ブロックチェーン技術を用いた 分散セキュリティログ管理手法の提案

田口 裕介[1]　金井 敦[1]　谷本 茂明[2]

概要：サイバー攻撃を受けた場合、インシデントレスポンスを迅速に行うことで被害の拡大を防ぐことが重要である．しかし，インシデントレスポンスでサイバー攻撃の正確な手口を調査するためには，システムに関わっている機器群のログデータが正しく収集・管理・保全されている必要がある．本研究ではブロックチェーン技術を利用した分散セキュリティログ管理手法を提案する．ブロックチェーン技術の特徴として管理対象データの耐改ざん性を高めることができる．複数の端末で同一のデータを共有することで対象データへのアクセスの可用性を高めることができるという 2 点が挙げられる．管理対象データの耐改ざん性を高めることで，インシデントレスポンスの際に完全性が保証されたログデータを解析することができる．さらに，多くの人が簡単にアクセスすることができるようになり，必要なときに必要なデータを参照することで解析の効率を上げることができる．提案手法のプロトタイプを実装し，データの収容と参照におけるスループットを測定した．測定結果を評価し実用の可能性を示した．

キーワード：ブロックチェーン，インシデントレスポンス，デジタルフォレンジック，ログ管理

Distributed security log management method using blockchain scheme

Yusuke Taguchi[1]　Atsushi Kanai[1]　Shigeaki Tanimoto[2]

Abstract: In the case of a cyber attack, it is important to prevent the spread of damage by quickly incident response. In order to investigate an accurate situation of cyber attack, it is necessary to collect, manage and preserve log data of devices related to the system. In this research, a distributed security log management method using blockchain scheme is proposed. Blockchain scheme has following two features. Tamper resistance of managed data can be improved. By sharing the same data with multiple terminals, the availability of access to the data become increase.　In this paper, feasibility of proposed method is evaluated by using prototype.

Keywords: Blockchain, Incident Response, Digital Forensic, Log Management

1. はじめに

IT 技術の発展により企業では IT を導入し，データをデジタルで管理することが主流となってきている．それに伴い企業のシステムやネットワークに不正にアクセスし情報の窃取することを目的としたサイバー攻撃もまた増加傾向にある．警視庁が公開している「平成 30 年におけるサイバー空間をめぐる脅威の情勢等について」[1]によると，情報窃取を目的とした標的型メール攻撃の観測件数が平成 26 年から平成 30 年にかけて増加し続けており，平成 30 年時点では 6740 件が観測されている．このことから企業が管理しているデータを狙ったサイバー攻撃が増えていることが分かり，企業ではサイバー攻撃による被害を防ぐためのセキュリティ対策がより重要となってきている．

サイバー攻撃の対策として CSIRT のようなセキュリティ対策チームを設置する企業が増えてきている．CSIRT がセキュリティ設計やインシデントレスポンスを行うことでサイバー攻撃による被害のリスクを減らすことができる．JPCERT/CC が公開している「2017 年度 CSIRT 構築および運用における実態調査」[2]によると，日本シーサート協議会[3]加盟組織数が前回調査時（2015 年）の 66 組織から 187 組織へと増加している．このことからサイバー攻撃に対策するために企業に CSIRT のようなチームを設置することが重要視されてきていることが分かる．

実際にセキュリティインシデントが発生した場合 CSIRT が迅速にインシデントレスポンスを行いインシデントの詳細や被害状況を調査し，発生したインシデントに対応することで被害を最小限に抑えることが重要である．しかし，迅速に正確なインシデントレスポンスを行うためには組織内で運用しているシステムに関わっているあらゆる機器のログデータが正しく収集・管理・保全されている必要がある．ログデータが収集されていない場合，インシデントレスポンスの際に必要となるログデータを参照することができずインシデントの詳細を調べることができなくなる可能

1 法政大学　大学院理工学研究科
Hosei University, Graduate School of Science and Engineering
2 千葉工業大学　社会システム科学部
Chiba Institute of Technology, Faculty of Social Systems Science

性がある．また，ログデータの条件を詳しく設定し参照できるように管理することで，より迅速なインシデントレスポンスを可能にすることができる．そして，ログデータの保全も重要である．ログデータの完全性を保証された状態で保管しておくことでログデータの改ざんの可能性を考慮する必要が無くなり，完全性が保証されたログデータを用いてインシデントレスポンスを行うことができる．

本研究ではインシデントレスポンスの際に効率的な解析を行うことができるように上記で述べた機能を満たし，インシデントレスポンスを補助するためのログ管理手法の実現を目的とする．

2. 研究の背景

既存のログデータの保全の手法としては，改ざん対策ソフトウェアで監視を行うといった手法がある．この手法では監視対象データのスナップショットを保持し，スナップショットと比較し改ざん検知を行う．しかし，常に監視対象データとその対となるスナップショットを保持し続ける必要があり，単純にログを管理する機器が扱うデータが増えてしまう．また，ログデータを保存する際にログ管理用機器へ集約保存する場合，その管理用機器に障害が発生した際にログデータへのアクセスができなくなってしまい，必要な時に必要なデータを用意できない可能性がある．これらの既存のログ管理手法の課題を解決するためにログ管理手法にブロックチェーン技術を利用することを提案する．

本章ではブロックチェーン技術とログ管理手法の親和性の高さと組み合わせた際に得られる効果について説明する．

2.1 ブロックチェーン技術

ブロックチェーン技術とはビットコインを実装するために考案された技術である[4]．ブロックチェーン技術の主な特徴として以下の 2 つが挙げられる．

(1) ハッシュチェーン

ブロックチェーン技術の特徴の 1 つはハッシュチェーンを構成しデータの管理を行うことである．ブロックチェーン技術では対象データといくつかの情報を格納したブロックと呼ばれる単位でデータの管理を行う．ブロックに「ブロックハッシュ」と「1 つ前のブロックハッシュ」をという情報を格納することでハッシュチェーンを構成する．ブロックハッシュとはブロックに格納されている情報全てを用いて生成したハッシュ値であり，このブロックハッシュを生成する際に 1 つ前のブロックのブロックハッシュを含ませることでハッシュチェーンが構成される．ハッシュチェーンを構成することで，もし管理しているデータが改ざんされた場合でも各ブロックの「1 つ前のブロックハッシュ」と「ブロックハッシュ」とを比較することでどこのブロックで改ざんされたかを容易に検知することができ，対改ざん性の高いデータ管理を行える．ハッシュチェーン概要図を図 1 に示す．

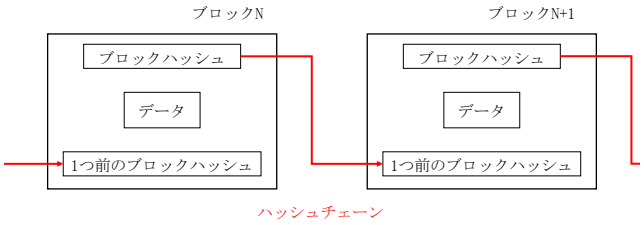

図 1　ハッシュチェーン概要図

(2) P2P ネットワーク

もう 1 つの特徴は P2P ネットワークを構成しデータの管理を行うことである．ブロックチェーン技術では複数のノードを用いてデータの管理を行う．ノード同士で P2P ネットワークを構築し，同一のブロックチェーンをそれぞれのノードが保持する．ブロックチェーン技術ではブロックチェーンを構成する P2P ネットワーク（以降ブロックチェーンネットワークと表記する．）に所属する全てのノードが常に最新のブロックチェーンを保持するため，単一のノードがダウンしたとしても全てのノードがダウンしなければブロックチェーンにアクセスすることができ，管理しているデータへのアクセスの可用性を高めることができる．ブロックチェーンネットワーク概要図を図 2 に示す．

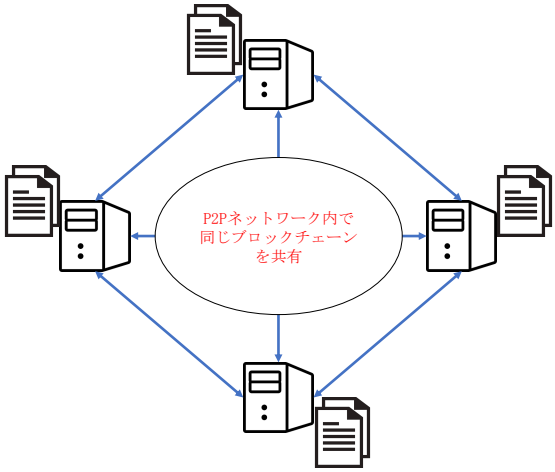

図 2　ブロックチェーンネットワーク概要図

2.2 ログ管理手法におけるブロックチェーン技術

2.1.(1)で述べたハッシュチェーンを構成しデータの管理を行うことで，ブロックチェーンのハッシュチェーンを検証することでデータの改ざんの有無を容易に検知することができる．このハッシュチェーン構成とハッシュチェーン

検証によりログデータの完全性の証明を容易に行うことができ，検証用にログデータを二重に保管する必要がなくなり効率的なログデータ保全を行える．

2.1.(2)で述べた P2P ネットワークを構築し複数のノードで同一のデータを共有管理することで，1 つのノードで障害が発生したとしても別のノードを用いてログデータにアクセスすることができる．これによりログデータへのアクセスの可用性を高める事ができ，必要な時に必要なログデータを参照することが可能になる．また，1 つのノードでデータの欠損や改ざんが発生した場合でも別ノードが保有するデータを用いることで，データの復元を容易に行うことが可能である．

上記のようにブロックチェーン技術とログ管理手法は親和性が高く，ログ管理手法にブロックチェーン技術を適用することで，インシデントレスポンスの際に信頼できるログデータを必要な時に呼び出せるログ管理手法を実現することができると考えられる．

2.3 関連研究

ブロックチェーン技術とログ管理手法の関連研究として「 BLOCK CHAIN BASED DATA LOGGING AND INTEGRITY MANAGEMENT SYSTEM FOR CLOUD FORENSICS」[5]がある．この論文では各クラウドサービスプロバイダがノードを用意し，クラウド上のインスタンスのログデータをブロックチェーンで管理・保全する手法を提案している．この論文では各クラウドサービスプロバイダが協力し，パブリックなクラウド空間のインスタンスのログデータの管理手法を提案しているが，本論文では，単一企業内のプライベートな空間で利用でき，設計を柔軟に行えるログ管理手法を提案する．

3. 提案手法

3.1 ブロックチェーンの種類

ブロックチェーンには 3 種類の構成がある．ブロックチェーンへのアクセス権限のポリシー設計によりそれぞれ「パブリック型」,「プライベート型」,「コンソーシアム型」に分けることができる[6]．ブロックチェーンのポリシーと種類について表 1 に示す．

表 1　ブロックチェーンのポリシーと種類

種類	アクセス権限
パブリック型	全てのノードが所有
プライベート型	管理者ノードのみが所有
コンソーシアム型	認証局により承認を得た全てのノードが所有

本論文では単一企業内のプライベートな空間での利用を想定している．なので，提案手法にブロックチェーン技術を適応させる場合，ログデータは外部へ公開されるべきではないのでプライベート型かコンソーシアム型のどちらかを構成するべきである．そして管理データの保全をより重視した場合，単一のノードのみに権限を与えるより複数のノードに権限を与え管理すべきだと考えられる．したがって本論文の提案手法ではコンソーシアム型のブロックチェーンを構成する．

提案手法の実装にあたり，OSS のコンソーシアム型ブロックチェーンアプリケーション開発用フレームワークである「Hyperledger Fabric」[7]を利用する．Hyperledger Fabric はコンソーシアム型ブロックチェーンを構成する際に必要となる認証局やブロックチェーン生成時の合意形成などの処理を行うモジュールを提供している．

3.2 提案手法概要

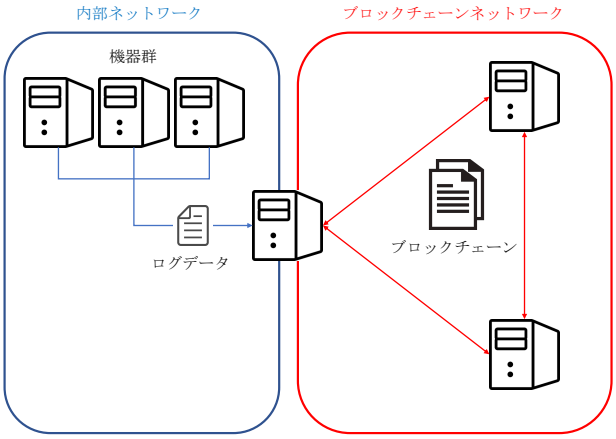

図 3　提案手法概要図

提案手法概要図を図 3 に示す．会社内ネットワークのようなプライベートな空間である内部ネットワークに属する機器群のログデータをブロックチェーンネットワークへのアクセス権限を持つノードへ集約する．そして一定周期でブロックチェーンネットワークのブロックチェーンへのブロック追加のリクエストを行う．ブロック生成後，ブロックチェーンネットワークの合意形成のポリシーに従い合意形成を行い，合意を得られたときのみブロックチェーンに生成したブロックを追加する．この一連の流れで収集したログデータをブロックに格納しブロックチェーンに追加していくことでブロックチェーンによるログデータの管理を実現する．

ブロックチェーン技術ではブロックチェーンネットワークに複数のノードを用意する必要がある．提案手法においてはログデータの保全を重視し，ブロックチェーンネットワークに参加するノードは複数のグループで管理する．1 つのグループで全てのノードを管理する場合，中央集権型のログ管理手法と同じになってしまいブロックチェーン

技術の強みである複数人での共有管理という特徴が損なわれてしまう.例えば,会社内の部署ごとにノードを用意し,異なるグループがそれぞれのノードを管理するといった体系をとることでセキュリティ管理の部署のみが管理するのではなく，社内の全ての部署で管理するより信頼のおけるログ管理を実現できる.

ブロックチェーンを呼び出し参照する際にはより細かな条件を設定し，ブロックチェーンの呼び出しを行える専用のクライアントアプリケーションを用いてデータを呼び出すことを想定している.

3.3 ブロックチェーンネットワーク

ブロックチェーンネットワークに所属するノードはブロックチェーンの生成・追加の処理を行う．ブロックチェーンへのブロックを追加するためにはブロックチェーンネットワークに参加している各ノードからの合意を得る必要があるため（合意形成のポリシーによって必要な合意数を変更できる.)，もし悪意のあるノードがネットワークに参加したとしても，ブロックチェーンを不正に操作することは難しい．ブロックチェーンネットワーク内でのブロック生成・追加フローを図 4 に示す.

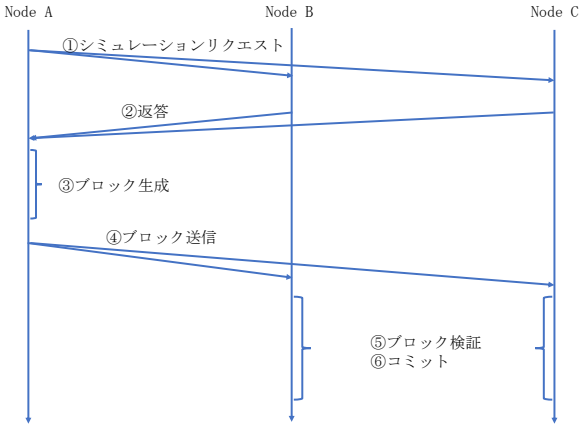

図 4　ブロック生成・追加フロー

図 4 のブロック生成・追加フローの各フローについて説明する.

① ブロックを生成したいノードがブロックチェーンネットワークに所属する全てのノードに，ブロックに格納したいデータの内容とシミュレーションリクエストを送る.
② リクエストを受けたノードは，送られたデータやリクエストを送ったノードの権限などをシミュレーションし，返答を送る.
③ ブロックを生成するのに十分な返答を受けた場合，ブロックを生成する.
④ 生成したブロックをブロックチェーンネットワークに所属する全てのノードに送る.
⑤ 送られたブロックデータに問題が無いか検証をする.
⑥ 問題が無かった場合，送られたブロックデータをブロックチェーンへと追加する.

以上のようなフローをブロックチェーンネットワーク内で行いブロックを生成・追加する.

3.4 ブロックチェーンの分別

Hyperledger Fabric の機能の 1 つにチャネルがある．チャネルを使うことで,3.3 節で述べたブロックチェーンネットワークを仮想化することができる．ブロックチェーンネットワークを仮想化することでチャネルごとに異なるブロックチェーンを生成することができる．チャネルを複数構築することで管理データの種類ごとにブロックチェーンを生成でき，データの分別を行うことができる．本論文で管理対象としているログデータにはプロキシサーバのログや DNS サーバのログなど，様々な種類のログデータが存在する．これらのログデータを全て 1 つのブロックチェーンで管理すると，必要なログデータを様々なログデータが含まれているブロックチェーンから探さなければならなくなってしまう．この場合インシデントレスポンスの際に必要なデータを素早く用意することができず，時間を浪費してしまうと考えられる．したがって，管理するログデータの種類ごとにチャネルを構築し,ブロックチェーンを分別する.この場合，インシデントレスポンスの際に必要な種類のログデータを管理しているチャネルにアクセスすることで，素早く必要なデータを用意することが可能になり解析の効率を上げることができる.

チャネルは単にブロックチェーンネットワークを仮想化するだけではなく，チャネルに参加するノードの設定もチャネル毎に行うことができ，この機能を利用することでより柔軟なブロックチェーンネットワークの設計を行うことができる．例えば，社内システムの根幹に関わるような機器群のログデータは全ての部署がチャネルに参加し管理すべきだが，幾つかの部署のみに影響を与えるような機器群のログデータを管理する場合は，関係する部署とセキュリティ管理をしている部署のみが参加するチャネルを構築するといった設計をすることが可能である．ブロックチェーンとチャネルの概要図を図 5 に示す.

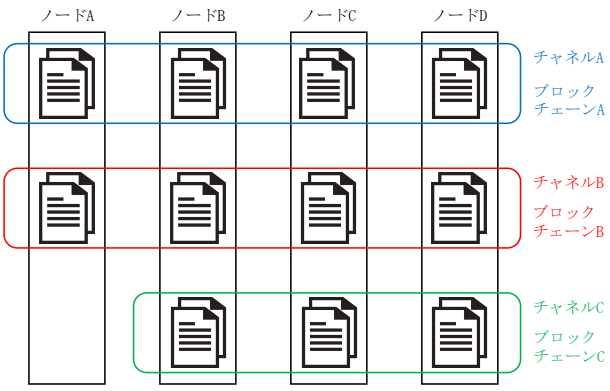

図 5　ブロックチェーンとチャネル

4. 実装と測定

ブロックチェーン技術を利用して提案したログ管理手法を実現することが可能か検証するために，Hyperledger Fabric を用いて対象データのブロックチェーンへの追加・呼び出しといった最低限の機能を持つプロトタイプを実装し，そのスループットを測定した．実装にあたって OSS の Hyperledger Fabric v1.4 を用いてプロトタイプを実装した．評価のために構成した環境の概要を図 6 に示す．

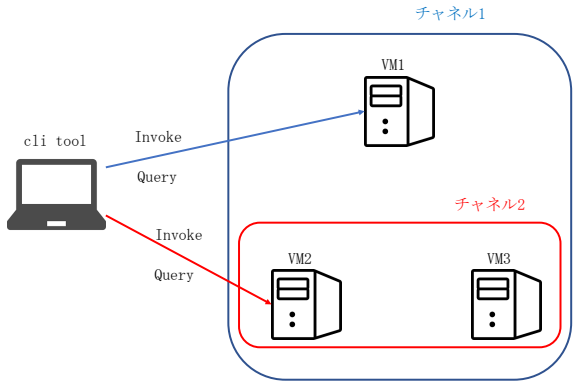

図 6　評価用環境の概要図

実験用サーバ内に KVM を用いて VM を 3 台用意し，VM3 台でブロックチェーンネットワークを構築した．また，全ての VM の OS には Ubuntu 18.04 を利用した．チャネル機能の検証のために 2 つのチャネルを構築しチャネル 1 には全ての VM を参加させ，チャネル 2 には VM2 と VM3 のみを参加させた．ブロックチェーンの操作や呼び出しには cli tool を用いて行った．Hyperledger Fabric ではブロックチェーンに操作を加える場合は Invoke メソッドを使い，ブロックチェーンの操作トランザクションを発生させる．ブロックチェーンの呼び出しの際には Query メソッドを使い呼び出す．測定時には cli tool から Invoke メソッドと Query メソッドを 100 回ループさせ，ブロックチェーンの操作と呼び出しの処理にかかった時間を測定し，Hyperledger Fabric による Invoke 処理と Query 処理のスループット（実行から処理が完了するまでにかかる平均時間[s]）を計算した．また，Invoke 処理をループしブロックチェーン長を長くしていった際のそれぞれの処理速度の変化も観測した．

実際のインシデントレスポンス時には CSIRT のメンバー複数人が利用し，さらに CSIRT 以外の部署での通常利用も継続されると考えられるので，その場合の処理能力を評価するために Query 処理を並列で実行し，その時のスループットの変化を観測した．

参加ノードが異なるチャネルを複数構築し，それぞれのノードからアクセス可能なブロックチェーンを確認することでチャネル機能が正しく働くことと柔軟なブロックチェーンネットワークの構築が可能であるか検証した．また，合意形成ポリシーを編集し必要な合意条件を変更した場合のブロックチェーンの操作を観測した．

5. 結果と評価

4 章で測定した結果を評価する．測定の結果得られた Invoke 処理と Query 処理のスループットをそれぞれ表 2 に示す．

表 2　Invoke 処理と Query 処理のスループット

Invoke 処理	Query 処理
0.98[s]	0.056[s]

表 2 より Invoke 処理と Query 処理のそれぞれのスループットが 0.98[s]と 0.056[s]であることが分かる．また，ここでの Invoke 処理のスループットは Invoke を実行してから実際にブロックチェーンが更新されるまでの時間である．この結果から Invoke 処理の方が Query 処理よりスループットが遅いということが分かる．これはブロックチェーンデータを呼び出しているだけの Query メソッドよりもブロックチェーンデータを操作する分だけ処理工程が増えているので Invoke 処理の方がややスループットが遅くなったと考えられる．また，ブロックチェーン長を 400 まで伸ばしたが各処理のスループットが変化することは無かった．ブロックチェーン長に関わらずスループットを維持することができたため，常に安定したブロックチェーン呼び出しが行えると考えられる．

ブロックチェーンを操作する Invoke 処理は，提案手法では一定周期にログデータをブロックチェーンに追加する時のみに実行することを想定している．0.98[s]は周期調整を行うことで十分利用できるスループットだと考えられる．

Invoke 処理と異なり Query 処理はブロックチェーンを呼

び出す処理であり直接利用する人間に関わってくる処理なので重要である．Alibaba Cloud が公表している調査によると[8]，シェアの高い OSS のログ管理ソフトウェアである「Elasticsearch」[9]の単純なクエリを実行した際の処理時間が 0.097[s]である．この結果と比較すると表 2 の Query 処理時間 0.056[s]はログ管理手法において十分利用できるスループットを満たしていると考えられる．また，Query 処理を並列実行した際のスループットの変化を図 7 に示す．

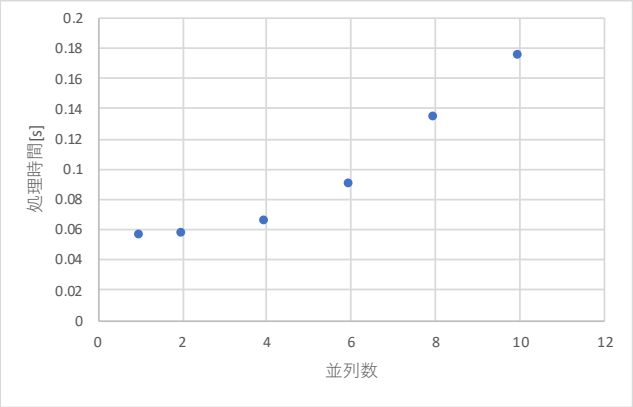

図 7　Query 処理の並列数と処理時間

図 7 の Query 処理の並列数と処理時間より並列数が増えていくにつれて処理時間の増加も大きくなっている事がわかる．今回の測定環境では Query 処理用の窓口が 1 つのみだったため並列数が増えるにつれて処理時間の増加も大きくなったと考えられる．この結果から実環境で利用する場合は Query 処理の窓口を増やし負荷分散する必要があると考えられる．

VM1 と VM2 にチャネル 1 のブロックチェーンとチャネル 2 のブロックチェーンへの Invoke 処理と Query 処理をそれぞれ実行した結果 VM2 は問題なく全ての処理が完了した．しかし，VM1 ではチャネル 1 のブロックチェーンへの処理は正常に完了したが，チャネル 2 のブロックチェーンへの処理ではエラーを発生した．このことから Hyperledger Fabric のチャネルの機能を活用し想定した通りのブロックチェーンネットワークを構築することができたと考えられる．また，チャネル 1 の合意形成ポリシーを編集し合意条件をチャネルに参加するいずれかのノードからの合意を得ることに変更し，意図的に VM3 の機能を停止した状態で VM1 にチャネル 1 のブロックチェーンの Invoke 処理と Query 処理を実行した．その結果 VM1 が実行した処理は両方正常に完了することが観測できた．このことから合意形成ポリシーが正しくチャネルに反映されたと考えられる．これらのことからチャネル設定とポリシー設定を状況に応じて使い分けることで，提案手法概要で述べた柔軟なブロックチェーンネットワークの設計を実現できると考えられる．また，VM3 を停止した際に正常にデータ呼び出しを行えたことから，データへのアクセスの可用性を高めることができたと考えられる．

6. 結論と今後の課題

6.1 結論

本論文でブロックチェーン技術を用いたログ管理手法の提案し，提案手法のプロトタイプの測定結果から十分なスループットを持つことを示すことができた．また，実社会で利用する際に役に立つと考えられるブロックチェーンネットワークの設計の柔軟性とログ管理にとって重要である可用性の向上も示すことができた．しかし，実用のためにはいくつかの課題があることが分かった．課題の解決策も含め，提案手法の実用の可能性を示すことができた．

6.2 今後の課題

本論文ではブロックチェーン技術とログ管理手法との相性の良さを主張し，プロトタイプを実装・評価し提案手法の可能性を示した．しかし，プロトタイプではデータを格納・管理するための最低限の機能のみ実装されており提案手法で上げる要件を満たすためにさらなる機能の実装，機能測定と評価をする必要がある．

6.2.1 専用クライアントアプリケーション実装

本論文では単純に Invoke 処理と Query 処理を行うためだけの cli tool を測定で用いたが，この機能だけではただブロックチェーン単位でのみデータを呼び出すことしかできず，インシデントレスポンスの効率向上には役立たないと考えられる．必要なデータの条件を細かく指定し呼び出すことができるクライントアプリケーションを実装する必要がある．また，ブロックチェーンによってデータの完全性が保証されるので管理者以外のユーザにも安全にデータの公開ができると考えられる．その場合，クライアントアプリケーションでユーザに「一般」や「管理者」といった属性を付与しデータの公開範囲を細かく指定する必要がある．

6.2.2 インシデント発生時想定のセキュリティ検証

提案手法のインシデント発生時に目的のインシデントレスポンスの補助を行えるかを評価するために，実例などを参考にし，考えられるサイバー攻撃を受けた際のシミュレーションを行う必要がある．また，シミュレーションの結果から考えられる提案手法の脆弱性を挙げ，機能改善が必要になる可能性があると考えられる．

6.2.3 実環境に近い状態での測定

提案手法を実環境で利用することを想定した場合，大量のログデータを長期間管理する必要がある．そういった状況を再現しログデータの容量やブロックチェーンの長さな

どが提案手法のスループットにどのような影響を与えるかを観測し，ノードの数やスペック，ブロックチェーンの転換期等の最適な値を考察する必要がある．

7. おわりに

本研究では，サイバー攻撃による被害の増加とともに注目度が高まってきている CSIRT が万全の状態でインシデントレスポンスに取り掛かれるように，ログが保全された状態で保管されるログ管理手法の提案と実装を目的としている．本論文を通してブロックチェーン技術とログ管理との親和性の高さを再確認し，新たな機能実装の可能性を見出すことができた．また，現状考えられる今後の課題を整理することで，これから本研究が解決すべき問題を明確に意識することができた．今後，本論文で得られた発見を手法に取り入れ，明確になった問題を解決し，よりインシデントレスポンス補助の要件を満たすログ管理手法の提案と実装を目標とする．

参考文献

[1] 警視庁．「平成 30 年におけるサイバー空間をめぐる脅威の情勢等について」．https://www.npa.go.jp/publications/statistics/cybersecurity/data/H30_cyber_jousei.pdf.（参照　2019-08-04）．

[2] JPCERT/CC．「2017 年度 CSIRT 構築および運用における実態調査」．https://www.jpcert.or.jp/research/20181218_CSIRT-survey2017.pdf.（参照　2019-08-04）．

[3] 日本シーサート協議会．https://www.nca.gr.jp/.（参照　2019-08-04）．

[4] Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. https://bitcoin.org/bitcoin.pdf.（参照　2019-08-07）．

[5] Jun Hak Park, Jun Young Park, Eui Nam Huh. “BLOCK CHAIN BASED DATA LOGGING AND INTEGRITY MANAGEMENT SYSTEM FOR CLOUD FORENSICS”. 2017

[6] KPMG．「ブロックチェーンの全体像～利用目的別に 3 種類」．https://home.kpmg/jp/ja/home/insights/2019/01/blockchain-fintech-system.html.（参照　2019-08-09）．

[7] Hyperledger．https://www.hyperledger.org/projects/fabric.（参照　2019-08-14）．

[8] Alibaba Cloud．https://jp.alibabacloud.com/help/doc-detail/59070.htm.（参照　2019-08-19）．

[9] Elasticsearch．https://www.elastic.co/jp/products/elasticsearch.（参照　2019-08-19）．