[31]。

6.3 分析

ユーザーのプライバシーを保護するためには、継続的なモニタリングによる悪用の防止／検知のベストプラクティスを導入する必要がある。プライバシー保護分析は、悪意のある行為者のデータセットからの成功を最小化するのに役立てることができる、研究の新領域である。しかしながら、現時点で実用的なソリューションはほとんどない。

差分プライバシーは、プライバシー保護に向けたよき第１ステップである。差分プライバシーは、プライバシーの形式的モデルを定義しており、計算処理上の負荷とノイズのある結果をデータ分析結果に加えた代償として、モデルを導入しセキュアなことを証明することが可能になる。恐らく、現行の差分プライバシーの定義は超保守的であり、新たなもっと実用的な定義によって、この原則の導入に関連する費用が示されるかもしれない。

アウトソーシングされた計算処理のリソースに対するもう１つの潜在的なソリューションは、ユニバーサルな準同型暗号化であり、アウトソーシングしたデータの暗号化を維持しながら、データ分析を提供することを約束する。この技術は現時点で揺籃期にあり、今導入するのは実用的でないが、長期的な研究の約束された領域である。

プライバシーは構成中に保護されなければならない。言い換えると、プライベートな情報の漏えいは、複数のデータベースがリンクしている場合でも、制御される。匿名化されたデータ間の整合性を維持する必要があるので、匿名化されたデータストアをリンクさせることは難しい。

6.4 導入

悪意のある外部者からの攻撃を防ぐ基本的な技術の導入原理には、停止状態でのデータ暗号化、アクセス制御、承認メカニズムが含まれる。利用可能な脆弱性を最小化するために、最新のセキュリティソリューションでソフトウェアインフラストラクチャの修正を当て続けることも重要である。

内部者による潜在的な悪用を最小化するために、ビッグデータのオペレーターは、義務原理の分離に従ってシステムを設計する必要があり、悪意のある内部者に共謀を強制するであろう。加えて、データセットへのアクセス履歴を記録するための明確なポリシーが、フォレンジックを支援し、抑止力として機能することを可能にして、潜在的な悪意のある内部者に対し、彼らの行動を追跡することができることを知らしめる。

プライバシー保護の文脈におけるデータ共有は、現段階では新たな研究領域である。このシナリオのためのベストプラクティスで推奨されるのは、再識別化を認識することと、匿名化がプライバシーを保証するには不十分である点を知ることである。

本稿は技術的な分析に注目しているが、導入に際してはまた、ユーザーのプライバシー規制に従う必要がある。例えば、欧州連合諸国は、個人データ保護指令 95／46／EC、プライバシーおよび電気通信に関する指令 2002／58／EC、電気通信で生成若しくは処理されたデータの保持指令 2006／46／EC に

よって規制される。米国では同様に、保存されたデータへのアクセスは、電気通信プライバシー保護法によって規制され、動画の分析は、米国愛国者法および通信傍受法によって規制される。ポリシー、法規制およびそれらとプライバシーの関係についての詳細な情報は、ディフィーとランダウの文献で見ることができる[53]。

7.0 暗号化により強制されたデータ中心のセキリティ

個人、組織、システムなど、異なる主体に対するデータの表示を制御するためには、根本的に異なる 2 つの方法がある。第 1 の方法は、OS、ハイパーバイザーなど、基盤のシステムへのアクセスを制限することによって、データの可視性を制御する。第 2 の方法は、暗号化を利用して、保護シェルにデータ自身をカプセル化する。いずれの方法にも、利益と不利益がある。歴史的に、第 1 の方法は導入がより簡単であり、暗号化で保護された通信と組み合わせることによって、計算処理／通信インフラストラクチャの大半の標準となる。

しかしながら、システムベースの方法は、おそらく非常に大きな攻撃面をさらし出す。システムセキュリティの文書には、アクセス制御の導入の回避や、直接データへのアクセスを狙った基盤システム上での攻撃（バッファ・オーバーフローや特権昇格）が充満している。他方、暗号化を介したエンドツーエンドのデータ保護は、より小さくてより明確な攻撃面をさらし出す。隠れたサイドチャネル攻撃 [36], [37] によって秘密鍵を抜き取ることは可能であるが、これらの攻撃は実装が非常に難しく、無害化された環境を必要とする。

7.1 ユースケース

ビッグデータは多様なエンドポイントに由来しており、多くの個人データを含むので、ソースにおけるデータの可視性を連携させることが次第に不可欠となっている。医療保険の相互運用性と説明責任に関する法律（HIPAA）のような法的フレームワークは、機微なデータが危険にさらされた後で、責任を有する主体を巻き込むのに役立つにとどまる。その時点で、損害はすでに起きているのだ。

これが、暗号化されたデータの索引付け、分析、意味のある処理に関する直接の問題を提起する。データの機密性の問題に補足して、データの完全性については、特に多様なソースを有するデータセットにおけるデータのポイゾニング防止を保証する必要がある。

7.2 モデリング

暗号化における脅威モデルは、プロトコルを導入するシステムと相手方の間の相互作用を介して、数学的に定義されるものであり、外部から可視化された通信にアクセスし、入力パラメータの確率的多項式時間関数を計算することができる。相手方がシステムの特定のプロパティを計算する機会がほとんどなければ、そのシステムはセキュアだと思われる。4 つの主要なシナリオの脅威モデルがある：

1. 暗号化を利用して、暗号により強制されたアクセス制御手法のために、相手方は、正確な平文と不正確な平文を選ぶ機会があったとしても、暗号文を探すことによって、対応する平文データを特定できるようなことがあってはいけない。これは、アクセス制御ポリシーによって排除された当事者が、相互間や相手方との間で共謀した場合でも、有効でなければならない。

2. 暗号化されたデータの検索／フィルタリング目的の暗号プロトコルのために、相手方が、対応する述語が満たされているか以上のことについて知り得ることがあってはならない。最近の研究では、悪意のある主体が平文またはフィルタリングの基準について、意味のあることを知ることがないように、検索述語自体を隠すことにも成功している。

3. 暗号化されたデータの計算処理を目的とする暗号プロトコルのために、相手方は、正確な平文と不正確な平文を選ぶ機会があったとしても、暗号文を探すことによって、対応する平文データを特定できるようなことがあってはいけない。相手方はオリジナルのデータの暗号化に関わる任意の関数の暗号化を計算処理することができるので、これが極めて厳格な要件であることに注意してほしい。実際、通常の暗号化のための選択暗号文セキュリティと呼ばれる、より強力な脅威モデルについて、この文脈 の中で意味のある相手方は存在せず、このようなモデルを発見するための探索が続いている[38]。

4. 特定のソースに由来するデータの完全性を保証する暗号化プロトコルのために、様々な脅威モデルが存在する。中核となる要件は、相手方が、意図されたソースに由来しないデータを偽造できないようにしなければならない点である。また、ソースがグループの一部としてのみ特定できるという意味で、ある程度の匿名性が存在し得る。加えて、特定の状況下（おそらく法的）において、信頼される第三者が、データを既存のソースに結びつけることが可能でなければならない。

7.3 分析

1 つの脅威モデルがあると仮定すると、暗号化プロトコルの候補は、削減の引数かシミュレーションの引数かによって、脅威に対する防御であることが証明される。削減の引数は、相手方がシステムの特定のプロパティを計算処理できるかを検証して、困難だと幅広く想定されていた、多くの論理的プロパティを破ることができる。あるいはまた、それは、システムのビルディングブロックとして利用されていた、より単純な暗号プリミティブのセキュリティを破ることができる。カネッティが言及しているような [39]、シミュレーションの引数においては、相手方が候補のシステムを破り、それから

「シミュレーター」が、本来、システムのセキュリティを代表する理想的な機能を破ることができる。我々は、個々のモデルについてセキュアであると証明されている上述の領域における現在の研究をいくつか示す。

1. アイデンティティと属性に基づく暗号化手法 [49] [50]は、暗号化を利用してアクセス制御を強制する。アイデンティティに基づくシステムでは、平文を所与のアイデンティティのために暗号化することが可能であり、そのアイデンティティを有する主体だけが暗号文を解読できることが期待される。他のいかなる主体も、たとえ共謀しても平文を解読することができない。属性ベースの暗号化は、この概念を属性ベースのアクセス制御に拡張したものである。

2. ボネとウォーターズ[40]は、比較クエリ、サブセットクエリ、そしてこれらのクエリの任意の結合をサポートする公開鍵システムを構築している。

3. 2009 年の画期的な結果[41]で、ジェントリーは完全準同型暗号化スキームを構築した。このようなスキームにより、元の平文の任意関数の暗号化を処理することが可能になる。早期の結果[42]では、部分的な準同型暗号化スキームを構築した。

4. グループ署名[43]により、個々の主体が各自のデータに署名しながら、公開されたグループのみで特定可能な状態を維持することができる。信頼された第三者のみが、個人のアイデンティティを特定することができる。

7.4 導入

アイデンティティ／属性ベース暗号化スキームとグループ署名を導入する現行のアルゴリズムは、双線形ペアリングマップをサポートする楕円曲線グループを利用する。これにより、グループの要素は、幾らか大きく表現される。加えて、ペアリング作業は計算処理上高価である。

ジェントリーの完全準同型暗号化（FHE）スキームの元来の構築では、多項式環を介して理想的な格子を利用していた。格子の構造は極端に非効率ではないが、FHE の計算処理のオーバーヘッドは、依然として実用的には程遠い。より簡単な構造[44] [45]、効率の改善[46],47]と、興味深い関数族に十分な部分的準同型スキームを求める研究が進行中である。

8.0 粒度の高いアクセス制御

アクセス制御の観点から問題となるセキュリティの特性は機密性であり、アクセスすべきでない人によるデータへのアクセスを抑制することである。過程の細かいアクセスメカニズムの問題は、そうでなければ共有されたであろうデータが、目に見えるセキュリティを保証するために、より厳格な分類

へと排除されることがよくある点である。粒度の高いアクセス制御によって、機密性に妥協することなく、データを共有する際にさらなる精度がデータ管理者に付与される。

8.1 ユースケース

ビッグデータ分析やクラウドコンピューティングでは、次第に、スキーマの量およびセキュリティ要件の量で膨大なデータセットを処理する点に注目が集まっている。データに関する法律およびポリシーの制限は、様々なソースに起因している。サーベンス・オクスリー法（SOX 法）は企業の財務情報を保護するための要件を設けており、医療保険の相互運用性と説明責任に関する法律(HIPAA)は、個人健康記録の共有に関する様々な制限を含んでいる。米国機密保護法の大統領令 13526（Executive Order 13526）は、国家安全情報保護のための精巧なシステムを概説している。また、プライバシーポリシーや共有同意書、企業ポリシーも、データの取り扱いに関する要件を示している。

この過度な制限を管理することによって、アプリケーション開発費用の増大や、誰もほとんど分析に参加できない壁に囲まれた庭のようなアプローチを招く結果となった。粒度の高いアクセス制御は、この急激に複雑化するセキュリティ環境において、分析システムを適合させるために必要である。

8.2 モデリング

粒度の高いアクセス制御に関わる基本的な脅威のベクトルは、アプリケーションレイヤへの整合性のない取り込みに起因している。粒度の高いアクセスを修正するためには、厳格な方法が要求される。これによって、アプリケーション開発が高価で複雑なものになるだけでなく、様々なアプリケーションが、粒度の高いアクセス制御を誤解する機会を多く提供する。

粒度の高いアクセス制御は、3 つの部分問題に分解することができる。第 1 は、個人のデータ要素に必要な機密性の要件を追跡し続けることである。多くの異なるアプリケーションで共有される環境下では、摂取者（データを提供するこれらのアプリケーション）は質問をする人にこれらの要件を伝える必要がある。この調整の要件はアプリケーション開発を複雑化させ、しばしば、複数の開発チームの間に分散される。

これらの要件を追跡する際のさらなる課題は、分析の変換を通してアクセスのラベルを維持することである。2 つないしそれ以上の要素から生成された要素は、ある格子に従って、他の要素の制限の少なくとも上限で制限される可能性がある。しかしながら、データアクセスの要件の中には、広く公開可能な、集約された医療記録のように、格子に従わないものがあり、これらの集約に寄与する要素は厳しく制限されている。これらの要件の追跡はまたアプリケーション空間に導入されており、正しくやることが難しくなる可能性がある。

第 2 の部分問題は、ユーザーの役割と権限を追跡し続けることである。一度ユーザーが適切に認証されると、1 つないしそれ以上の信頼されたソースから、そのユーザーのセキュリティに関連する属性を

引き出す必要が依然としてある。LDAP、Active Directory、Oauth、OpenID その他多くのシステムは、この空間で成熟化し始める。継続的な課題の１つは、権限の空間を適切に連携させることであり、それで単一の分析システムが、広いエコシステムを通して定義された役割や権限を尊重することができる。

第３の部分問題は、強制的なアクセス制御と共に、機密性の要件を適切に導入することである。これは、データに伴う要件と、アクセスの意思決定をするユーザーに伴う属性を組み合わせる、論理的なフィルターである。粒度の高いアクセス制御の適切なレベルをサポートするインフラストラクチャの構成要素がほとんどないため、このフィルターは、通常、アプリケーション空間に導入される。

8.3 分析

粒度の高いアクセス制御に関わる基本的な脅威のベクトルは、アプリケーション空間における導入への信頼性である。この脅威を軽減する自然な方法は、粒度の高いアクセス制御をアプリケーションに追加することによる複雑性を軽減することである。言い換えると、インフラストラクチャのレイヤにできるだけ多く導入し、アプリケーション空間に依然として存在するこれらの懸念を簡略化する標準とプラクティスを適合させることである。

第１の課題は、所与の領域に要求される粒度の適正なレベルを選ぶことである。列レベルの保護において、列は単一の記録を表しており、しばしば、データソースによって変わるセキュリティに関連している。行レベルの保護において、行は全ての記録を横断する特定のフィールドを表しており、しばしば、機微なスキーマの要素に関連している。列と行のレベルのセキュリティによる組み合わせは、より粒度が高いが、依然として分析利用中に細分化することができる。テーブルの変換或いは質問に注目したデータセットはしばしば列または行の方向付けを保護しないことがあるので、より粒度の細かいソリューションを必要とする。セルレベルのアクセス制御は、公開しやすさによって、全ての不可分な情報の集まりの分類をサポートしており、幅広いデータ変換の用途をサポートすることができる。

残念ながら、アクセスレベルの隙間を狭めることは困難である。世界中に分散したネットワーク上で、数多くの寄与者と共に、ペタバイト規模のデータセットまでソリューションを拡張する時、いくつかの拡張性の問題が生じる。これらの問題には、権限とアクセス制限が時間と共に変化する際のラベルの保存、クエリ時間のアクセスチェック、アップデートの費用が含まれる。

費用を低く抑えるためには、時間の変化に伴う見通しをモデル化することが重要である。例えば、ログがいつ生成されたか、何のシステムが生成したかなど、比較的変わらないログデータの属性がある。また、現在の雇用あるいは現在の訓練など、非常に変わりやすいユーザー関連属性もある。読み書きやシステムメンテナンスの費用をモデル化する時、クエリのパフォーマンスを改善するために、変わらない要素が非正規化される一方で、アップデート費用を削減するために、変化しやすい要素が正規化されるソリューションを選択するであろう。これによって、粒度の高いデータ要素と共に、アクセ

ス制御要件の多くを明白に保存する方法が構築され、クエリの時間に合わせて動的に、既存の属性に加わることになる。

8.4 導入

粒度の高いセキュリティのアクセスを導入するためには、ビッグデータのエコシステムを繋ぐ要素が必要になる。データと共にアクセス制限を追跡するためのプロトコルが必要であり、HDFS や NoSQL データベースなど、ストレージシステムに導入されるべきである。本稿で取り上げたその他の問題の多くは、認証、強制的なアクセス制御を含む、完全な粒度の高いアクセス制御ソリューションのための必要条件である。

Apache Accumulo は、成熟したセルレベルのアクセス制御をサポートする NoSQL データベースの例である。Accumulo では、全ての不可分な鍵／値のペアが、そのエントリを読むために要求されるロールを記述する表現とタグ付けされ、全てのクエリにロールチェックが含まれる。多くの用途で、Accumulo における圧縮やキャッシングの手法は、パフォーマンスに軽微なインパクトを与え、アプリケーションに導入される必要があるきめの細かいアクセス制御の要素を大いに簡素化する。

9.0 粒度の高い監査

リアルタイムのセキュリティモニタリング（5 章参照）を利用して、攻撃が起きた瞬間に通知されることが目標となる。実際には、これがいつも当てはまるとは限らない（例．最新の攻撃で、本当は正しいのに見落とされた場合）。見落とされた攻撃の真相を究明するためには、我々は監査情報が必要である。これは、何が起きて、何を誤ったのかを理解するためだけでなく、コンプライアンスや法規制、フォレンジックの理由からも重要である。監査は目新しいものではないが、リアルタイムのセキュリティの文脈において、適用範囲や粒度が異なることがある。例えば、これらの文脈では、より多くのデータオブジェクトが存在し、それらは（必ずしも必要ではないが）分散している可能性がある。

9.1 ユースケース

コンプライアンス要件（例．PCI、SOX 法）は、金融機関に対し、粒度の高い監査記録を要求する。加えて、プライベートな情報を含む記録の損失は 1 件当たり 200 ドルと推計されている。地理的な場所にもよるが、データ違反の場合、その後に法的手続が取られる可能性がある。金融機関の主要人員は、社会保障番号（SSN）など個人識別情報（PII）を含む大規模なデータセットへのアクセスを要求する。もう 1 つの潜在的ユースケースとして、マーケティング企業は、オンライン広告に関する顧客中心の手法を最適化するために、個人のソーシャルメディア情報にアクセスしたいと考える。

9.2 モデリング

監査の主要な要因としては、以下のものが含まれる：

1. 要求される監査情報の完全性（例．デバイスまたはシステムから、全ての必要なログ情報にアクセスする）
2. 監査情報へのタイムリーなアクセス。これは、例えばどこが期限厳守の時かなど、フォレンジックの場合、特に重要である。
3. 情報の整合性、または言い換えると、改ざんされていない監査情報。
4. 監査情報への承認されたアクセス。承認された人間だけが情報のうち職務を実行するのに必要な部分のみにアクセスすることができる。

これら主要な要因に対する脅威（例．権限のないアクセス、データの除去、ログファイルの改ざん）が監査データやプロセスを危険にさらす。

9.3 分析

ビッグデータインフラストラクチャを超えた監査機能を可能にする必要がある。特定の機能は、サポートされたインフラストラクチャ構成要素の監査機能に依存する。例として、ネットワークの構成要素（例．ルーターの syslog）、アプリケーション、OS、データベースからのログ情報がある。異なる構成要素（しかしながら全ての構成要素が必要な情報を送信できるわけではない）の利用可能な監査情報を活用して、攻撃の凝集した監査画面を生成することが課題となる。

9.4 導入

監査機能の導入は、個々の構成要素レベルから始まる。例として、ルーターの有効な syslog、アプリケーションのログ付け、OS レベルの有効なログ付けがある。以降、フォレンジックまたはセキュリティ情報／イベント管理（SIEM）ツールが、この情報の収集、分析、処理を行う。記録の量は、監査データの容量や速度を処理する際の SIEM ツールの限界に左右されて、それなりに、ビッグデータインフラストラクチャによって処理される必要がある。ビッグデータインフラストラクチャの利用とこのインフラストラクチャの監査を分離するためには、フォレンジック／SIEM ツールを、実現可能な時に、ビッグデータインフラストラクチャの外部で導入し、利用することが推奨される。もう 1 つの方法は、「監査レイヤ／オーケストレイヤー」を生成することであり、監査人から、必要な（技術的な）監査情報を抽出する。このオーケストレイヤーが、監査人の要求を取り込み（例．D の日付にオブジェク

トXのデータに誰がアクセスしたか）、必要な監査情報を必要なインフラストラクチャの構成要素から収集して、この情報を監査人に返す。

10.0 データ来歴

来歴を可能にするビッグデータアプリケーションのプログラミング環境から生成される大規模な来歴グラフにより、来歴のメタデータは複雑化していく。メタデータのセキュリティ／機密性アプリケーションへの依存度を検知するために行う、このような大規模の来歴グラフ分析は集中的なコンピュータ処理になる。

10.1 ユースケース

いくつかの主要なセキュリティアプリケーションは、生成に関する詳細など、デジタル記録の履歴を要求する。例えば、金融機関のインサイダートレーディングを検知したり、研究調査のためにデータソースの正確性を決定したりする場合がある。これらのセキュリティ評価は元々時間に厳格であり、この情報を含む来歴のメタデータを処理するために、速いアルゴリズムを要求する。加えて、データ来歴は、PCI、SOX 法など、コンプライアンス要件のための監査ログを補完するものである。

10.2 モデリング

ビッグデータアプリケーションのセキュアな来歴については、来歴記録が信頼でき、来歴が統合され、プライバシーが保護され、アクセス制御可能であることが最初に要求される。同時に、ビッグデータの特徴から、来歴の可用性と拡張性についても慎重に処理されるべきである。特に、ビッグデータアプリケーションの来歴メタデータに対する脅威は、公式上、3 つの分類にモデル化できる。

1. インフラストラクチャの構成要素の誤作動 – 多数の構成要素が協働して、来歴可能なプログラミング環境から、大規模な来歴グラフを生成するビッグデータアプリケーションにおいて、インフラストラクチャの構成要素がいくつか散発的に誤作動する可能性があるのは避けられない。一度誤作動が発生すると、来歴データがタイムリーに生成できなくなり、いくつかの誤作動によって、不正確な来歴記録につながる可能性がある。その結果、インフラストラクチャの構成要素の誤作動によって、来歴の可用性や信頼性が損なわれるであろう。
2. インフラストラクチャ外部攻撃 – 来歴は、ビッグデータアプリケーションのユーザビリティにとって重要であり、当然、ビッグデータアプリケーションの標的になる。外部の攻撃者は、記録を傍受

／分析することによって、来歴のユーザビリティを破壊したり、プライバシーを侵すために、転送される間に来歴記録を偽造、修正、再生したり、過度に遅延させたりすることができる。

3. インフラストラクチャ内部攻撃 – 外部攻撃と比較して、インフラストラクチャ内部攻撃は、より有害である。内部攻撃者は、ビッグデータアプリケーションの来歴システムを破壊するために、保存された来歴記録や監査ログを修正／削除する可能性がある。

10.3 分析

上述の脅威に対処するために、ビッグデータアプリケーションにおけるセキュアな来歴の信頼性とユーザビリティを保証する 2 つの研究課題に取り組む必要がある（例．来歴収集のセキュア化、来歴の粒度の高いアクセス制御）。

来歴収集のセキュア化のために、インフラストラクチャで来歴を生成するソースの構成要素を最初に認証する必要がある。加えて、ソースの構成要素の健全な状態を保証するために、定期的なステータスの更新を生成する必要がある。来歴記録の正確性を保証するために、偽造／修正されていないことを保証する完全性チェックを通して、来歴が実行される必要がある。さらに、整合性の欠如は誤った意思決定につながる可能性があるので、来歴とそのデータの整合性についても検証される必要がある。時々、データに関連して機微な情報が来歴に含まれることから、来歴の機密性とプライバシー保護を達成するために、暗号化技術が要求される[32] 。最後に、小容量で静的なデータのアプリケーションと比較して、ビッグデータの来歴収集では、増え続ける容量、種類、速度の情報を効率的に収容する必要がある。このようにしてセキュアな来歴収集が、ビッグデータアプリケーション上で効率的に実現される。言い換えると、来歴収集は、インフラストラクチャの構成要素の誤作動や外部攻撃に対してセキュアとなる。

インフラストラクチャ内部攻撃に抵抗するためには、きめの細かい来歴のアクセス制御が望まれる。ビッグデータアプリケーションにおいて、来歴記録は、異なるアプリケーションの来歴データだけでなく、ビッグデータインフラストラクチャ自身の来歴も含む。従って、ビッグデータアプリケーションにおける来歴記録の数は、小容量で静的なデータのアプリケーションよりも、非常に多くなる。これら大容量、複雑で、時に機微な来歴のために、アクセス制御が要求される。そうでなければ、インフラストラクチャ内部攻撃を処理することは不可能であろう。きめの細かいアクセス制御は、ビッグデータアプリケーションの来歴にアクセスするために、異なる権限を異なるロールに対して付与する。同時に、大きな来歴グラフを更新する際、データに依存しない持続性も充足する必要がある。例えば、データの目標物が除去されても、他のデータの目標物の大元として機能するため、その来歴は来歴グラフで保持される必要がある。そうでなければ、来歴グラフは切断されて不完全なものになるであろう。さらに、きめの細かいアクセス制御については、動的で拡張性を有する必要があり、柔軟性のある廃止メカニズムもサポートされる必要がある。

10.4 導入

多くのビッグデータアプリケーションにおいて、来歴は、証明、監査証跡、再現性の保証、信頼性、障害検知のために重要である。ビッグデータアプリケーションの既存のクラウドインフラストラクチャに来歴を追加導入するためには、セキュアな来歴収集と粒度の高いアクセス制御を効率的に処理しなければならない。セキュアな来歴収集を処理するためには、迅速で軽量の認証技術が、既存のクラウドインフラストラクチャにある現行の来歴に統合される必要がある（例．PASOA [33]）。 加えて、エンドツーエンドのセキュリティを達成するために、インフラストラクチャの構成要素間に、セキュアなチャンネルが構築される必要がある。最後に、ビッグデータアプリケーションにおける来歴ストレージとアクセスのセキュリティを実現するために、粒度の高いアクセス制御（例．取消可能な属性ベース暗号（ABE）アクセス制御 [34]）が、現行の履歴ストレージシステム（例．PASS [35]）に統合される必要がある。

結論

ビッグデータが定着しつつある。実際それなしで、データを消費し、新たな形態のデータを生成し、データ主導のアルゴリズムを含む次世代のアプリケーションを想像することはできない。コンピュータ環境がより安価になり、アプリケーション環境がネットワーク化され、システム／分析環境がクラウド上で共有されると共に、システマティックな方法で対応しなければならない脅威として、セキュリティやアクセス制御、圧縮、暗号化、コンプライアンスが挙がっている。

本稿で我々は、ビッグデータの処理およびコンピューティングのインフラストラクチャをよりセキュアなものにするために対処することが必要な、セキュリティ／プライバシーの十大脅威を取り上げてきた。これらビッグデータ特有の上位十項目リストに共通な要素の中には、複数のインフラストラクチャ層（ストレージおよびコンピューティング）の利用、セキュリティ問題の観点から完全に精査されてこなかった NoSQL データベース（ビッグデータの容量により高速のスループットが要求される）など、新たなコンピューターインフラストラクチャの利用、大規模データセット向けの暗号化における拡張性の欠如、小容量データ向けには実用的なリアルタイムモニタリング技術における拡張性の欠如、データを生成するデバイスの多様性、そして、プライバシー／セキュリティを確実なものにするために個別のアプローチにつながる様々な法務／ポリシーの過剰な制限による混乱から生じるものがある。本リストの項目の多くは、このような脅威について分析をすべきビッグデータ処理インフラストラクチャ全体の攻撃対象領域に特有な点を明確化するのに役立つものである。

我々は、本稿が、研究開発コミュニティにおいて、ビッグデータプラットフォームにおけるセキュリティ／プライバシーの拡大の障害となるものに協力して焦点を当てるための行動を促進することを期待する。

参考文献

[1] I.Roy, S.T.Setty, A. Kilzer, V. Shmatikov and E. Witchel, “Airavat: security and privacy for MapReduce” in USENIX conference on Networked systems design and implementation, pp 20-20, 2010.

[2] L.Okman, N.Gal-Oz, Y Gonen, E.Gudes and J.Abramov, “Security Issues in NoSQL Databases” in TrustCom IEEE Conference on International Conference on Trust, Security and Privacy in Computing and Communications, pp 541-547, 2011.

[3] Srini Penchikala, “Virtual Panel: Security Considerations in Accessing NoSQL Databases”, Nov.2011.

[4] B.Sullivan, “NoSQL, But Even Less Security”, 2011.

[5] E.Chickowski, “Does NoSQL Mean No Security?”, Jan.2011.

[6] K.Ding and W.Huang, “NoSQL, No Injection!?” in Def Con 18 Hacking Conference, Jul.-Aug. 2010.

[7] N. Tippenhauer, C. Popper, K. Rasmussen, S. Capkun, “On the requirements for successful GPS spoofing attacks,” in Proceedings of the 18th ACM conference on Computer and communications security, pp. 75-86, Chicago, IL, 2011.

[8] P. Gilbert, J. Jung, K. Lee, H. Qin, D. Sharkey, A. Sheth, L. P. Cox, “YouProve: Authenticity and Fidelity in Mobile Sensing,” ACM SenSys 2011, Seattle, WA, November, 2011.

[9] B. Levine, C. Shields, N. Margolin, “A Survey of Solutions to the Sybil Attack,” Tech report 2006-052, University of Massachusetts Amherst, Amherst, MA, October 2006.

[10] B. Agreiter, M. Hafner, and R. Breu, “A Fair Non-repudiation Service in a Web Service Peer-to-Peer Environment,” Computer Standards & Interfaces, vol 30, no 6, pp. 372-378, August 2008.

[11] A. Anagnostopoulos, M. T. Goodrich, and R. Tamassia, “Persistent Authenticated Dictionaries and Their Applications,” in Proceedings of the 4th International Conference on Information Security, pp. 379-393, October, 2001.

[12] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, “Scalable and efficient provable data possession,” in Proceedings of the 4th international conference on Security and privacy in communication networks (SecureComm ’08), pages 9:1.9:10, New York, NY, USA, 2008.

[13] W. Bagga and R. Molva, “Collusion-Free Policy-Based Encryption,” ISC, volume 4176 of Lecture Notes in Computer Science, pp 233-245. Springer, 2006.

[14] D. Boneh, C. Gentry, and B. Waters, “Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys,” Lecture Notes in Computer Science, 2005.

[15] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” in Proceedings of the 16th ACM conference on Computer and communications security (CCS ’09), pages 213.222, New York, NY, USA, 2009.

[16] J. Feng, Y. Chen, and D. Summerville, “A Fair Multi-Party Non-Repudiation Scheme for Storage Clouds,” the 2011 International Conference on Collaboration Technologies and Systems (CTS 2011), Philadelphia, PA., USA, May 23 - 27, 2011.

[17] J. Feng, Y. Chen, W.-S. Ku, and P. Liu, “Analysis of Integrity Vulnerabilities and a Non-repudiation Protocol for Cloud Data Storage Platforms,” the 2nd International Workshop on Security in Cloud Computing (SCC 2010), in conjunction with ICPP 2010, San Diego, California, USA, Sept. 14, 2010.

[18] J. Feng, Y. Chen, and P. Liu, “Bridging the Missing Link of Cloud Data Storage Security in AWS,” the 7th IEEE Consumer Communications and Networking Conference - Security for CE Communications (CCNC ‘10), Las Vegas, Nevada, USA, January 9 - 12, 2010.

[19] A. Juels and B. S. K. Pors Jr., “Proofs of retrievability for large files,” in Proc. ACM CCS, pages 584.597, 2007.

[20] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable Secure File Sharing on Untrusted Storage,” in USENIX Conference on File and Storage Technologies (FAST), pages 29-42, 2003.

[21] M. Majuntke, D. Dobre, M. Serafini, and N. Suri, “Abortable Fork-Linearizable Storage,” in Proc. of OPODIS, p255-269, 2009.

[22] K. K. Muniswamy-Reddy, P. Macko, and M. Seltzer, “Provenance for the Cloud,” the 8th USENIX Conference on File and Storage Technologies (FAST ’10), Feb. 2010.

[23] J. Onieva, J. Lopez, and J. Zhou, “Secure Multi-Party Non-repudiation Protocols and Applications,” Advances in Information Security Series, Springer, 2009.

[24] R. A. Popa, J. Lorch, D. Molnar, H. J. Wang, and L. Zhuang, “Enabling Security in Cloud Storage SLAs with CloudProof,” Microsoft TechReport MSR-TR-2010-46, May, 2010.

[25] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in INFOCOM, 2010 Proceedings IEEE, pages 1 .9, March 2010.

[26] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, “Enabling public auditability and data dynamics for storage security in cloud computing,” Parallel and Distributed Systems, IEEE Transactions on, 22(5):847 .859, May 2011.

[27] D.Boyd, and K.Crowford, "Critical Questions for Big Data", in information, Communication & Security, 15:5, pp 662-675, May 10, 2012.

[28] C.Duhigg, "How Companies Learn Your Secrets", The New York Times, February 16, 2012.

[29] M.Barbard and T.Zeller, “A Face is Exposed for AOL Searcher No.4417749”, The New York Times, August 9, 2006.

[30] A.Hough, “Google engineer fired for privacy breach after ‘staking and harassing teenagers’”, The Telegraph, Sep 15, 2010.

[31] Electronic Privacy Information Center, “Re-identification”, http://epic.org/privacy/reidentification

[32] R. Lu, X. Lin, X. Liang, and X. Shen, “Secure provenance: the essential of bread and butter of data forensics in cloud computing”, in Proceeding of 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS ’10) pp. 282--292, New York, NY, USA, 2010.

[33] Provenance Aware Service Oriented Architecture (PASOA) Project, http://www.pasoa.org/.

[34] V. Goyal, O. Pandey, A. Sahai, B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data”, ACM Conference on Computer and Communications Security, pp. 89-98, 2006.

[35] K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, “Provenance-aware storage systems”, in USENIX Annual Technical Conference, General Track, USENIX, 2006, pp. 43-56.

[36] C. Percival, “Cache missing for fun and profit”, BSDCan, 2005.

[37] Ac.icmez, Onur, Cetin Koc, and Jean-Pierre Seifert. “Predicting secret keys via branch prediction.” Topics in Cryptology.CT-RSA 2007 (2006): 225-242.

[38] J.Loftus and A.May and N.P.Smart and F.Vercauteren, “On CCA-Secure Fully Homomorphic Encryption”, Cryptology ePrint Archive, Report 2010/560, 2010. http://eprint.iacr.org

[39] R.Canetti, “Universally composable security: A new paradigm for cryptographic protocols. “Foundation of Computer Sciences, 2001. Proceedings, 42nd IEEE Symposium on IEEE, 2001.

[40] D.Boneh, and B.Waters, “Conjunctive, subset, and range queries on encrypted data, “Theory of Cryptography (2007): 535-554.

[41] C.Gentry, “Fully homomorphic encryption using ideal lattices”, Proceedings of the 41st annual ACM symposium on Symposium on theory of computing-STOC’09, ACM Press, 2009.

[42] D.Boneh, E.-J.Goh, and K.Nissim, “Evaluating 2-DNF formulas on ciphertexts”, Theory of Cryptography (2005): 325-341.

[43] D.Boneh, X.Boyen and H.Shacham, “Short group signatures”, Advances in Cryptography-CRYPTO 2004. Springer Berlin/Heidelberg, 2004.

[44] M. van Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, “Fully homomorphic encryption over the integers”. Advances in Cryptology.EUROCRYPT 2010, 24-43.

[45] J.S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, “Fully homomorphic encryption over the integers with shorter public keys”. Advances in Cryptology.CRYPTO 2011, 487-504.

[46] C. Gentry, S. Halevi, N.P. Smart: Homomorphic Evaluation of the AES Circuit. CRYPTO 2012: 850-867

[47] Craig Gentry, Shai Halevi, Nigel P. Smart: Fully Homomorphic Encryption with Polylog Overhead. EUROCRYPT 2012: 465-482.

[48] M. Naehrig, K. Lauter and V. Vaikuntanathan. “Can homomorphic encryption be practical?.” Proceedings of the 3rd ACM workshop on Cloud computing security workshop. ACM, 2011.

[49] D. Boneh and M. Franklin. “Identity-based encryption from the Weil pairing.” SIAM Journal on Computing 32.3 (2003): 586-615.

[50] V. Goyal, O. Pandey, A. Sahai, and B. Waters (2006, October). Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security (pp. 89-98). ACM.

[51] T. Ptacek and T. Newsham. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. Tech. Report. 1998.

[52] M. Barreno, B. Nelson, R. Sears, A. Joseph, J.D. Tygar. “Can Machine Learning be Secure?”. Proc. Of the 2006 ACM Symposium on Information, Computer, and Communications Security, ASIACCS 2006, pp. 16-25, March 21-24, 2006.

[53] W.Diffie, S.Landau, Privacy on the Line: The Politics of Wiretapping and Encryption, The MIT Press, 2010.

[54] C. Gentry, “Computing arbitrary functions of encrypted data,” Communications of the ACM, Vol. 53, No. 3, 2010.

[55] S. Kamara and K. Lauter, “Cryptographic cloud storage,” Financial Cryptography and Data Security, 2010.

[56] http://www.emc.com/leadership/digital-universe/iview/big-data-2020.htm