ブロックチェーンの信頼性を向上する脅威分析手法およびスマートコントラクト検証

Threat Analysis Method and Smart Contract Verification to Improve Reliability of Blockchain

小櫻　文彦　　藤本　真吾　　野村　佳秀　　山下　一寛

あらまし

ブロックチェーンは，ビットコインの基盤技術として考案された技術である。非中央集権で信頼性を担保できるため，仮想通貨以外にも不動産やヘルスケアといった様々な分野への応用が期待されている。しかし，システム全体を構成する要素間の境界部分や，サブシステムとして導入されることの多いスマートコントラクトに問題があると，ブロックチェーンで管理している仮想通貨が盗まれるなど，ビジネス上重大な損失に直結する恐れがある。そのため，アプリケーションを含めてシステム全体として信頼性の向上が課題になっている。今回，富士通研究所は，ブロックチェーンシステムの構築・運用の際に問題点をチェックする脅威分析手法，およびスマートコントラクト内の脅威を静的解析技術によって網羅的に検出するスマートコントラクト検証技術を開発した。これによって，ブロックチェーン開発者は，より安全性の高いブロックチェーンを使ったシステムを迅速に開発できるようになる。

本稿では，開発した脅威分析手法およびスマートコントラクト検証技術について述べる。

Abstract

Blockchain is a technology devised as a fundamental technology of Bitcoin. It allows reliability to be ensured in a decentralized manner, and expectations are high for its application in various areas beyond virtual currency, such as real estate and healthcare. However, any problem in a boundary between elements that constitute an entire system or in a smart contract, often introduced as a subsystem, may lead directly to significant losses in business such as the theft of the virtual currency managed by a blockchain. Accordingly, improving the reliability of the system as a whole, including applications, is an issue. Fujitsu Laboratories has developed a threat analysis method, which checks a blockchain system for any problems at the time of its construction and operation, and a smart contract verification technology that exhaustively detects threats in smart contracts by using static analysis technology. This will enable blockchain developers to quickly develop systems that use blockchains with greater security. This paper describes the threat analysis method and smart contract verification technology that we have developed.

1. まえがき

ブロックチェーンは，ビットコインの基盤技術として考案された。これは，取引をまとめたブロックの間をハッシュ値と電子署名によってチェーンのようにつないでいく技術である。これによって，非中央集権の環境でも取引履歴を改ざんしにくい性質を実現し，仮想通貨のような信頼性が必要な取引を可能としている。また，電子署名を応用した信頼性の高いデジタル資産管理システムにも活用され，金融以外にも不動産やヘルスケアといった様々な分野への応用が期待されている。

ブロックチェーンは，もともと仕組みが単純で耐故障性の高いP2P（Peer to Peer）ネットワークによって全ての取引記録を共有しているため，様々な攻撃にも耐えてきた実績がある。しかし，取引情報を安全に管理する機能しか持たず，金融以外の分野への応用には不十分である。そのため，Webフロントシステムや，ブロックチェーン内に定義した取引を厳格なルールに沿って自動処理するプログラムである「スマートコントラクト」などのサブシステムを介して利用されることが多い。しかし，サブシステムとの境界部分には，利便性の向上と引き換えに二重実行やなりすましにつながる新たな脅威が発生する可能性がある。そのため，システム設計を誤ると，ビジネス上深刻な損失に直結する恐れがある。

ブロックチェーンは，その運用形態によって大きくパブリック型，コンソーシアム型，プライベート型の3種類に分類される。これらの違いは，管理者の役割で説明される。パブリック型のブロックチェーンは管理者不在でありながら，ハッシュ値計算の競争を行うマイニング処理によってシステムの安全性を担保している。一方，コンソーシアム型およびプライベート型には，システムを構築・運用する参加者を許可する管理者が存在する。

今回富士通研究所は，システムの問題点をチェックする脅威分析手法，および静的解析技術を利用して，リスクを網羅的に検出するスマートコントラクト検証技術を開発した。これらの技術は，限られた組織での使用を前提としたコンソーシアム型，プライベート型のブロックチェーン実行基盤であるHyperledger Fabric(以下, Fabric)[(1)] を用いている。これによってブロックチェーン開発者は，より安全性の高いブロックチェーンを使ったシステムを迅速に開発できるようになる。

本稿では，開発した脅威分析手法およびスマートコントラクト検証技術について述べる。

2. ブロックチェーンシステムを構築する際の課題

本章では，ブロックチェーンシステムを構築する際の課題を，大きく二つに分けて説明する。一つは，システム全体を構成する要素間の境界部分の設計に関する課題，もう一つはスマートコントラクトと呼ばれるブロックチェーン上の取引の自動処理プログラムに関する課題である（図-1）。

脅威分析を行うためには対象を決めて取り組む必要があるため，今回は具体例として，仮想通貨取引をスマートコントラクトで定義したサービスを挙げて，課題を掘り下げる。

2.1 境界部分の設計における課題

パブリック型のブロックチェーンのビットコインでは，UTXO（Unspent Transaction Output：未使用の資産管理情報）と呼ばれる手法によって二重使用を防ぐことで，仮想通貨取引の正当性が検証されている。

一方，FabricやEthereum(注)などのスマートコントラクトによる残高管理では管理者による干渉を受けないため，スマートコントラクトの実行ロジックによって取引の正当性を検証する。しかし，呼び出す側に不備があると，実行ロジックが正しい場合であっても同じ取引が複数回実行されるなど，取引の正当性が担保できなくなる。このような，実行ロジックとそれを呼び出す側の間など信頼性が異なる境界を，信頼境界と呼ぶ。正当性担保の観点から，各信頼境界での設計には注意が必要である。しかし，その重要性が浸透していないため，設計上の不

（注） スマートコントラクトを動かすためのプラットフォームとして，オープンソースで開発が進んでいるプロジェクト。

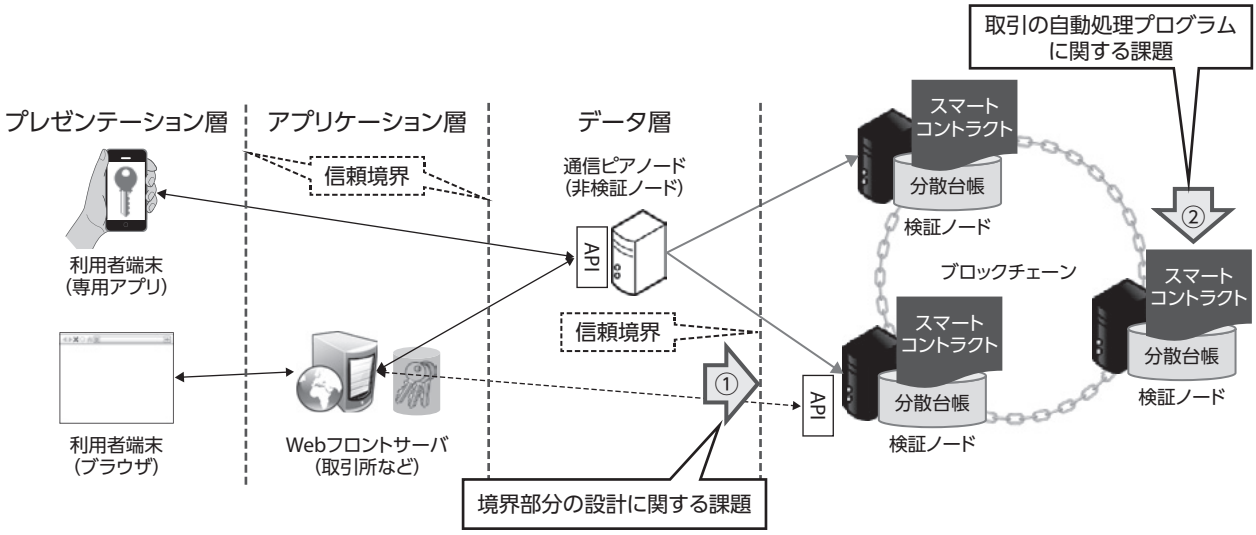


図-1　ブロックチェーンシステムの全体構成と課題点

備が発生しデータが改ざんされるなど，脅威になる可能性が高い。

スマートコントラクトを使用したブロックチェーンシステムは，一つの機能をシステム全体で実現することが多い。そのため，信頼境界において脆弱性が発生する可能性がある。安全性を担保するためには，システム全体の脅威分析を行い，信頼境界で脆弱性が発生しないように適切な対策を取る必要がある。しかし，ブロックチェーンそのものが最先端の技術であるため，まだ事例が少なく，ブロックチェーンを使用したシステムを対象とした脅威分析手法の確立が課題となっている。

2.2 スマートコントラクト開発における課題

前節でも述べたとおり，Fabricではスマートコントラクトを介して資産を扱う。そのため，スマートコントラクトに欠陥が存在すると，誤った条件で取引が行われるなど問題となる。例えば，2016年に発生したThe DAO事件では，十分にレビューされたスマートコントラクトにもかかわらず，適切な関数の使用や，適切な処理順序での関数呼び出しではなかったために，本来一度しか呼び出せないはずの関数を複数回呼び出すことが可能になっていた。その結果，この欠陥を利用した不正操作によって，当時の価値として約65億円の仮想通貨が不正に移動された。

そのため，今後スマートコントラクトの品質向上の取り組みは不可欠である。現在，主に取り組まれているのは，ブロックチェーンやスマートコントラクトの専門家がレビューすることで，問題が起こらないかどうかを確認する方法である。専門家によるレビューは，効果も高く有効である。しかし，専門家の人手によるレビューではコストがかかり，また大量の作業を行うには限界があるため，チェックミスが発生する可能性が高まることが問題であった。

3. ブロックチェーンを使用したシステムの脅威分析手法

本章では，2.1節で述べた設計の課題を解決するために今回開発した，ブロックチェーンを使用したシステムに対する脅威分析手法について説明する。

3.1 脅威分析手法

システム全体のリスク分析は，IPA（独立行政法人情報処理推進機構）が公開している制御システム向けの分析手法「制御システムのセキュリティリスク分析ガイド」（以下，IPAガイド）[(2)] を参考に，表-1に示した五つのステップで実施した。IPAガイドは，重要度の高いリスクへの対処を目的とした分

表-1　IPAガイドを基に実施したリスク分析のステップ

順番	分析内容	詳細
1	システム構成の明確化	リスク分析の対象を明確化し，システム全体の仕組みを理解できるようにする
2	ユースケースの明確化	ユースケースごとに信頼境界を境にしたデータの流れを明らかにする
3	資産の重要度の決定	各資産のリスクを数値化する
4	事業被害とそのレベルの定義	対策が必要なセキュリティ要件を求める
5	リスク分析	セキュリティ要件を元にリスク分析を行い，確認が必要な項目を作成する

析手法である。

今回のリスク分析の目的は，システム全体の分析ではなく，ブロックチェーン開発者がブロックチェーンに関わる部分の脅威抽出である。そこで，我々は検証方法を工夫することで，負担軽減しても十分に品質の確保が可能であると考えた。具体的には，ステップ5を中心にステップ3，ステップ4の分析手法を変更し，ステップ5の「リスク分析」において，トップダウンでブロックチェーンに関わる部分のみを分析することで，負担軽減を行った。

一般に，ブロックチェーンを使用するシステムは複数の階層構造から成る。ブロックチェーンは下位層に当たり，上位層でブロックチェーンを使用する機能が限定されれば，ブロックチェーンで使用する機能が削減される。そのため，ブロックチェーンの全ての機能を網羅的に分析する必要はなく，分析対象を絞ることが可能である。またトップダウンの分析では，事例の細分化の過程で同様の事例が発生する場合がある。これらは，同一の事例として扱えるかを検証することで，事例の重複を抑えることが可能である。これらの特性を活用し，分析の負担を削減しながら，トータルでボトムアップと同等な精度の分析とした。

また，ステップ3において重要度の数値化を省略し，ステップ4において重要度を基にリスク値を求める際に，簡易的な手法でリスク値を求める。その結果，IPAガイドと比べてブロックチェーンに関する詳細な分析は必要になるが，リスク評価の負担をブロックチェーンに限定しながら，それに関わる脅威抽出に注力できるようになった。

IPAガイドをベースとして，ブロックチェーンの脅威抽出に特化することで，効率的に脅威分析を行うことができた。併せて，ブロックチェーン上で発生する脅威を，一部の対策を含めて体系化した。この体系化によって，ブロックチェーンに特化した脅威分析をブロックチェーンシステムの開発者が効率良く行えることを確認した。

3. 2　脅威分析結果

本脅威分析を実際のブロックチェーンのシステムに適用することで，効率的に脅威を分析できた。ここでは，分析の結果抽出した脅威の一部およびそれを防ぐ方法を紹介する。

(1) 振込の二重実行

今回の分析対象であるFabricでは，同じ依頼を二重実行させないために，同じトランザクションを複数回実行できない仕組みが実装されている。一般的には，再送攻撃のように同様の依頼がブロックチェーンに対して繰り返し実行された場合は，識別子が異なるトランザクションが発生する。例えば，ユーザーが誤って2回振込を依頼した場合，別々のトランザクションが発生するため，2回振り込んだことになる。

この問題を防ぐためには，複数の方法が考えられている。一つは，振込処理を行うスマートコントラクトのインターフェースに，二重振込ができないインターフェースを導入することである。もう一つは，アプリケーションプログラム上にユーザーが誤って2回振込を依頼することを防止する機能が備わっているかを確認することである。更には，スマートコントラクトのインターフェースや処理内容と上位アプリケーションの処理内容をすり合わせながら，スマートコントラクトで二重で振り込まれないように確認することである。

(2) 他人からの振込

スマートコントラクトに定義した振込処理には，振込元口座，振込先口座，振込金額の3要素が最低限必要である。振込処理では，振込元口座の持ち主

以外からの依頼によって振込が行われないようにする必要がある。そのためには，スマートコントラクト内で依頼者が振込元口座の持ち主であるかを確認し，持ち主でない場合は振込処理をさせない仕組みが必要となる。

またシステムによっては，振込元口座の持ち主ではなく，持ち主が他人に実行権限を委譲し代理実行させたい場合もある。その場合は，振込元口座の持ち主の署名を付けるなどして，振込元口座の持ち主からの依頼であることを証明する必要がある。その署名も，繰り返し使える署名ではなく，タイムスタンプやナンス（使い捨てのランダムな数値）を付け，スマートコントラクト内で1回しか使用できないようにする必要がある。

（3）仮想通貨の発行

不特定の参加者の間で合意形成を行い，不正な取引を抑制する仕組みをマイニングといい，マイニングの報酬として仮想通貨を手に入れることができる。マイニングを行うブロックチェーンでは，マイニングごとに仮想通貨が発行される。一方，マイニングを行わないシステムでは，システム内で扱う仮想通貨を発行して，特定口座に振込を行う機能が必要になる。通常，管理者が仮想通貨の発行と特定口座への振込を行う。その管理者が悪意を持っていれば，仮想通貨を不正に発行し，管理者の口座に振り込んで私腹を肥やすことが可能である。このような人的な脅威を防ぐためには，一定数以上の管理者の承認が得られない場合には処理が進まない，マルチシグ技術などを導入する。

このように，振込処理一つを取っても複数の脆弱性を含む処理を作り込んでしまう危険性があり，それを防ぐための方法も多岐にわたる。これらを網羅的に抑えるためには，分析手法だけではなく，スマートコントラクトで実現する機能を把握した上で，システム全体を認識する必要がある。また，スマートコントラクトで実現する機能やシステム構成によって脅威は異なるため，変更の度に分析が必要になる。

4. スマートコントラクト検証

スマートコントラクト開発における2.2節で挙げた課題を解決するために，我々は現在知られている3分類のリスクについて，静的解析に基づくスマートコントラクト検証技術を開発した。本章では，開発した検証技術と，それを用いて検出したリスクの一部を説明する。

4.1 開発した静的解析技術

静的解析技術の大まかな流れを，図-2に示す。まず，スマートコントラクトのソースコードを木構造にマッピングした抽象構文木から，取り得る処理の

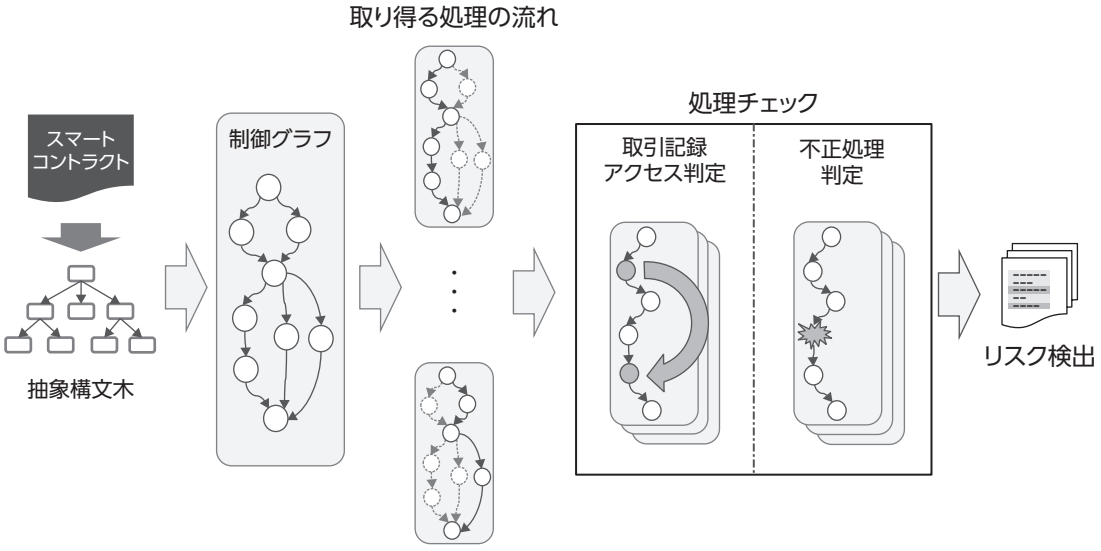


図-2　静的解析技術の流れ

流れを表す制御グラフを生成する。これを用いて，各処理を順番にチェックすることでリスクを検出する。本技術では，各処理のチェックにおいて，ブロックチェーン特有の取引記録アクセス判定や，あらかじめルール化した不正処理判定を行うことによって，リスクの存在を特定する。

この静的解析技術によって，以下で説明するリスクの抜け・漏れを生じることなく自動的に検出できる。これによって，開発者のレビューコストの削減や開発工程の短縮，およびスマートコントラクトの品質向上を実現できる。また，スマートコントラクトの品質が向上することによって，誤った取引が起こらず安全性が高いブロックチェーンシステムの構築につながる。

4.2 検出されたリスクの例

本節では，今回の検出対象となっている3分類のリスクについて，具体例を挙げて説明する。

(1) プログラミング言語に依存したリスク

Fabricでは，GoやNode.js，Javaといった汎用言語を用いて，スマートコントラクトを開発する。汎用言語を使うことは，開発者の学習コストを下げられるなどのメリットがある。その一方で，Solidity[3]やVyper[4]のようなスマートコントラクトを開発するために設計された言語では，ブロックチェーンの特性から，スマートコントラクトでは利用すべきではない機能が制限されている。汎用言語ではそのような制限がないため，開発者が誤って利用してしまうリスクがある。

その具体的な処理の例として，乱数がある。スマートコントラクトは各検証ノードにおいて独立して実行され，その結果がノード間で等しくなるかを検証する。しかし，乱数が含まれる場合には，結果はそれぞれの環境によって異なることが想定される。その場合には，検証ノード間で合意形成に至らず，正しく取引が行われなくなる。このようなリスクを避けるため，Solidityのようなスマートコントラクトのために設計された言語では乱数が存在しない。

(2) Fabricの仕様に依存したリスク

Fabricの仕様に依存したリスクが存在する。

例えば，Fabricでは一つのトランザクションで読み込みと書き込みを同時に行うことができる。しかし，書き込み後読み取り一貫性（Read-Your-Writes consistency）をサポートしていない。そのため，コードの記述順序を誤ると，あるトランザクションで書き込まれたデータを同じトランザクションで読み取ることはできない。その場合，書き込み前のデータが読み込まれるため，利用者の意図とは異なる取引結果になったり，二重支払いが発生したりする。

(3) データベースの仕様に依存したリスク

Fabricでは，ブロックチェーンの最新の状態を保管する，ワールドステートと呼ばれる機構を持つ。このワールドステートは，Apache CouchDB[5]などのデータベースを使って実装される。取引の実行にブロックチェーンの最新情報を必要とする場合，スマートコントラクトはワールドステートを参照する。そのため，Apache CouchDBの仕様上起こりうるPhantom Read[6]などの影響を受け，最新の取引状態を正しく確認できず，誤った取引を行ってしまう場合がある。このように，データベースの仕様に依存したリスクについても，開発者は注意を払う必要がある。

また，以上に挙げたようなブロックチェーンやブロックチェーンプラットフォームに特有なリスクのほかに，一般的なプログラミングにおいて問題となるようなリスクも，開発の際には考慮する必要がある。しかし，これは一般的な課題であるため，今回の技術の対象外となっている。

5. むすび

本稿では，開発した脅威分析手法およびスマートコントラクト検証技術について述べた。

脅威分析手法では，IPAガイドをベースとして一部の分析で分析アプローチを変えて分析することで，簡易にブロックチェーンに関わる部分についての脅威分析が可能であることを確認した。

今後これらの知見を基にして，富士通が構築するブロックチェーンを使用したシステムの信頼性強化に活用する。スマートコントラクト検証技術は，まずは富士通が提供するブロックチェーン関連サービスのスマートコントラクトに適用し，安全性の高い

サービス実現に貢献していく。

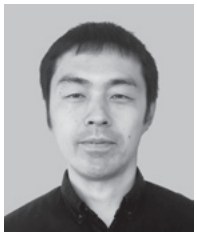
参考文献

(1) E. Androulaki et al.：Hyperledger Fabric：A Distributed Operating System for Permissioned Blockchains. EuroSys, 2018.
(2) IPA：制御システムのセキュリティリスク分析ガイド.
https://www.ipa.go.jp/files/000061925.pdf
(3) Solidity.
https://solidity.readthedocs.io/en/v0.4.24/
(4) Vyper：Vyper documentation.
https://vyper.readthedocs.io/en/latest/index.html
(5) Apache CouchDB.
http://couchdb.apache.org/
(6) H. Berenson et al.：A Critique of ANSI SQL Isolation Levels. SIGMOD Rec., 1995.
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-95-51.pdf

著者紹介

小櫻　文彦（こざくら　ふみひこ）
（株）富士通研究所
スーパーミドルウェア・ユニット
ブロックチェーン技術開発の研究に従事。

藤本　真吾（ふじもと　しんご）
（株）富士通研究所
セキュリティ研究所
ブロックチェーン技術開発の研究に従事。



野村　佳秀（のむら　よしひで）
（株）富士通研究所
ソフトウェア研究所
分散環境向け開発支援技術，ブロックチェーンシステムの開発支援技術の研究に従事。

山下　一寛（やました　かずひろ）
（株）富士通研究所
ソフトウェア研究所
ブロックチェーンシステムの開発支援技術の研究に従事。