分散型台帳技術を用いた金融取引に関する調査研究
ブロックチェーンを用いた金融取引における
技術リスクに関する調査研究

調査研究報告書

2018 年 3 月

金融庁　　株式会社電通国際情報サービス

【目次】

【図表目次】

1. はじめに

1.1. 調査研究の目的

フィンテックで用いられる IT 技術の中でも特にブロックチェーン技術については、大幅なコスト削減、利用者の利便性向上等により、市場に大きな変革をもたらす可能性が高いと予測されている。

こうした流れを受けて、金融庁は、2017 年 11 月に、平成 29 事務年度金融行政方針を公表し、ブロックチェーン技術について、その国際標準化の動きも視野に入れつつ、海外の最先端の人材や当局との連携強化に向けて、ブロックチェーン技術に関する国際的協同研究の推進を行う方針を示している。本調査研究は、その国際共同研究の一環として行うものである。

仮想通貨取引に代表される要素技術としてのブロックチェーンについて、実証実験等を通じて、様々な金融取引への応用に向けた研究開発が行われているものの、技術面でのリスクや脆弱性の評価について必ずしも進んでいるとはいえない。我が国においても、取引所システム等の脆弱性によって、仮想通貨そのものが盗まれるという事例も発生しているが、昨今、ビットコイン等の仮想通貨取引が普及し、悪意のある攻撃等を行うインセンティブは高くなっており、技術面でのリスクにも晒されるおそれがある。

我が国では、資金決済法を改正し、仮想通貨交換業者に対する登録制を導入するなど、制度面での整備が進められているが、ブロックチェーン技術に関する先進的な試行・取り組みを促進するためには、とりわけ、パブリックブロックチェーンが実装されている仮想通貨取引について、仮想通貨交換事業者というビジネスプラットフォームのリスクだけでなく、技術基盤であるブロックチェーン技術のリスクや脆弱性にも着目した、エコシステム全体の課題を洗い出し、その対応策等について検討していくことが重要であると認識している。

そこで本案件では、仮想通貨取引のエコシステム全体を考慮し、利用者保護等の観点から、ブロックチェーン技術のセキュリティに焦点を当て、技術面での課題や脆弱性などについてどのように今後対処していくべきかという観点で、調査研究を行った。

1.2. 現状の仮想通貨に関する方針

仮想通貨は、ブロックチェーン技術など従来見られなかった IT 関連技術が活用されており、仮想通貨交換業者においては、改正資金決済法の下、利用者保護等を図る上で、システム面を中心に高度な業務管理が求められている。また、2017 年初以来の仮想通貨価格の乱高下や仮想通貨の分岐など、仮想通貨市場では様々な動きが見られており、仮想通貨を取り巻く環境が利用者、あるいは金融システムに与える影響等を把握することが重要である。

金融庁は、イノベーション促進と利用者保護等のバランスに留意しつつ、仮想通貨市場の動向等を注視するとともに、改正資金決済法等の下で、仮想通貨交換業者において適切な業務運営体制が整備されているかモニタリングを行ってきた。

具体的には、仮想通貨交換業者において、仮想通貨を取り巻く環境の変化に応じて利用者に対する適切な説明・情報提供など、利用者保護を図るための態勢が整備されているか検証し、 また、安全かつ安定的なシステム運営や不正防止を通じた利用者からの信頼性確保の観点から、適切なリスク把握に基づいたシステムリスク管理態勢が整備されているか、 マネー・ローンダリングなどの不正行為を防止するための実効的な対策を検討・実施しているかを検証してきた。

このほか、最近では、仮想通貨を利用した資金調達である ICO（Initial Coin Offering）が 増加しているところ、ICO で発行される一定のトークンは資金決済法上の仮想通貨などに該当すると考えられ、その実態を十分に把握していくことが重要である。他方、詐欺的な ICO に対しては、関係省庁と連携して対応していくとともに、業界による自主的な対応の促進や利用者および事業者に対する ICO のリスクに係る注意喚起等を通じて、利用者保護を図っていくこととしている。

1.3. 方針と前提条件

1.3.1. 調査研究の方針

ブロックチェーンを用いた金融取引には、技術的要因によるリスクから仮想通貨の分岐などコミュニティのガバナンスに係るリスクまで、様々なリスクが存在する。ブロックチェーンを用いた金融取引への攻撃のリスクは、ブロックチェーンの形態（許可型／非許可型）、リスクの種類、取引の確定方法等によって異なる。

本調査研究では、これらの違いを踏まえ、ブロックチェーンを用いた金融取引のリスクについて、リスクを与える攻撃に対する回避策の有無や、将来的に金融取引の基盤として永続性を満たすべき要件を揺るがすような事象に注目して、既存の攻撃手段を俯瞰し、リスク評価を行うとともに、必要な実証実験を行うこととする。また、実証実験を通して技術的・制度的対策を考察する。

1.3.2. 前提条件

1.3.2.1. 対象とするブロックチェーンの形態

ブロックチェーンには大きく分けて以下の 2 種類が存在する。

① パブリックブロックチェーン（非許可型） ： 参加者が不特定多数（ビットコイン等）

② プライベートブロックチェーン（許可型） ： 参加者が限定される

この違いを踏まえ、本調査研究では上記①について机上でリスク比較を行う。リスク評価の結果、現時点においてリスクが高く、かつ、重大な問題となりうる脆弱性の存在が明らかになった場合には、その対応の可否や対応した場合のブロックチェーンへの影響等を調査するために、実証実験を行う。

1.3.2.2. 対象とするリスクの種類

本調査研究では、ブロックチェーンの技術的要因によるリスクを対象としてリスクを分析する。

ブロックチェーンに内在するリスクから引き起こされる攻撃に対し、攻撃ポイントとなる対象（暗号、プロトコル、システム、インターネット等）や利用者への影響を分析する。

1.3.2.3. 調査対象とするブロックチェーン上の取引の選定

現在、分散したノードに保存されたブロックチェーン上の台帳データの確定方法である合意アルゴリズムや実行可能な命令文（スクリプト）の種類などに応じて様々なパブリックブロックチェーンの仕組みやユースケースが提案されている。本調査研究として対象とするブロックチェーンは、先行研究が豊富に存在し、最も長い歴史を有するビットコインで利用されているブロックチェーン上で実行される取引を想定することとした。

【用語定義①】

■ ブロックチェーンとは？

ブロックチェーンとは、分散型台帳技術で、ビットコインやイーサリアムなどの仮想通貨の基盤となっている分散データベースである。ブロックチェーンは、ブロックと呼ばれるデータの単位がチェーン（鎖）のように連結されることで、データが保管される。

ブロックチェーンには確立した定義は未だなく、国際的な標準化団体等において定義の標準化等に向けた検討が進められている。他方、未だ確立された定義はないものの、日本ブロックチェーン協会は、ブロックチェーンを下記にように定義している。

「電子署名とハッシュポインタを使用し改ざん検出が容易なデータ構造を持ち、かつ当該データをネットワーク上に分散する多数のノードに保持させることで、高可用性およびデータ同一性等を実現する技術を広義のブロックチェーンと呼ぶ。」

ブロックチェーン技術は、仮想通貨や金融取引に限らず、ヘルスケアやサプライチェーンなど様々な分野での応用が期待されている。本報告書では、ブロックチェーン技術を利用した仮想通貨のうち、現時点で最も一般的に使用されているビットコインをモデルに、ブロックチェーンのメカニズムを説明する。

ビットコインでは通貨を、「トランザクション」（Transaction; 取引）によって表現する。各トランザクションは、複数のインプット（支払元）と複数のアウトプット（支払先）により構成されている。全てのインプットは支払元の秘密鍵により署名される必要がある。インプットには支払元のコインを入手したトランザクション（ハッシュ）とそのトランザクション内のアウトプットの何番目であるかを示すインデックスのみが書かれており、支払元のアドレスもコイン数量も書かれていない。アウトプットは支払先が所有している秘密鍵に対応する公開鍵をハッシュ関数に入力して生成される。

各トランザクションは、後述するマイナーノード（単にマイナーとも呼ぶ）により約10分毎に1ブロックにまとめられ、一つ前のブロックに連結されブロックチェーンに組み込まれると、その取引は承認されたことになる。このマイナーがブロックを一つ前のブロックに連結する行為をマイニングと呼び、そのブロックに含まれるトランザクションのインプットの合計数量とアウトプットの合計数量の差がブロック連結時にマイナーが手に入れることができる手数料となる。

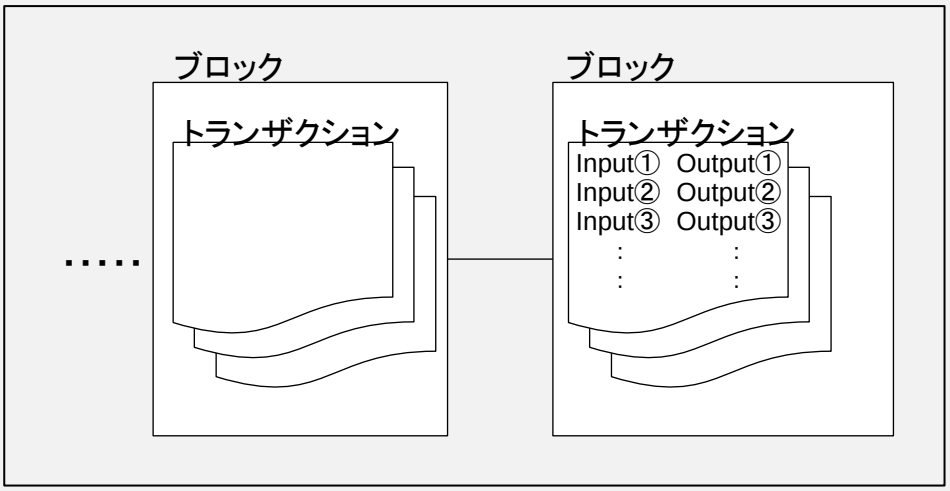


図 1-1 ビットコインにおけるブロックとトランザクションの構造

【用語定義②】

■ ビットコインネットワークとは？

ビットコインのネットワークは P2P 型で、ビットコインネットワークに参加している各コンピュータをノードと呼ぶ。ノードは、①ルーティング②フルブロックチェーンデータベース③マイニング④ウォレットという機能の集合体である。これら 4 機能を全て持つものがフルノードである。全てのノードはビットコインネットワークに参加するために必ずルーティング機能を持つが、その他の機能を持っているかいないかはノードの役割による。

① ルーティング

他のノードへの接続を管理し、データの送受信や検証を行うための機能である。

② フルブロックチェーンデータベース

現時点までに承認された全ての取引が記録される。

③ マイニング

ブロックを採掘するための機能である。

④ ウォレット

入出金に使用するビットコインアドレスなどを管理する機能である。

フルノードと呼ばれるノードは、フルブロックチェーンデータベースを有しているノードで、最新の完全なブロックチェーンの管理も行っている。フルノードは外部を参照することなく、自律的に後述するトランザクションを検証する。これに対し、フルブロックチェーンデータベースを所有せずブロックチェーンの一部の管理のみを行うノードもあり、SPV（Simplified Payment Verification）という簡便な方法でトランザクションを検証する。このようなノードは SPV ノードまたは軽量ノードと呼ばれている。

マイニング機能を有するマイナーノードは新しいブロックを作成する競争をしており、後述する Proof of Work アルゴリズムを解くための特別なハードウェアを稼働させている。いくつかのマイナーノードはフルノードでもあり、ブロックチェーンの完全なコピーを管理している。それ以外は後述するマイニングプールに参加している軽量ノードであり、フルノードを管理しているプールサーバに依存している。

ウォレット機能を有するユーザウォレットのノードの一部はデスクトップのビットコインクライアントという形で、フルノードとして参加している。スマートフォンなどリソースが限られているデバイスでは多くのユーザウォレットが SPV ノードになっている。

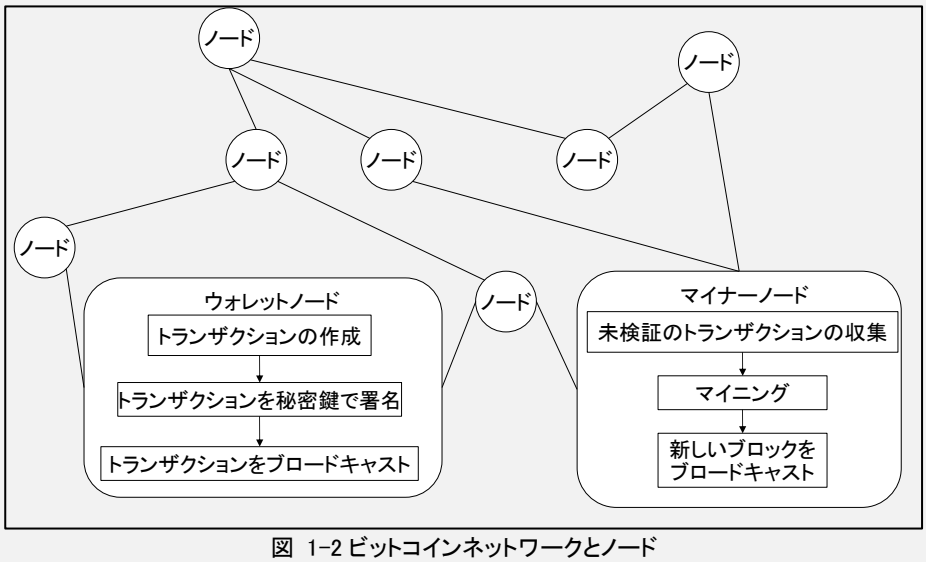


図 1-2 ビットコインネットワークとノード

【用語定義③】

■ ビットコインでの取引と合意形成

ビットコインネットワークで Alice から Bob に x ビットコインの送金が行われるプロセスは下記の通りである。

① 送金トランザクションの作成

Alice が過去に受け取ったトランザクションのうち未使用のアウトプットを、合計の金額が x ビットコイン以上となるように選択し、それらをインプットとし、Bob の秘密鍵に対応する公開鍵から作成したアドレスと支払額である x ビットコインをアウトプットとするトランザクションを Alice のウォレット上で作成する。インプットの合計金額を x'、マイナーに支払う手数料を x_f とすると、$x'-x$ が x_f よりも多い場合は、Alice は $x'-x-x_f$ をお釣りとして受け取るために、そのトランザクションのアウトプットとして Alice の秘密鍵に対応する公開鍵から作成したアドレスと $x'-x-x_f$ ビットコインを追加する。

② トランザクション送信

Alice がそのトランザクションをビットコインネットワークの隣接ノードに送信すると、そのトランザクションはノードからノードへ順次伝達されビットコインネットワーク内に拡散される。ここで、ビットコインネットワークでは帯域負荷を軽減するため、データ本体を送信する前に、そのデータのハッシュを inv メッセージとして相手ノードに送信し、相手ノードがそのデータを必要な場合にのみ getdata メッセージを送ってデータ本体を要求するという手順を取る。

③ トランザクション受信

マイナーノードはトランザクションを受信すると、各自のコンピュータ内に構築されたメモリプールに一時的に保管する。また、同時に上述のマイニングのための計算を行う。マイニングの手順は下記の通りである。まず、新しいブロックに格納するトランザクションをメモリプールから選択する。通常は手数料が高いトランザクションから優先的に格納される。また、直前のブロックのハッシュと、nonce と呼ばれる任意の文字列をそのブロックに格納する。次に、nonce を 1 文字ずつ変更し、ブロック全体のハッシュを計算する。このハッシュが、予め設定された「ディフィカルティー」（採掘難易度）の値よりも小さいと、その時の nonce が正解となり、その nonce を発見したマイナーがマイニング競争の勝者となる。勝者のマイナーのブロックがブロックチェーンに連結される最新のブロックとして採用され、その他の敗者のマイナーは、その nonce を使って計算したハッシュがディフィカルティーよりも小さいか、また、勝者が問題ないと判定したブロックの中身にエラーがないかを確認する。マイニングに参加したマイナーノードのうち過半数のノードの承認が得られると、そのブロックが公式にブロックチェーンに連結される。勝者のノードはコインベースと呼ばれるマイニングの報酬と、そのブロックに含まれるトランザクションの手数料を得ることができる。

④ フォーク

ブロックチェーンが連結されていく過程でフォークと呼ばれるブロックの枝分かれが発生する場合がある。例えば、同時に 2 人以上のマイナーがマイニングに成功し、それぞれが同時にビットコインネットワークに新たなブロックを伝播するとブロックチェーンは分岐する。この時点ではどちらも正当なブロックであるが、その後に作成されたブロックが連結されたブロックが正当とみなされ、他方のブロックは不正とみなされる。このように、枝分かれした場合、最長のブロックが「正当なブロックチェーン」となり、短い方のブロックは「オーファンブロック（孤立ブロック）」と呼ばれ、破棄される。つまり、オーファンブロックに入った取引は成立しなかったものとされる。ただし、オーファンブロックに入った取引は正当なものであれば、再度最長の「正当なブロックチェーン」に組み込まれる。

以上の合意形成アルゴリズムを、Proof-of-Work と呼ぶ。

このマイニングにおける計算作業量をハッシュパワーと呼び、マイニングに成功するためには高いハッシュパワーが必要である。したがって現在は、多くのマイナーは、マイニングプールと呼ばれる、複数のマイナーが集まり協力してマイニングを行うサービスに所属してマイニングを行なっている。

参考文献

「ビットコインとブロックチェーン:暗号通貨を支える技術」アンドレアス・M・アントノプロス（著），今井 崇也（翻訳），鳩貝 淳一郎（翻訳）

2. 調査研究の方法

2.1. 論文整理

2.1.1. 調査対象とする論文の選定

ブロックチェーンにおける一般的な攻撃手法とその対策技術について、調査対象とする論文を選定する。調査対象論文は以下の国際会議等を中心に、攻撃手法と対策技術に関するサーベイ論文を選定することとした。

（個別の対象論文については 別紙「調査対象論文一覧」 参照）

- Financial Cryptography および派生ワークショップ
- IACR Cryptology ePrint Archive

論文の選定結果を受けて、そこに記述された攻撃手法を「リスク」とみなし、評価を行う。

2.1.2. リスク評価観点の設定

リスクの評価方法としては、様々な手法が取りうるが、一般的にモデルベース・アプローチか、あるいは、指標ベースアプローチの2つの方法が考えられる。モデルベース・アプローチは文字通り理論的なモデルを用いて数学的にリスクを測定する方法である、一方、指標ベースアプローチは、リスクとの関連性を示すいくつかの指標を組み合わせて判定に用いる方法である。

本報告書は、ブロックチェーンを用いた金融取引への攻撃リスクについての計量化モデルを開発し、リスク評価を精緻化することは目的としているわけではない。今後、モデル等を用いたリスク評価の精緻化も期待されるが、今回の評価では、リスクレベルを直感的に把握しつつ、それらのリスクを類型化していくことに重点を置くこととした。

したがって、リスクの発生確率や影響度といったモデルベース・アプローチの基礎的な考え方も取り入れつつも、定性的な評価に基づく簡素なスコアリング方式を採用することとした。

① リスク・スコアリング方式

「利用者保護」と「金融取引システムの健全性」を維持するという2つの側面から、リスクが顕在化した場合の「影響度（被害の大きさ）」と「発生確率」に着目し評価を行う。

② リスク対応策の有無

回避・軽減・転嫁・許容等、リスク対応策の有無・状況について評価を行う。

2.1.3. リスク評価軸

2.1.3.1. リスク・スコアリング方式

以下の評価軸で各リスクを評価する。

なお、本研究のリスク評価では、従来の金融取引システムにおける攻撃リスク評価に基づき、影響度（被害規模）を「利用者」と「金融取引システム」に分割して評価することとした。

- 利用者への影響度（被害規模）
  - 影響範囲（大・中・小）
  - 深刻度（顕在化した際の被害の大きさ）（高・中・低）
- 金融取引システムへの影響度（被害規模）
  - 影響範囲（大・中・小）
  - 深刻度（顕在化した際の被害の大きさ）（高・中・低）

なお、本調査研究では利用者と金融取引システムを以下のように定義した。

- 利用者
  - 売手　：　仮想通貨を受け取るエンティティ。
  - 買手　：　仮想通貨を支払うエンティティ。
- 金融取引システム
  - マイナー　：　ブロックチェーンの作成に寄与するエンティティ。
    ノードにおいてブロックの発掘を行い、その対価として仮想通貨を受け取る。
  - 交換所　　：　仮想通貨を他の（仮想）通貨と交換するエンティティ。

ここで、マイナーはマイニングにより報酬として仮想通貨を受け取るため、本金融取引において利用者であるとも解釈できる。しかし、マイナーはブロックチェーンを運用する上で欠かせないエンティティであるため、本調査研究にではマイナーを金融取引システムの一部と定義することとした。

- 発生確率
  - 攻撃容易性（高・中・低）
    攻撃に際しての障壁（攻撃に要するコスト）が存在しない場合は攻撃容易性が高く、障壁が存在する場合は攻撃容易性が低くなる、と考える。
  - 攻撃者へのインセンティブ（高・中・低）
    攻撃に対するリターンが高い場合は攻撃者に高いインセンティブを与え、リターンが低い場合は低いインセンティブしか与えられない、と考える。

なお、本来のリスク評価であれば、各々の影響や深刻度について、定量的な数値をおく必要性があるが、前述のとおり、今回のリスク評価においては、ブロックチェーンにおける様々なリスクを定量的に把握することを目的とはせず、直感的に把握し、リスクを類型化していくことに重点を置くこととした。

各評価軸について、評価基準を表 2-1 に定める。

表 2-1 評価軸の評価基準

評価軸				評価基準
影響度（被害規模）	利用者への影響度	影響範囲	大	利用者のほとんどが攻撃対象となる［広範］
			中	利用者の一部が攻撃対象となる［中程度］
			小	利用者のほとんどが攻撃対象とならない［限定的］
		深刻度・被害の大きさ	高	利用者の資産価値のほとんどが無効化される［甚大］
			中	利用者の資産価値の一部が無効化される［中程度］
			低	利用者の資産価値のほとんどは無効化されない［軽微］
	金融取引システムへの影響度	影響範囲	大	マイナー・交換所のほとんどが攻撃対象となり、システミックリスクの可能性有。［広範］
			中	マイナー・交換所の一部が攻撃対象となるが限定的［中程度］
			小	マイナー・交換所のほとんどが攻撃対象とならない［限定的］
		深刻度・被害の大きさ	高	金融取引システムのほとんどが無効化される［甚大］
			中	金融取引システムの一部が無効化される［中程度］
			低	金融取引システムのほとんどは無効化されない［限定的］
発生確率		攻撃容易性	高	攻撃に際しての障壁がほとんど存在しない
			中	攻撃に際しての障壁が一部存在する
			低	攻撃に際しての強力な障壁が存在する
		攻撃者へのインセンティブ	高	攻撃者のほとんどにインセンティブを与える
			中	攻撃者の一部にインセンティブを与える
			低	攻撃者のほとんどにインセンティブを与えられない

次に、各評価軸の項目値にハイレベルなものから順に 3 点/2 点/1 点と点数を割り当てる。各項目値の最高値をその評価軸の値とする。例えば、「利用者への影響度」において、「影響範囲」が 3 点で「深刻度」が 1 点の場合、評価軸「利用者への影響度」の評価値は 3 点となる。

このように算出した評価値から、「利用者への影響度」×「発生確率」および「金融取引システムへの影響度」×「発生確率」の 2 種類でリスクレベルを評価する。

表 2-2 リスクレベル（利用者への影響度×発生確率）

リスクレベル		利用者への影響度		
		大	中	小
発生確率	大	9（高）	6（中）	3（低）
	中	6（中）	4（中）	2（低）
	小	3（低）	2（低）	1（低）

表 2-3 リスクレベル（金融取引システムへの影響度×発生確率）

リスクレベル		金融取引システムへの影響度		
		大	中	小
発生確率	大	9（高）	6（中）	3（低）
	中	6（中）	4（中）	2（低）
	小	3（低）	2（低）	1（低）

2.1.3.2. リスクへの対応策

各リスクについて、リスク対応策を以下の 3 種類に分類する。

① 実装上/運用上の対応策有

② 理論的な対応策有（実際のシステム等には未適応）

③ 対応策無

ここで扱う対応策は、前述 1.3.2.2.対象とするリスクの種類のとおり技術的要因によるリスクを分析対象としたため、ここでの対応策も技術的対応策に限定して記述、分類を実施した、当局規制を含む制度的な対応については 別途 5.考察にて 検討、記述する。

3.リスク評価結果

前述の評価基準に基づき評価を実施した。評価の準備として、まずブロックチェーンの攻撃に関する論文を整理し、考えられるリスクを一覧化する。そして、それぞれのリスクが顕在化した際の被害の内容および影響範囲、およびリスクの回避や軽減に必要な対応策を明確にする。

それぞれのリスク要因に対して、以下の情報を整理した。(個別の攻撃リスクの詳細については 6.1 を参照)

- 概要(攻撃シナリオと攻撃手順)
- 概要(対応策)
- 参考情報
  - 攻撃の内容
  - 影響
  - 主要な攻撃対象
- 該当論文
- 著者およびタイトル
- 発表媒体
- 発表媒体の種類

3.1. リスク・スコアリング結果

それぞれのリスク要因に対し、2.1.3.リスク評価軸にしたがってリスク評価を実施した結果を表 3-1 に示す。
各々のリスクに関する内容については、6.1 を参照されたい。

表 3-1 リスク評価

番号	リスク要因	リスクレベル (利用者への影響度)	リスクレベル (金融取引システムへの影響度)
1	Double spending or Race attack	3(低)	1(低)
2	Finney attack	3(低)	1(低)
3	Brute force attack	6(中)	4(中)
4	Vector 76 or one-confirmation attack	2(低)	2(低)
5	>50% hashpower or Goldfinger (Majority attack)	3(低)	3(低)
6	Block discarding or Selfish mining	2(低)	4(中)
7	Block withholding	2(低)	4(中)
8	Bribery attacks	4(中)	4(中)
9	Refund attacks	6(中)	2(低)
10	Punitive and Feather forking	3(低)	1(低)
11	Wallet theft	9(高)	6(中)
12	Transaction malleability	6(中)	3(低)
13	Time jacking	3(低)	6(中)
14	Sybil	6(中)	6(中)
15	DDoS	6(中)	9(高)
16	Eclipse or netsplit	4(中)	4(中)
17	Tampering	4(中)	4(中)
18	Deanonymization	3(低)	1(低)
19	Compromise of underlying cryptographic algorithms	9(高)	9(高)

また、表 3-1 の内容を、リスクレベル（利用者への影響度）をx軸に、リスクレベル（金融取引システムへの影響度）をy軸にとる二次元座標にプロットしたものが図 3-1 である。

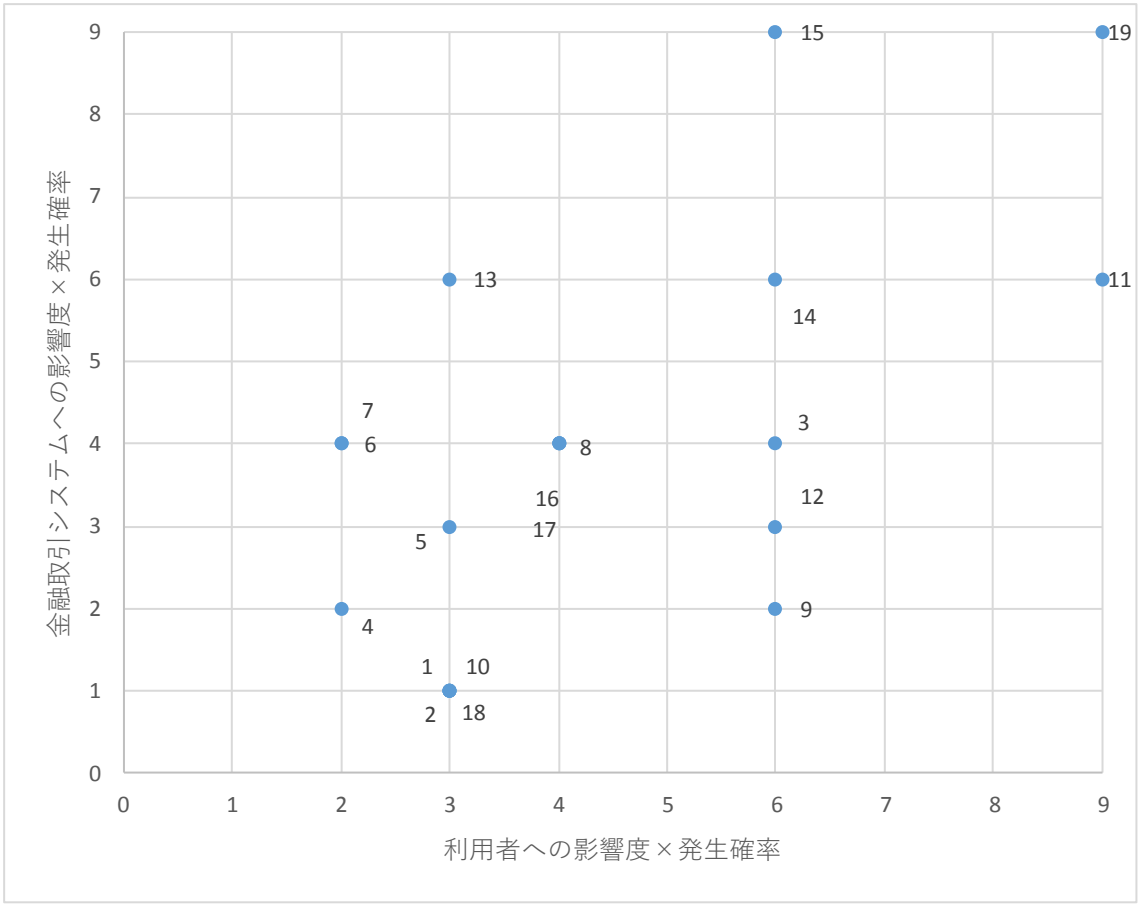


図 3-1 攻撃リスク評価

図 3-1 から、利用者・金融取引システム双方に深刻な影響を与える攻撃手段は、「DDoS（15）」と「Compromise of cryptographic algorithm（19）」であることが分かる。

3.2. リスクへの対応策

さらに、それぞれのリスク要因に対し、リスクへの対応策を分類した。分類結果を表 3-2 に示す。

表 3-2 リスクへの対応策の分類

リスクへの対応策	リスク要因
A)　実装上/運用上の対応策有	1. Double spending or Race attack 2. Finney attack 3. Brute force attack 4. Vector 76 or one-confirmation attack 11. Wallet theft 12. Transaction malleability 16. Eclipse or netsplit 18. Deanonymization
B）理論的な対応策有	6. Block discarding or Selfish mining 7. Block withholding 9. Refund attacks 13. Time jacking 15. DDoS 17. Tampering 19. Compromise of underlying cryptographic algorithms
C）対応策無	5. >50% hashpower or Goldfinger attack 8. Bribery attacks 10. Punitive and Feather forking 14. Sybil

3.3. 総合評価

仮想通貨取引においては、様々なリスクが存在することは従前より認識されていたことである。抽出した 19 のリスクについて、上記のリスク評価手法を用いた結果、リスクが高いと判断される攻撃（あるいは脆弱性）、およびリスクは高くないが対応策未確認の攻撃を概ね分類することができた。ここでは高リスクと分類された攻撃手法、対応策未確認の攻撃手法について各々の概略を述べ、利用者への影響度を考慮した上で評価をすると共に、一般的に情報システムが暗号技術を適切に扱うための論点も踏まえて、総合評価としてまとめたい。

3.3.1. 高リスクと評価された攻撃、および脆弱性について

本リスク評価では、リスクレベルを利用者への影響度と、金融取引システムへの影響度の二種類の評価を実施した。その結果、利用者への影響度において、単独でリスクレベル高となったのは、「11.Wallet theft」であり、金融取引システムへの影響度において、単独でリスクレベル高となったのは、「15.DDOS」である。また「19.Compromise of cryptographic algorithm」は双方でリスクレベル高となった。特に Wallet theft の対応策は提示されているものの、その対応策が適切に実施されていないことから実際に仮想通貨が盗難される被害も発生している。また Compromise of cryptographic algorithm においては、潜在的に仮想通貨そのものの価値を毀損するリスクが高く、顕在化した場合の影響が大きいと判断される。以下では、各々の攻撃の概略について記載する。

3.3.1.1. Wallet theft の概略

ユーザーのウォレットに侵入することで、不正に仮想通貨を取得することを目的とした攻撃であり、攻撃対象は仮想通貨のユーザーである。ウォレットには様々な種類があるが、特にウェブウォレット形式の場合、秘密鍵の管理を仮想通貨交換業者等のサービス提供者が行い、ユーザーは ID とパスワードを入力してログインしてウォレットを利用する形態が一般的である。したがって、攻撃者はハッキングやウイルス等で ID とパスワードや秘密鍵そのものを入手できる可能性がある。

3.3.1.2. DDOS 攻撃の概略

この攻撃では、攻撃者が多数のクライアントからネットワーク上に偽のブロックやトランザクションのような不正なデータを大量に送信することで、ネットワークのリソースを枯渇させ、ユーザーがネットワークにアクセスすることを阻害し、マイナーに通常のユーザーからのブロックを棄却させることが可能となりうる。

この攻撃は、ビットコインネットワークにおいて、誰でも低コストで不正なデータを送信できるという脆弱性を突いている。

3.3.1.3. Compromise of cryptographic algorithm の概略

ブロックチェーンのアーキテクチャでは、使用されている暗号アルゴリズムに関してアルゴリズムの移行が考慮されていないといえる。ブロックチェーンで使用されている暗号アルゴリズムが危殆化した時に、適切な措置が適用されない場合に、それを基盤とする金融取引の全ての価値が失われるリスクがある。

暗号アルゴリズムの危殆化には二種類の顕在化パターンが存在する。ひとつは、計算機能力の発展に伴い暗号アルゴリズムが計算量的に解読される場合である。もうひとつは、暗号アルゴリズムに対して効率的な解読方法が発見された場合[1]である。これまでの近代の暗号の歴史に鑑みても、前者は数十年ごとに必ず発生するリスクである一方、後者は今すぐにでも発生するリスクである。

3.3.2. 対応策未確認（対応策が論理的に存在するが、未実装）の攻撃について

対応策未確認の攻撃において、多くが、ブロックを覆す、あるいは遅延させる、マイニングプールの計算資源を無駄にする等の攻撃である。本調査研究開発時点においては、それらの対応策が未実装であり、いずれもビットコインの仕組み上の脆弱性をついたものである。ただし、いずれも現時点においては、こうした攻撃を通じて、利用者の資産が毀損されたことは確認できていない。

3.3.2.1. Block discarding or Selfish mining の概略

一般のマイナーやマイニングプールの計算資源を無駄にすることで、攻撃者によるブロックの検証結果が採用され易くし、通常よりも多くの報酬を得ることを目的とした攻撃である。一般のマイナーには公開しないマイニング済みブロック（*1）を確保しておき、一般のマイナーがブロックをマイニングした直後などに、*1 を公開することで、自分がマイニングしたブロックの採用確率を高める攻撃である。現実的に可能な対策として提案されている案は、一般のマイナーが競合する 2 つのブロックを受信した場合に、どちらのブロックに対してマイニングを継続するかを一様ランダムに選択するというアルゴリズムに変更するということである。

3.3.2.2. Block withholding の概略

マイニングして発見したブロックを隠し持っておくことで、マイニングプールに損害を与える、あるいは、不当に利益を得る、等を目的とした攻撃である。攻撃対象はマイニングプールの運営者や参加者である。攻撃に必要とされる条件としては、攻撃者はマイニングプールに参加している必要がある。また、攻撃者のハッシュパワーが大きいほど攻撃の影響が大きい。この攻撃への対応策として提案されている案は、マイニングプールにおいて、マイナーが発見したブロックがマイニングプールの運営者に提出されるまでは有効かどうかが識別されない（マイナーがブロックを隠し持つ判断ができなくなる）ように、ビットコインのプロトコルを改良することである。

3.3.2.3. Refund attacks の概略

払戻を活用して、取引履歴を隠し不正な利益を得ることを目的とした攻撃である。この攻撃の種類として、BIP70（Bitcoin Improvement Proposal 70）における認証の脆弱性に着目した Silkroad attack と、既存の支払処理の払戻規約を悪用した Marketplace Trader attack の 2 種類が考案されている 。

[1] 2005 年に広く利用されている暗号核的ハッシュ関数である SHA-1 の効率的な攻撃手法が暗号学者のブログで発表された。ブログは https://www.schneier.com/blog/archives/2005/02/sha1_broken.html を参照。

この攻撃は、ビットコインでの取引において払戻を行う際に、商品の売手が払戻のアドレスを、支払を行った買手と同一人物であることを確認できないという脆弱性を突いている。この攻撃への対応策として提案されている案は、売手に、受信した払戻のアドレスが、支払を承認した買手と同一人物によって承認されたことを暗号学的に証明できる検証可能な証拠を提供することである。

3.3.2.4. Time jacking の概略

ブロックチェーンのネットワーク時間を不正に操作することで二重使用を試みること、あるいは、他のマイナーの計算資源の浪費、トランザクションの承認スピードの低下、などの攻撃を成功させることを目的とした攻撃である。この攻撃は、ビットコインネットワークにおいて、不正なバージョンメッセージを送信することにより任意のノードのネットワーク時間を進めたり遅らせたりすることが可能であるという脆弱性を突いている。この攻撃への対応策として提案されている案の1つは、ブロックを検証する際、ブロックタイムスタンプとして、過去のブロックのタイムスタンプの中央値から計算した値を使用することである。

3.3.2.5. Tampering の概略

特定のノードからのトランザクションとブロックの伝搬を遅らせ、DDoS 攻撃に利用したり、攻撃者のマイニング能力を相対的に向上させたり、二重使用攻撃に利用することを目的とした攻撃である。この攻撃への対応策として提案されている案の1つは、タイムアウト時間を各ノードがメッセージのサイズなどに応じて動的に設定する方式に変更することである。

3.3.3. 評価

3.3.3.1. リスク評価結果

これまでの評価結果について、以下の 2 軸で整理すると図 3-2 のようになる。

- リスクレベル高
- 対応策未確認（対応策が理論的に存在するが未実装）

整理した結果を図示したものを図 3-2 に示す。図より、リスクレベル高かつ対応策未確認となるリスクが「DDoS（15）」と「Compromise of cryptographic algorithm（19）」となる。

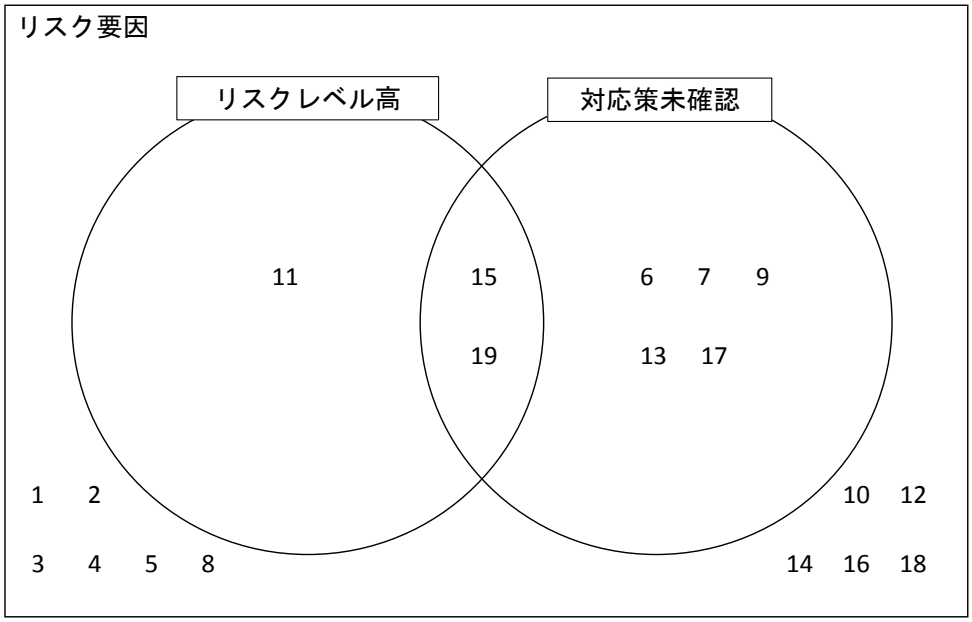


図 3-2 総合評価結果

今回の調査、およびリスク評価結果から仮想通貨取引において浮き彫りになるリスクは、仮想通貨利用者の資産保全、および仮想通貨そのものの価値の毀損のリスクが存在することであった。

前述の Wallet theft（11）のような攻撃は仮想通貨取引所に対するハッキング被害として、既に顕在化しているリスクである。

また、暗号技術の危殆化は現時点で顕在化していないものの、潜在的に仮想通貨自体の価値が損なわれるリスクである。暗号の危殆化については、本調査研究開始時点では、対応策未確認の状態であり、仮想通貨そのものの価値の毀損化を招くというリスクレベルも高い。加えて、このリスク自体の存在が十分に認知されているとは考えられないことから、さらに調査する必要があると判断し、4 章で説明する実証実験を行うことに至った。その他、同じようにリスクレベル高、かつ、対応策未確認に分類される DDOS 攻撃については一定の効果が見込める対応策も提示されていることから、実証実験の対象からは外すこととした。

3.3.3.2. 暗号技術を用いた情報システムとしての評価

ここまでリスク評価を行ってきた仮想通貨の取引はブロックチェーンという暗号技術に立脚した分散システムのうえで、その価値を維持、交換できる仕組みである。そして、その仮想通貨の基盤技術であるブロックチェーンは暗号技術を巧みに利用した情報システムととらえることができる。したがって、ここでは暗号技術を用いた情報システムとしてのブロックチェーンに焦点を当てて、一般的に情報システムが暗号技術を適切に扱うために検討すべき事項である 4 つの論点を整理した上で、改めて現状のパブリックブロックチェーンにおいて、暗号技術を適切に取り扱えているかを整理してみたい。

一般的に情報システムが暗号技術を適切に扱うための論点を整理すると以下の通りとなる。

(1) 安全な暗号方式であるか

(2) 安全な暗号通信プロトコルとして安全性が保たれているか

(3) 暗号技術を用いたセキュリティ対策がソフトウェアに適切に実装できているか

(4) 暗号鍵管理が適切に運用されているか

まず、(1)(2)については、暗号の理論や暗号を使った情報システムの設計の問題である。したがって、一般的に学術的な分析や専門家による安全性の評価が重要となる。

例えば、(1)の暗号方式の安全性に関しては、「暗号技術検討会および関連委員会（CRYPTREC）」が安全性、および実装性能を評価しており、利用実績が十分であるもの、あるいは、今後普及が見込まれるものを公開しており、こうした評価結果が参考となる。

(2)はハッシュ関数と電子署名によるチェーンや、分散合意形成等の仕組みを実現するために、暗号通信プロトコルを用いた適切な設計されているか否かを評価する論点である。例えば、ビットコインでは、ハッシュ関数を用いたブロックチェーンの生成や秘密鍵・公開鍵とスクリプトを用いて取引の正当性の確認、Proof of Work によるブロックへの書き込みなど、暗号技術を活用して全体として運用主体の存在を前提としない、価値の移転を実現している。

(3)は、適切な暗号方式、および暗号通信プロトコルをパブリックブロックチェーンに実装ができているか否かを評価する論点である。例えば、安全な暗号方式を用いて、適切なプロトコルを設計したとしても、それを実行するプログラムにバグが存在すれば、取引の安全性は確保することができない。このように、情報システムの安全性を評価する際には実装におけるリスクについても確認することが必要となる。

(4)の論点は情報システムにおいて、暗号技術を利用する際に、必ず検討しなければいけない暗号鍵管理の運用の課題である。(1)～(3)をいくら安全にしても、(4)の暗号鍵の管理がおろそかであれば取引の安全性は確保できない。特に仮想通貨においては、秘密鍵が実質的に資産そのものとなっていると考えられることから、暗号鍵の管理は非常に重要なポイントとなる。なお、(4)については、安全な暗号鍵技術を採用し、実装することと、その暗号鍵の運用の適切に行うこと、両方を満たす必要がある。

なお、これら 4 つの論点は、すべてが適切に実施できて初めて情報システムの安全性が保たれることに留意する必要がある。つまり、このうちどれか一つでも問題が生じればそうした情報システムに立脚して実施される取引の安全性は確保することができない。

また、ブロックチェーンは、一般的な暗号を用いた情報システムに比しても、安全性を確保するために暗号技術に依存する割合が高いため、上記のような視点にしたがって安全性を評価し、高めていく必要性がより高いと考えられる。

以下では、このような視点に基づいて、今回リスク評価対象とした全 19 の攻撃手法、および脆弱性の問題点を分類した上で、それぞれの論点について評価を行う。ただし、今回リスク評価対象とした全19の攻撃手法、および脆弱性には、暗号技術に依存しない問題等もあることから、全てをこの分類にできるわけではない。

表 3-3 暗号技術を適切に扱うための論点整理に基づく分類

分類	攻撃手法、および脆弱性	理由
(1)	19. Compromise of cryptographic algorithm	19.暗号危殆化を考慮した設計をされていないため
(2)	6.Block discarding or Selfish mining 7.Block withholding 9.Refund attacks 12.Transaction malleability 15.DDOS	6.および 7.ブロック内のトランザクション検証作業を、暗号学的ハッシュ関数の計算競争により実現をさせる設計にした結果、自己都合のブロック作成を誘発する仕組みとなったため。 9.払戻しアドレスと支払い元アドレスを暗号学的に証明する手段が実装できていないため 12.ハッシュ関数による生成されるトランザクション ID が改変できるという脆弱性であるため 15.トランザクションの伝播においては、Inventory と呼ばれるトランザクションのハッシュ値が送信され、受信したノードはそのハッシュ値が未知の場合、送信元に問合せを行う仕組みとなっており、不正なトランザクションハッシュ値となる Inventory を多く発生させることで遅延を比較的容易に発生させることができるため。
(3)	該当無し	
(4)	11. Wallet theft	11.暗号鍵管理が適切にできていないことから発生しているため

3.3.3.2.1 (1)の論点からみた脆弱性について

19. Compromise of cryptographic algorithm が該当する。いわゆる、暗号技術そのものの危殆化に伴う、脆弱性である。例えば、SHA-1 のように広く利用されている暗号学的ハッシュ関数においても、2005 年に危殆化によるハッシュ衝突性が確認されており、SHA-1 から別のハッシュ関数への移行が多くの情報システムにおいて発生した。この移行が可能であったのは、それらの情報システムに運営主体が存在し、暗号技術の入れ替えも想定した作りとなっていたためである。

一方、仮想通貨においては、価値そのものを、暗号技術を利用したブロックチェーン上に保存している。このため、利用している暗号技術に危殆化が発覚した場合においては、特に深刻な問題を引き起こす可能性がある。今回調査研究対象としたビットコインにおいては、様々な目的で暗号学的ハッシュ関数が多用されているが、ビットコインで利用されている代表的な暗号技術とその利用用途をまとめると以下のようになる。

表 3-4 ビットコインで利用されている暗号技術の利用用途

#	ビットコインでの暗号技術利用用途	主な目的	暗号技術
1	ブロックのハッシュチェーン	ブロック間の順序連続性の維持、欠陥や追加がないことの証明目的	SHA-256
2	Markle Tree	トランザクションが当該ブロックに格納されていることの証明目的	SHA-256
3	トランザクション識別子（ID）	トランザクションデータ参照のポインタ、およびトランザクションの完全性を検証する目的	SHA-256
4	ビットコインアドレス	公開鍵のハッシュ値として生成することでアドレスの一意性を確保する目的	SHA-256, RIPEMD-160
5	ビットコインアドレスのチェックサム	入力されたアドレスの間違え防止目的	SHA-256
6	Proof of Work におけるハッシュ、および nonce	ブロックの検証作業を終えたことを証明する目的	SHA-256
7	トランザクションへのデジタル署名	トランザクション作成者の確認目的	ECDSA 楕円暗号

例えば、SHA-256 の危殆化が発生した場合、ブロック間の順序連続性が崩れ、過去のブロックを改ざんできる可能性や、本来存在しないトランザクションを Markle Tree の値を変えずにブロックに取り込むことが可能となる。したがって、ブロックチェーンそのものの信頼が損なわれることになり、そのブロックチェーン上に価値を保存している仮想通貨そのものの価値の毀損を招く問題となりうる。このリスクに対して、上記の通りビットコインでは、現状、暗号技術の移行は想定されていない。

3.3.3.2.2 (2)の論点からみた脆弱性について

6.Block discarding or Selfish mining、7.Block withholding、12.Transaction malleability 、15.DDOS 等が該当する。例えば、12.Transaction malleability はトランザクション ID を書き換えることができる脆弱性であり、トランザクション ID のみで、取引の識別を行っている場合に発生する問題である。かつて、Mt.Gox 社の問題においても、1つの原因となったと言われている事象である[2]。その他、6.Block discarding or Selfish mining、7.Block withholding 等は、ブロックが覆ることやブロックの遅延を引き起こす問題であるものの、現時点までにおいては利用者保護の観点から大きな問題とはなっていない。ただし、一般的に、暗号を用いた情報システムの安全性は長期的な運用を通じてその安全性が確認されること、またビットコインで活用されているプロトコル全体の安全性を学術的に確認することは完了していないことには留意が必要である。

今後ブロックが覆るような攻撃手法が多様化、あるいは、高度化してきた場合においては、仮想通貨を決済手段として利用している事業者間においてシステミックリスクが発生しうることも、留意していく必要がある。さらに、ビットコイン以外の仮想通貨においては、ビットコインとは全く異なる暗号技術やプロトコルが用いられることもあり、ビットコインにおいて現時点までにおいて問題が生じていないことが他の仮想通貨にもあてはまるわけではないことにも留意が必要である。

3.3.3.2.3 (3)の論点からみた脆弱性について

今回の調査対象とした 19 の論文からは該当するものがなかった。一般的に暗号技術を情報システムに用いた場合においては、この(3)に該当する実装上の不具合（所謂、バグ）が一番多く発生しうる脆弱性であるといえる。ビットコインで用いられるビットコイン・コア・プログラムはあらゆる仮想通貨の実装の中で最も多くの開発者が関わっているプログラムの一つであり、バグ等の検証が入念に行われているが、他の実装や他の仮想通貨においては安全な実装となっているか十分な検証がなされていない場合も存在することには留意が必要である。[3]

[2] なお、Transaction malleability は 2017 年に SegWit が導入されたことで一定の解決がなされている。

[3] 一部のハードウェアウォレットの実装に不具合があり、仮想通貨を受領するために、自身の新規アドレスを生成しているつもりが、ハッカーのアドレスを生成するプログラムが仕組まれており、仮想通貨をハッカーに送信してしまう事象も発生している。

3.3.3.2.4 (4)の論点からみた脆弱性について

11. Wallet theft が該当する。現状、パブリックブロックチェーンの仕組みは利用者（ここでは仮想通貨交換所や、支払い手段として仮想通貨を利用するリテール企業、あるいは、利用者個人等）が適切に暗号鍵を管理することを前提として設計されている。したがって、その前提が崩れた場合には、安全な取引を実現することができないことを正しく理解して利用することが重要である。実際、昨今発生している仮想通貨における盗難等の問題も多くがパブリックブロックチェーンを利用する側の秘密鍵管理に起因する脆弱性を利用した攻撃によるものである。したがって、暗号鍵を誰が、どのように管理するのかは取引の安全性を確保する上で非常に重要な論点となる。

具体的には、現状、多くの場合に仮想通貨交換業者が顧客の暗号鍵を管理していることを踏まえ、そうした事業者のサイバーセキュリティを高める[4]ことが必要となる。十分なセキュリティ水準を確保するためには既存の暗号技術を活用した情報システムにおけるサイバーセキュリティの知見を活用した上で、ブロックチェーンの暗号鍵特有の脆弱性にも対応することが必要となる。

以上、パブリックブロックチェーン上の取引について、一般的な情報システムにおいて、暗号技術を適切に取り扱うための論点(1)～(4)に基づき整理をしてきた。この論点整理の仕方は、いわゆる管理主体が存在する情報システムにおいて適用されてきたものであるが、暗号技術に立脚したパブリックブロックチェーンの評価においても一定の有効性があると考えられる。

3.3.3.3. まとめ

以上、今回採用したリスク評価手法、加えて、暗号技術を情報システムに適切に取り扱うための論点から仮想通貨取引におけるリスク評価をしてきた。上記の通り、安全性を確保するためには(1)～(4)すべてに対応することが必須であるが、現状、現にリスクが顕在化している 11.Wallet theft（暗号鍵の管理・運用の論点）と中長期的に必ず顕在化し、かつ今すぐに顕在化することも否定できない 19. Compromise of cryptographic algorithm については特に重点的に検討を行う必要があると評価される。したがって、5.考察にて、この二つのリスクを中心に今後の取りうる施策について検討を行いたい。

3.4. 実証実験の必要性について

前述の 3.3 総合評価　において、「DDoS（15）」と「Compromise of cryptographic algorithm（19）」については、リスクレベル高でかつ対応策未確認（対応策が理論的に存在するが未実装）な二大リスク要因であり、さらなる調査が必要と考えられる。対応策をさらに調査したところ、以下が判明した。

- DDoS（15）

  対応策に関する論文では理論的な提案がされており、対応策を適用した場合の性能についても論文中で詳細に検討され、具体的な数値データが示されている。

- Compromise of cryptographic algorithm（19）

  対応策に関する論文では理論的な提案がされているが、対応策を適用した場合の性能については論文中では議論されておらず、また機能追加した場合の性能評価も実装による評価が必要である。

上記のとおり、DDoS（15）については、対応策について、完全ではないものリスク軽減の一定の評価がなされていることが判明したが、Compromise of cryptographic algorithm（19）理論的な提案のみにとどまり、対応策を実施した場合のブロックチェーンネットワークに与える影響の性能評価も未実施であることが判明した。

暗号の危殆化は情報システム全般にかかる共通のリスクであるが、仮想通貨の価値がブロックチェーンの暗号技術の信頼性に立脚しており、リスクが顕在化した時の影響が大きいことから、本研究において危殆化対応策の性能評価を、実施することは、大変有意義と認められる。客観的な性能評価の実施には例えば BSafe.network のような国際的な大学間のパブリックチェーンを利用した実証実験が最も適していると考えられる。

[4] セキュリティを高めるための手法としてハードウェアウォレットのようなオフラインで秘密鍵を管理する手法も提案されているが、最も利用者が多いと考えられる Bitcoin でさえも、ハードウェアウォレットの種類は数種類しか存在しない点にも留意する必要がある。仮にハードウェアウォレットの採用が取引所や一般ユーザーに広まった場合、そのハードウェアウォレットの実装に問題があった場合には、1つの脆弱性によって多くのユーザーが被害をこうむる可能性もある。実装が一部の開発事業者に限られるという点は、分散システムにおいても、単一障害点を発生させる可能性がある点に留意する必要がある。

3.5. 実証実験対象の選定

既存攻撃や学術的な論文に対する評価の結果（詳細は攻撃論文調査結果を参照）として、ブロックチェーンで用いられている暗号アルゴリズムが危殆化するリスクは、ブロックチェーンを用いた金融取引の価値そのものに大きな影響を与え、全ての利用者が影響を受けると分析され、ブロックチェーンを用いた金融取引に対して大きなリスクとなることが分かった。

暗号アルゴリズム危殆化のリスクに対し、長期署名技術をブロックチェーンに適用する方式が提案されている [1]。この方式では、ブロックチェーンで用いられている暗号アルゴリズムを新しいものに移行する際[5]、それまでに生成されたブロックチェーンが確かに正当であることを保証するデータを、新しい暗号アルゴリズムから計算して付与する。これにより、過去にブロックチェーンで用いられていた暗号アルゴリズムが危殆化したとしても、過去のブロックチェーンの正当性を確認することができる。つまり、ブロックチェーンの正当性を暗号アルゴリズムのライフサイクルによらず長期的に保証できる。

長期署名技術を適用したブロックチェーンは、通常のブロックチェーンに対し追加のデータを必要とするため、通常のブロックチェーンより負荷が大きくなると予想される。しかし、長期署名技術を適用したブロックチェーンを利用することで、ノードに対しどの程度影響（実行性能、データ量など）を与えるかや、トランザクションを実施する上でどの程度通信データ量に影響を与えるかが、机上での検討では把握しにくい。そこで、本実証実験でこのようなデータを収集して分析することで、ビットコインをはじめとするブロックチェーンを用いた金融取引への大きな指針となるものと判断した。

[5] 新しい暗号アルゴリズムに移行する際、その時点で安全性が十分保証されているアルゴリズムを選定する必要がある。例えば、NIST (アメリカ国立標準技術研究所)や CRYPTREC (Cryptography Research and Evaluation Committees) 等で推奨されている暗号アルゴリズムの中から選定することが望ましい。

4. 実証実験

本章では実証実験の目的、実験対象機能、実験結果、およびその考察を記載する。

4.1. 目的

本実証実験の目的は、ブロックチェーンで利用される暗号技術が危殆化した場合の対応策として、長期署名技術が有効となりうるか、また有効となりうる場合、長期署名技術による既存のブロックチェーンへの影響を評価することである。また長期署名技術に加えて、Proof of work で利用されるハッシュ関数の変更や、トランザクションの署名における鍵長の変更も実施する。これらの各種対応策による性能面での影響を確認するために、以下の項目を測定する。

- 各ノードの実行速度への影響
- 各ノードのデータ量への影響
- トランザクションを実施する上でのネットワークの通信データ量への影響

また各種影響は下記のとおりと想定している。

- 各ノードの実行速度への影響の想定

  長期署名技術自体は過去のブロックを新たなハッシュアルゴリズムで計算することであり、かつ、移行時の一時的なものであるため、処理速度への影響は大きくはならない、あるいはなったとしても、一時的なものと想定している。一方、ブロックチェーンの処理性能に大きく影響を与えるのが署名の検証である。このため、後述する署名に使う鍵長の変更は処理性能に大きく影響を及ぼす可能性があると想定している。

- 各ノードのデータ量への影響の想定

  長期署名導入によりブロックヘッダーサイズが増加するが、1MB のブロックに対し軽微な増加となると想定している。一方、署名データの増加はトランザクション数に応じて増加するため、データ量に与える影響は大きくなると想定している。

- トランザクションを実施する上でのネットワークの通信データ量への影響の想定

  ブロックチェーンのデータ通信はトランザクション自体が多くの割合を占めるため、長期署名技術導入によるブロックヘッダーサイズ増加によるデータ通信への影響は小さいと想定される。一方、署名データの増加はデータ通信量に影響を与えると想定している。

4.2. 実証実験の対象機能と実験環境

ビットコインのブロックチェーンに長期署名機能を適用するために、その移行方式、対象機能、および実験環境を、以下に記載する。

4.2.1. 長期署名機能の移行方式

ブロックチェーンで利用される暗号技術の危殆化に対する対応策として Compromise of underlying cryptographic algorithms[19]においてブロックチェーンに長期署名技術を適用する方法が提案されている。実現手法として、元のブロックチェーンからそのまま移行する方式（ [19]における Figure 6、以下移行方式と呼ぶ）と、サポートチェーンを利用する方式（ [19]における Figure 7、以下サポートチェーン方式と呼ぶ）の 2 通りが示されている。それぞれの方式の構成を図 4-1 に図示する。移行方式はハードフォークにより実現可能であり、サポートチェーン方式はソフトフォークにより実現可能である。

このうちサポートチェーン方式は、以下の点において実装が困難である。

(1) サポートチェーン方式では、元のブロックチェーンにおけるマイナーの報酬を表すトランザクションと、サポートチェーンにおけるマイナーの報酬を表すトランザクションが異なるため、元のブロックチェーンとサポートチェーンが同一に保てない問題がある。

(2) 現在のビットコインでは、ネットワークに対してブロックチェーンがひとつであることが前提のため、サポートチェーンを実装するためにはストレージと P2P ネットワークのレイヤーにも改修を加える必要があり、改修コストが大きくなる。

以上の理由により、本実証実験では移行方式のみを実装し評価することとした。

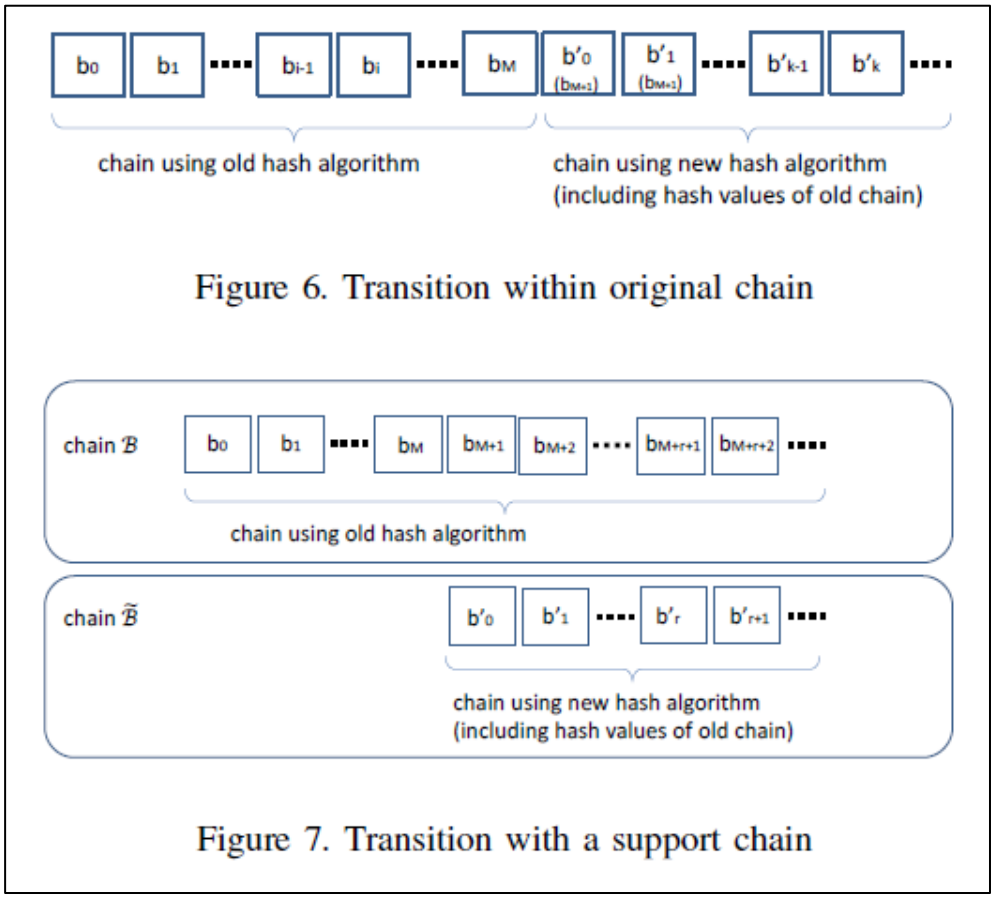


図 4-1 長期署名技術を用いたブロックチェーンの実現方法

以下、移行方式の概要を記載する。

- 上記図（Figure 6）のうち、b_0 から b_M までが、ブロックチェーンが確かに正当であることを保証するデータ（archiveHash）を付与する対象ブロック（生成済みのブロックチェーン）である。
- この対象となるブロックを先頭から s ブロックずつグループを分け、r 個のグループとする。
- r 個の各グループから計算した archiveHash を新規のブロック（b_{M+1}〜b_{M+r}）に埋め込んでいく。結果として r 個のブロックに archiveHash が付加される。
- 今回は以下のパラメータで実験を行う。
  - s=10 ブロック、r=10 グループ

4.2.2. 実証実験の実現機能

暗号アルゴリズムが危殆化した場合に備えて、ある時点における過去のブロックが正当であることを示すために、長期署名技術となる archiveHash 機能を追加する。また Proof of Work で利用されている SHA-256 の変更、およびトランザクション署名に利用されている ECDSA の鍵長の変更を行う。

① 長期署名機能（archiveHash）の計算および検証機能

長期署名機能を付与するために、ある時点におけるビットコインのブロックチェーンが valid であることを保証するためのデータをブロックヘッダーに対して付加する。付加するデータは archiveHash と呼ばれる。この機能は危殆化した暗号アルゴリズムで生成された過去のブロックをインプットとしたハッシュ計算する機能であり、そのアウトプットを新しいブロックのヘッダーに入れることで、過去のブロックが正当であることを示すことである。

② Proof of Work で用いられるハッシュ関数（SHA-256）の変更機能

ビットコインの Proof of Work では現在 SHA-256 が使用されている。SHA-256 が危殆化した場合を想定し、これを SHA-512 に変更する。これにより、SHA-256 が危殆化した場合においても、ブロックチェーンの連続性が維持できる機能を実現する。

ただし、今回の実証実験においては SHA-512 の計算結果を 256bit に切り詰める対応を実施している。ビットコインでは SHA-256 を多用な目的で利用しており、その結果の出力は 256bit 固定で保持しており、この固定値の変更による影響箇所が多岐にわたるため、本実証実験内では変更することが困難であった。

③ ECDSA の鍵長（および曲線）変更機能

ビットコインのトランザクションに付与する署名方式として、256bit の ECDSA が使用されている。256bit の ECDSA が危殆化した場合を想定し、これを 384bit の ECDSA に変更する。これにより、トランザクションの署名の改ざんを防止する機能を実現する。

ただし、ビットコインの署名アルゴリズムは内包されているが、今回の実証実験においては ECDSA の署名アルゴリズムを外部ライブラリで実装することとした。

4.2.3. 実証実験環境

ブロックチェーンの学術研究用ネットワークである BSafe.network を利用する。Bsafe,network 参加大学の各拠点に、実装したコードを配備し、フルノードを立ち上げる。ノードを立ち上げる大学は以下の 3 大学である。

- Toho University（日本）
- University of British Columbia（カナダ）
- Keio University（日本）

4.3. 実証実験の結果

今回の実証実験では、長期署名技術等を導入することによる影響を以下の 3 つの視点で測定した。

- 各ノードの実行速度への影響
- 各ノードのデータ量への影響
- トランザクションを実施する上でのネットワークの通信データ量への影響

以下に実証実験における測定結果を示す。

4.3.1. 各ノードの実行速度への影響

今回の実験では、Proof of Work に用いられる SHA-256 を SHA-512 に変更したが、その変更による処理性能への影響は小さい結果となった。また archiveHash がヘッダーに含まれるブロックに対するハッシュ計算性能も、大きく差異はない結果となった。尚、archiveHash がブロックヘッダーに含まれるケースは過去ブロックを archiveHash 化する移行時のみであり、恒久的に発生するものではない。

表 4-1 ハッシュ関数の計算時間[マイクロ秒/block]

ハッシュ関数[6]	測定回数	最小値	最大値	平均値
BlockHash	1179648	0. 846	0. 983	0. 888
BlockHashNew	1310720	0. 733	0. 983	0. 809
BlockHashArchive	917504	0. 539	1.222	1.120

一方、ECDSA の鍵長変更により署名生成速度は 10 倍程度時間を要する結果となった。1 回の署名生成速度は小さいため、支払処理のような軽量な利用では問題とならないと思慮するが、ブロック内の全トランザクションを検証するフルノードの運用では影響がある可能性がある結果となった。

表 4-2 ECDSA の計算時間[マイクロ秒]

ECDSA 鍵長	測定回数	最小値	最大値	平均値
ECDSA 256bit	24576	21.223	46.720	41.296
ECDSA 384bit	1792	282.156	598.073	575.776

4.3.2. 各ノードのデータ量への影響

今回の実験結果からは archiveHash によるブロックサイズの増加は、約 100 バイト程度の増加となった。また Proof of Work に利用されるハッシュ関数を SHA-256 から SHA-512 に変更したことによる影響は 32 バイト(256bit)増加することになる。以上のことから、長期署名移行時においてもブロックヘッダーのサイズ増加は 132 バイト程度となり、ブロックサイズ 1MB に対して、0.013%程度の増加することになる。

加えて、今回の ECDSA の鍵長を 256bit から 384bit に変更したことに伴う署名サイズの増加は、1署名あたり 50 バイト程度となった。

[6] BlockHash：古い Proof of Work ハッシュ関数でブロックのハッシュ値を計算したときの時間
BlockHashNew：新しい Proof of Work ハッシュ関数でブロックのハッシュ値を計算したときの時間
BlockHashArchive：新しい Proof of Work ハッシュ関数で archiveHash つきブロックのハッシュ値を計算したときの時間

4.3.3. トランザクションを実施する上でのネットワークの通信データ量への影響

今回の実証実験では、長期署名や鍵長の変更等、ブロックサイズに影響を与える修正を行っているものの、実験結果からは、データ量の増加分はネットワーク送受信データ量にほぼ影響しない結果となった。またトランザクションが送受信データの多くの割合（今回の実験では 96%程度がトランザクションデータ）を占めていたものの、署名アルゴリズムの鍵長変更によるデータ量の増加はネットワーク送受信データ量にほぼ影響しない結果となった。

4.4. 実証実験の考察

今回の実証実験では、暗号の危殆化対策として長期署名技術が有効であるか検証を行った。長期署名技術として、過去に生成されたブロックを新しい暗号アルゴリズムでハッシュ化する archiveHash の実装、Proof of Work で用いられる SHA-512 の実装、トランザクション署名である ECDSA の鍵長を 384bit へ変更を行い、学術ネットワーク Bsafe.network で稼動できることが確認できた。

しかしながら、実証実験においては、いくつか、実装上の制約事項等があり、4.3 で記載した実験結果データそのものを本番のビットコイン環境に適用できるわけではないことに留意する必要がある。このため、以下では実験環境であるが故の制約事項や、今回の実証実験の実装上の留意事項も踏まえて、実験結果を考察する。

またその他、実証実験を行ったことから得られた長期署名をビットコインブロックチェーンに適用するための課題もあわせて考察したい。

4.4.1. 長期署名技術等、暗号危殆化対応施策適用後のリソース負荷への影響

4.4.1.1. 実行速度への影響

今回の実験結果からは archiveHash の追加や Proof of Work のハッシュ関数変更は性能には大きく影響しない結果となった一方、ECDSA の鍵長の変更により処理時間が 10 倍程度増加した。これは現状のビットコインでは ECDSA の署名アルゴリズムは内包されているが、今回の実験では、署名アルゴリズムを外部ライブラリ化したことによる影響が強く出た結果となったと考えられる。また ECDSA の署名検証はフルノードでは最も時間を要する処理であり、性能への影響については更なる研究が必要と考える。

4.4.1.2. データ量への影響

今回の実験結果からはブロックヘッダーへの増加は大きな影響は無いと考えられる一方、署名サイズの増加によりストレージ容量増加が懸念される結果となった。署名データ容量はトランザクション数に応じて増加することになり、実際のビットコインのブロックあたりのトランザクション数はおよそ 1000 件前後あるため、単純に 1 トランザクションあたり平均して 2 つの署名があるとした場合、署名サイズの増加はブロックあたり、0.1MB 程度の増加となる。Segwit[7]対応のトランザクションは半数にも満たない状況であり、また仮に Segwit の導入が進んだとしても、ストレージに与える影響は大きくなると思慮される。

4.4.1.3. データ通信量への影響

今回の実験結果からはデータ通信への影響は小さい結果となった。しかしながら、今回の実証実験は 3 台のノードで実施したことや、特定のノードからトランザクションを流す形態となったこと、またトランザクション数も限定的であることや、トランザクションあたりの署名数も少なかったこともあり、実際のビットコインのデータ送受信とは大きく異なる点に留意する必要がある。

以上、実証実験であるが故の実装上、あるいは環境上の制約等があるが、archiveHash を用いた長期署名技術をビットコインに適用することによるリソースへの影響は少なく、運用は可能であることが示された。

[7] ビットコインのスケーラビリティ対策の 1 つの方策として 2017 年 8 月にリリースされた、各トランザクションから署名を分離し、署名領域（Witness）に保存する仕様

一方、Proof of Work で用いるハッシュ関数の変更による影響は、本実証実験からは完全に考察できない。現在のビットコインのマイニングは ASIC を用いた専用のハードウェアを使ったものであり、CPU で実験した結果と単純に比較できない。アルゴリズムの変更により既存のハードウェアはすべて無効となるので、このような変更が受け入れられるかどうかの観点から議論も必要となる。

また本実証実験では署名サイズの変更についてストレージへの影響は計測できていないが、こうしたリソースへの影響以外にも、ビットコインの取引手数料増加にも繋がる。ビットコインの取引手数料はトランザクションサイズに依存する点もあるため、現在ビットコインコミュニティ内でも議論されているマルチシグネチャの署名を小さくする技術であるシュノアー署名等、別の署名アルゴリズムの採用も含めて、様々な研究を実施した上で適用の議論がなされる必要がある。

4.4.2. 長期署名適用における運用時の課題

実証実験用コードの実装や実験結果を踏まえて、長期署名を導入したブロックチェーンの運用にあたり、以下のような考慮事項が挙げられる。

4.4.2.1. 暗号アルゴリズム移行に関する考慮事項

今回は限られたノード間での実験であるため、複数の改修を同時に実行できたが、実際には Proof of Work で用いられるハッシュ関数の変更や、archiveHash 機能等は別々にリリースされることが望ましい。特に Proof of Work で用いられる SHA-256 はビットコインでは様々な目的に多用されているので、変更においては十分な検証が必要であると思慮する。

4.4.2.2. 古い暗号アルゴリズムと新しい暗号アルゴリズムの混在による考慮事項

移行時には、古いアルゴリズムのみを対応しているクライアントと、新規アルゴリズムも対応しているクライアントが混在することが考えられる。例えば、新しい署名を検証できないクライアントでは署名の検証ができないため、入金確認できないなどの問題が発生する可能性がある。

4.4.2.3. 新規ノード立上時の考慮事項

ビットコインに長期署名が導入された後に新規ノードを立ち上げる場合、長期署名導入以前の過去のブロックは archiveHash が組み込まれたブロックを受信するまでは信用できないブロックとなる。これは時間の経過とともに、発生する可能性は大きくなるため、ブロックのダウンロード順も再設計する必要も考えられる。

4.4.2.4. SPV ノードによるトランザクション検証の考慮事項

SPV ノードではブロックヘッダーしか保存しないため、トランザクションの検証は他の古いノードを信用することになる。archiveHash の対象となるような古いトランザクションの検証には従来の Merkle tree ではなく archiveHash 内の Merkle tree も取得する必要があり、P2P プロトコルの再設計が必要である。

本実証実験を通じて、これまで理論のみで実証されていなかった長期署名技術を、archiveHash 機能として実装し、稼動確認することができた。これにより、危殆化した暗号アルゴリズムで作成された過去のブロック情報の正当性を保障する機能を、ビットコインブロックチェーンに示せたことは 1 つの成果と考えている。

ただし、この実験は、初期段階にある。上記に記したとおり、archiveHash 採用に伴い、ブロック受信順序や SPV ノードへの課題等、1 つの技術採用にあたっては考慮すべき事項が多く残されており、こうした課題の解決にあたってはビットコインコミュニティ内で技術面、経済面などを踏まえた十分な議論が行われる必要があると考えられる。従って、この問題については可能な限り早い段階で議論が開始されるべきであろう。

5. 考察

本調査研究開発においては全 19 の攻撃手法、および脆弱性についてリスク評価を実施してきた。仮想通貨取引においては wallet theft のように仮想通貨そのものが他者に奪われてしまう顕在化されたリスク、暗号の危殆化のように、仮想通貨そのものの価値が毀損され、そのことがいつ発生するかわからない潜在的なリスクがある。その他にも Block discarding や Selfish mining の攻撃等で見受けられるようにブロックを覆す、あるいは遅延させることによる利用者の仮想通貨取引トランザクションがブロックに取り込まれない、あるいは遅延するリスク等がある。

本考察においては、3.3.3.3 において述べた通り、暗号鍵の管理・運用および暗号の危殆化について検討すべき課題を明らかとした上で、今後取り組むべき方向性について検討を行いたい。

5.1. 検討すべき課題について

5.1.1. 暗号技術の危殆化について

仮想通貨で利用されている暗号アルゴリズムは常に危殆化のリスクに晒されており、特にそのリスクが突然に顕在化した場合、現状で考えられる手段としてはハードフォーク対応となると思慮される。実際に、2017 年にハッシュアルゴリズムを独自実装した一部の仮想通貨において、その独自実装に欠陥が見つかり、その対応でハードフォークをしたという事象もある。

この事例から示唆されることとして 2 つの要素がある。

ひとつは仮想通貨で利用されている暗号アルゴリズムが妥当なものであるかどうかの判断がなされているかどうか。この点については、アメリカ国立標準技術研究所（NIST）や日本においても CRYPTREC[8]が暗号技術の安全性評価を実施している。仮想通貨に採用されている暗号技術がこれらの機関で安全性評価がなされているものであるか、またその実装が正しくできているかを検証する仕組みが必要と考えられる。

もうひとつは暗号アルゴリズムというものは、常に危殆化のリスクを抱えていることから、その危殆化が発生した場合の対応策を予め準備されているかという点である。さらに 2 点目の要件としては、暗号アルゴリズムの更新が可能な実装になっていることと、暗号アルゴリズムの移行にあたっての計画がなされているかである。特に移行にあたっては、ユーザー側のウォレット対応も必要となるケースがあることに留意する必要がある。

今回の実証実験においても、ハードフォークにより暗号危殆化対応を実施した。実際にビットコインにおいても暗号アルゴリズムの移行が考慮されておらず、本格的な対応には大きな労力が必要となりうる。特にビットコインで多用されている SHA-256 の移行に関しては、修正箇所が 1000 箇所以上に及ぶ可能性があり、危殆化した場合、ビットコインのソフトウェア対応としても大きなリスクになりうる。また既に CRYPTREC からも運用監視暗号リスト（非推奨）となっている RIPEMD-160 がビットコインのアドレス生成機能として利用されており、上記の観点は特定の仮想通貨の問題ではないと言える。

いずれも既存の暗号技術に関する標準化や安全性評価を実施している機関があり、先行事例としてのベストプラクティスを参考にすることが有効であると考えられる。

暗号技術の危殆化は仮想通貨取引に関わらず、多くの金融システムに影響を及ぼす可能性があるものである。暗号技術の危殆化のリスクについて既に学術の分野、金融の分野で研究が進められており、学術会、産業界、当局、仮想通貨の開発に携わるエンジニア等の多様な連携により対応が迫られるものである。仮想通貨が暗号アルゴリズムの堅牢性に立脚した価値である以上、暗号アルゴリズム、およびその実装において、客観的な評価を行う専門的な人材の育成や専門的な機関との連携について、検討の必要がある。

[8] 総務省および経済産業省が共同で運営する暗号技術検討会と、国立研究開発法人情報通信研究機構（NICT）および独立行政法人情報処理推進機構（IPA）が共同で運営する暗号技術評価委員会および、暗号技術活用委員会で構成される。

5.1.2. 暗号鍵の管理・運用について

既に述べた通り、ビットコインをはじめ仮想通貨の取引にあたっては、利用者自身の秘密鍵の保全が極めて重要となる。秘密鍵を適切に保全できない場合、その秘密鍵に紐づく資産が失われるリスクがあり得る。ブロックチェーンという取引システムにおいては、あるアドレスから別のアドレスへ価値の移転を行う取引の結果は基本的には公開されており、そのアドレスが誰のものであるかは不明である。ただし、そのアドレスに紐づく秘密鍵を持っている利用者のみが、そのアドレス宛に移転されてきた価値は自分自身のものだと主張できる仕組みになっている。よって、仮想通貨の取引においては、秘密鍵が盗まれた場合、利用者自身がもつ仮想通貨そのものが盗まれたことと同値であると考えられる。

一方、既存の金融取引においては、仮にインターネットバンキングの ID やパスワードが盗まれた場合においても、金融機関が何らかの形で補償するケースも見受けられる。これは最終的に現金として引き出されていない場合には、金融機関が差し押さえることができるケースがあることや、また金融機関によっては、利用者保護の観点から補償しているケースもある。

しかし、仮想通貨の取引においては、盗まれた秘密鍵を使って、仮想通貨取引をブロックチェーン上のブロックに記録されてしまうと、そのブロックを取り消すことは不可能であり、既存の金融取引のように、守る手段が存在しない。[9] このリスクを利用者自身が十分に認識をしたうえで仮想通貨取引に参加することが求められる。

本リスクから示唆される秘密鍵の保全に関する論点は 2 つある。

1 つは秘密鍵の保全をどのような技術的手段で担保するべきか、またその安全性を客観的に評価する基準を明確化することである。

もう 1 つは秘密鍵の保全を誰が行うべきか、という点である。基本的には利用者自身が秘密鍵の保全を行うべきであるが、利用者の IT リテラシーの問題も考慮する必要がある。仮想通貨交換事業者等のサービス提供者は利用者の秘密鍵を預かることがあるが、すなわち、それが利用者の仮想通貨を預かっていることと同値であり、この秘密鍵を預かること自体を、いわゆる資金を預かることと同値とすべきか否かは、秘密鍵を守るための対策技術の採用状況も含めて、検討する必要がある。

特に二点目の論点は通常、情報システムに暗号技術を用いる場合は、その情報システムの運営主体が鍵管理を主体的に行うものとなるが、パブリックブロックチェーンにおいては運営主体が不在であるため、パブリックブロックチェーンを利用する側（仮想通貨取引所や一般のユーザー）が行う必要がある。

[9] 仮想通貨取引において、取り戻したケースとして、2016 年 6 月 17 日に発生した、所謂 The DAO 事件がある。これは Ethereum ブロックチェーンのスマートコントラクトの実装上のバグによる、仮想通貨（DAO Token）の流失事件である。その後、Ethereum コミュニティは Ethereum のハードフォークを行い、実質的に DAO Token を取り戻した。しかしながら、この事件により、Ethereum コミュニティは分断することになり、Ethereum は2つのブロックチェーンに分岐することに至った。その他、パブリックブロックチェーン上の1つのアプリケーションであるコントラクトの脆弱性の解決を、プラットフォーム側が対応することや、一部の利害関係者の都合でハードフォークが行われること等、ブロックチェーン業界にとっては、多くの教訓を残した事件であった。

5.2. 検討すべき施策

5.1 で示唆したウォレットの秘密鍵の保全対策と、暗号アルゴリズム危殆化に伴う対策について、現時点で考えられる具体的な施策の方向性について検討したい。なお、以下で議論する施策等については、仮想通貨取引の高度化や技術の発展などを踏まえて不断に見直す必要がある点には留意が必要である。また、さまざまな基準やガイドラインの策定を実施する主体についても慎重な検討が必要となる。既に金融システムや情報システムにおいて策定されているガイドラインを参考に今後の取り組みの方向性を検討するべきではないか。

5.2.1. 危殆化に関する施策

仮想通貨で利用されている暗号技術の安全性の評価基準を定め、適合性審査を実施する対応が考えられる。こうした適合性審査の結果は仮想通貨交換業者が取り扱う仮想通貨を登録する際の検討項目とすべきであるとともに、登録後も適合性審査の結果を踏まえて、不断に見直していくことが必要であろう。

5.2.1.1. 仮想通貨の暗号技術の安全性について評価・モニタリングを促す対応

5.2.1.1.1 安全性対策基準の策定

仮想通貨で採用されている暗号技術に関して、安全性の評価ができる基準を明確化する。このためには、「暗号技術検討委員会および関連委員会（CRYPTREC）」等により評価されている暗号技術等を参考に、FISC[10] や IPA 等の既存機関にて策定し、公表する対応も考えられる。

5.2.1.1.2 安全性の評価・審査

安全性の評価にあたっては上記安全対策基準に基づき仮想通貨交換事業者が実施する自己評価と、自主規制団体あるいは監査法人等が実施する第三者評価（監査）の両建てとしてはどうか。仮想通貨交換事業者が自己の取引所で取り扱う仮想通貨の登録申請時に、自己審査を実施する。あるいは、自主規制団体にて、予め仮想通貨交換事業者が取り扱う可能性のある仮想通貨の安全性評価を実施する方式も考えられる。こうした安全性評価を踏まえて仮想通貨交換業者が取り扱う仮想通貨を見直す必要があるかどうか業者による自己査定および検査・監督を通じて確認していく必要があると考えられる。

5.2.1.2. 危殆化が懸念される仮想通貨の暗号アルゴリズムの新方式への移行を促す対応

危殆化リスクについて、上記の審査基準を明確化し、公表することが考えらえる。危殆化リスクが高まった仮想通貨については、当局者は、交換事業者に法定通貨との交換や取扱停止などの対応を求めることを可能とする枠組みの検討も必要と考える。

暗号技術の危殆化のリスクについて既に学術の分野、金融の分野で研究が進められており、学術会、産業界、当局、仮想通貨の開発に携わるエンジニア等の多様な連携により対応が迫られるものである。仮想通貨が暗号アルゴリズムの堅牢性に立脚した価値である以上、暗号アルゴリズム、およびその実装において、客観的な評価を行う専門的な人材の育成、や専門的な機関との連携について、検討の必要がある。

また、これらの研究成果について、仮想通貨交換事業者および開発者コミュニティへの周知を図ることも重要である。

5.2.1.3. 暗号技術者の育成

以上までに述べてきたとおり仮想通貨取引を取り巻く新しい産業を育成し、更なる成長をさせていくためには、暗号技術に精通したエンジニアの存在が必要と考える。またその産業を育成していくためには既存の暗号技術を審査している機関においても、人材の増強の必要性が見込まれる。

仮想通貨の開発においては、暗号技術に精通したエンジニアの存在が大きくなっており、その仮想通貨を取り巻くビジネスにおいても暗号技術、およびその活用に長けたエンジニアの存在、育成による増強が必要と考えられる。

[10] 金融情報システムセンター(https://www.fisc.or.jp/)

5.2.2. 秘密鍵の安全性を促す対応

秘密鍵の管理は利用者自らが行う場合もあるが、現時点では多くの場合において仮想通貨交換業者が行っていると考えられる。したがって、こうした業者が顧客から仮想通貨を預かっていると見做すことが可能と考えられる。顧客の資産（秘密鍵）を預かる場合には、その重要性に鑑み、同様の機能を果たしている既存の金融機関等における対応を踏まえ、相応の安全性確保が必要となるのではないか。その際、既存のサイバーセキュリティ対策を実施することに加え、仮想通貨特有の論点について検討を進める必要がある。

例えば、ハードウェアウォレット等、ネットワークから引き離された環境で秘密鍵を保全することを促す対応が必要ではないか。また単にネットワークからの切り離しのみならず、ハードウェアウォレットのセキュリティの安全性基準の確立、ウォレットの運用態勢まで含めた対応を検討する必要があると考えられる。

以下ではハードウェアウォレットでの管理を含め、現時点で考えられるいくつかの施策について検討を深めたい。ただし、以下の対応を行えば安全性が完全に確保されるわけではなく、今後、セキュリティの専門家を交えて対策が十分であるのかを幅広く検討していくことが必要と考えられることには留意が必要である。

5.2.2.1. ハードウェアウォレットによる仮想通貨の保全を促す対応

5.2.2.1.1 ハードウェアウォレットでの管理

仮想通貨交換事業者に対して、顧客からの預かり仮想通貨の一部、あるいは全額を、耐タンパ性[11]をもつハードウェアウォレットで管理することを促す仕組みを検討する必要がある。

顧客からの預かり資産の一部をハードウェアウォレット等に移行し、ネットワークから物理的に切り離すことで外部からのハッキングを未然に防ぐことが可能となる。ただし、一部の仮想通貨においてはハードウェアウォレットが存在しない場合もあることや、利用が困難なものもある点に留意する必要がある。

5.2.2.1.2 ハードウェアウォレットの管理態勢

仮想通貨交換事業者にハードウェアウォレットによる顧客の秘密鍵保存を義務付けた場合、ハードウェアウォレットを導入するだけでは必ずしも解決が図れるものとは限らない。ハードウェアウォレットにアクセスするユースケースを洗い出した上で、ハードウェアウォレットへのアクセス権限管理やハードウェアウォレットそのもの物理的な隔離、アクセス証跡の保存等、管理態勢を明確化していく必要がある。

5.2.2.1.3 ハードウェアウォレットの安全性基準

ハードウェアウォレットの安全性を検証できる仕組みが必要である。現状、ビットコインにおいてもいくつかのハードウェアウォレットが開発、販売されているが、国際的な情報セキュリティ基準等の認定を受けていない製品もある。ハードウェアウォレットの安全対策基準を明確化すること、加えて、その実装が適切であることを評価する仕組み、および機関が必要である。

ハードウェア上で暗号鍵を保存する仕組みはかねてより Hardware Security Module（HSM）がある。この HSM の安全性については、前述の NIST が作成した「FISP-140 暗号モジュールに関するセキュリティ要件の仕様を規定する米国連邦標準規格」がある。同様に、日本においては IPA[12]が暗号モジュール試験、および認定制度を実施している。[13]

このような既に存在する暗号モジュールに関する安全対策基準を参考に、ブロックチェーンの暗号鍵管理に関する安全対策基準の作成、およびその評価の枠組みを活用することも検討すべきと考える。

なお、現状ハードウェアウォレット製品が少ないことも問題となりうる。製品が少ないということは、消費者がその製品を選択せざるを得ず、当該ハードウェアウォレットに脆弱性が見つかった場合、多くの消費者に被害が拡大する可能性がある。このような事態をさけるためにも、ハードウェアウォレットが満たすべき安全対策基準を明確化し、認定方式を定めることは、ウォレット開発企業の参入を促し、消費者に選択の多様性を提供する意味でも実施することが望まれる対応と考える。

---

[11] 内部に保存した情報を物理的に取り出すことができない

[12] 独立行政法人情報処理推進機構(https://www.ipa.go.jp/)

[13] ハードウェア、ソフトウェアに利用する暗号モジュールの試験、および認定制度である。https://www.ipa.go.jp/security/jcmvp/index.html

5.2.2.2. マルチシグニチャー技術による仮想通貨の保全を促す対応

仮想通貨交換所への攻撃による被害を受けて、複数の秘密鍵を合わせることで、1つのトランザクションに対する署名を有効とする、マルチシグという技術を用いることの重要性が指摘されている。これは、複数の秘密鍵を、保持主体を分けて管理することで、仮に1つの秘密鍵が盗まれた場合においても、仮想通貨を移動できない仕組みを導入することを促す対応である。ただし、仮想通貨においてはマルチシグを採用できない実装のものもある点に留意が必要である。

5.2.2.2.1 秘密鍵の一部を預託

マルチシグニチャーを導入した上で、一部の秘密鍵を第三者に預託できる制度を確立することも考えられる。仮想通貨交換事業者以外の第三者に秘密鍵を預けることで、秘密鍵の管理を仮想通貨交換事業者から切り離し、かつ、鍵管理のオフライン化を促す。こうした対応により、例えば仮想通貨交換所がハッキングの攻撃を受けた場合であっても、預託先も同時にハッキングをされない限り資産の引き出しを防ぐことができると考えられる。

ただし、この施策にあたっては、仮想通貨の取引を行うときに利用者が第三者に秘密鍵を利用したトランザクションへの署名を指図する等のプロセスを明確化し、そこに新たな脆弱性が発生しないか十分に検討をする必要がある。また、施策の実現にあたっては、技術面の検討のみならず、制度面の対応を行う必要がある可能性にも留意が必要と考えられる。

5.2.2.2.2 秘密鍵の預託を受けた側の遵守事項の明確化

秘密鍵の預託を受けた側の遵守事項については、仮想通貨交換事業者に課せられている管理運営体制を踏まえたものとし、既存の金融機関のように、管理プロセスが厳格化されている機関の運用を踏襲することが望ましい。

5.2.2.3. その他、仮想通貨の保全を促す対応

5.2.2.3.1 顧客預かり仮想通貨の管理方法の明示義務

顧客からの預かり仮想通貨の管理方法について明示義務を課すことも考えられる。仮想通貨毎に顧客の秘密鍵の管理方法、管理態勢を明確化し、それに基づき、当局による検査・監督によるモニタリングを実施することも想定される。この対応により顧客が仮想通貨交換事業者を安全性の面から選択できる面があるといえる。

一方、一般の顧客が適切な判断ができるか、できるようにするためには、仮想通貨における秘密鍵の重要性を一般の顧客に理解を促す施策も検討することが望ましい。

5.2.2.3.2 複数アドレスでの管理

仮想通貨交換事業者に顧客からの預かり仮想通貨を複数のアドレスで管理することを義務付ける施策も必要ではないか。顧客資産を複数のアドレスに分けて管理することで、1 つのアドレスの秘密鍵が奪われた場合においても、全顧客の仮想通貨を失うことを防ぐことができる。本施策は技術的な課題もないことから、比較的導入することは容易と考えられる。

5.2.2.3.3 利用者の慣習

仮想通貨取引において、秘密鍵を失うことが仮想通貨そのものを失うことと同値であることを利用者に改めて認識してもらう必要がある。通常、日々財布の中に多額の現金を持ち歩くことはしないことと同様に、多額の仮想通貨を1 つのウォレットで管理していることや、秘密鍵を仮想通貨交換事業者等の第三者に預けていることのリスクに対する十分な理解を促す仕組みが必要と考える。

5.3. 昨今の仮想通貨をめぐる取引の不確実性について

ここまでは、仮想通貨取引にかかる安全性について技術面を中心に調査し、対応策について検討を行ってきた。しかし、仮想通貨の取引においては、技術面のみならず、開発者やマイナー等のコミュニティへの参加者によるガバナンスの問題に端を発するリスクも存在することにも留意する必要がある。

近年、仮想通貨の法定通貨に対する交換レート（仮想通貨の価格）の高騰等を背景とし、仮想通貨の取引を行う利用者数は増加傾向である。特にビットコインにおいては利用者の増加に伴い、この数年ビットコインの取引滞留が問題となっていた（いわゆるスケーラビリティ問題）。その対応をめぐって開発者とマイナーの間で利害関係が一致せず、一方は Segwit をソフトフォークで行い、他方はハードフォークを行い、新たにビットコインキャッシュとして仮想通貨を分裂させる事態にまで発展した。その後、ビットコインから分裂し、新たな仮想通貨を生み出すハードフォークがなされる事態にまで至った。

こうした状況を受け、以下のような不確実性が生じている。なお、以下の不確実性は主にビットコインを想定して議論したものであるが、他の仮想通貨取引時にもコミュニティのガバナンスの問題にかかる不確実性が生じる可能性がある。したがって、仮想通貨の取引を行う場合には、仮想通貨の価格動向のみならず、開発者やマイナー等における取組みの動向にも注意を払う必要がある。

5.3.1. フォーク時の取引制限

SegWit 導入時において、各取引所では一時的にビットコインの売買を停止する対応を行っている。またそれに伴い、家電量販店等、ビットコインでの支払を受け付けている店舗においても、ビットコインでの支払を停止する事態となった。従前に利用者には適切にメールや HP にて案内をしており、大きな混乱には至らなかったものの、利用者が更なる拡大をしたときへの対応として今回の対応どおりで十分であるかは検討する必要がある。

5.3.2. SegWit2X 対応の中断

ブロックのサイズを 1MB から 2MB に変更する SegWit2X が提案され、2017 年 11 月にもハードフォークにより対応される予定であったが、急遽採用は中断された。中断されたため、利用者への影響は無かったと考えられるが、一方、仮想通貨交換事業者としては、本対応をめぐり自社の対応、および利用者への照会対応等、多くの労力をかけていると思慮される。こういった仮想通貨特有の意思決定プロセスの不確実性に対して、個別に仮想通貨交換事業者が対応している状況について、今後の検討する必要がある。

なお、SegWit2X はハードフォークを必要としリプレイアタック（後述）の対策がなされていないことで対応を中断したとも言われている。

5.3.3. ハードフォーク時のリプレイアタック

ハードフォークにおいて、ハードフォークをしたブロックチェーン側で特有の署名を採用していない場合に、ユーザーが意図せずともハードフォーク側のトランザクションが実行されてしまう攻撃を言う。仮に SegWit2X 対応がなされた場合、ハードフォーク元の SegWit とハードフォーク側の SegWit2X の 2 つが存在し、SegWit のトランザクションを実行すると、SegWit2X のトランザクションも実行されてしまい、利用者にとっては大きな混乱に至る可能性もあったと思慮される。

5.3.4. フォークコインの扱い

ハードフォークにより分岐したコインは原則的にはフォーク前の利用者全員に同額のハードフォーク側コインが付与されることになる。しかしながら、現状の取扱いは仮想通貨交換事業者がハードフォーク側のコインを取り扱うか否かにより、利用者への付与が決まっている。つまり仮想通貨交換事業者毎に対応が別れる事態となっており、利用者にとっては、仮想通貨交換事業者のフォークコインの対応如何により、資産が変わる事態になる。

5.3.5. 仮想通貨取引所におけるシステム障害

仮想通貨取引所に置いてシステム障害が発生したことにより、仮想通貨取引ができないケースが散見されている。仮想通貨の利用者であるユーザーが取引できないという問題もあるが、他の仮想通貨取引所にも影響をするケースもある。具体的には、仮想通貨取引価格を他の仮想通貨取引所における売買価格を基準とし、自己勘定で仮想通貨売買を行う仮想通貨取引所において、売買価格を提示している仮想通貨取引所のシステム障害により、表示されている価格が実勢と異なり、そのために、仮想通貨取引所が損害を受けたケースがある。[14]仮想通貨のボラティリティが極めて高いなかで、実勢価格と異なる取引が成立してしまう現状の取引システムは脆弱であると言える。

[14] これは実勢と異なる注文に対して取引を停止する機能が備わっていないことを示している。このような問題はかつて株式取引においても発生している。所謂、ジェイコム問題である。ある証券会社が人材サービス会社ジェイコム（現ライク）社の株式注文を「61 万円 1 株売り」とすべきところを、「1 円 61 万株売り」と注文したことで、当該株式の乱高下を招き、当該証券会社は多額の損失を招いた。

5.4. 今後さらに調査・研究を深めるべき論点について

暗号技術はこれまでの金融システムでも利用されてきているが、仮想通貨はそれ以上に暗号技術を多用しており、またそこで採用されている暗号技術の安全性評価や認定制度は無い。また仮想通貨取引における秘密鍵の安全な保管は利用者（仮想通貨交換事業者やユーザー）任せとなっているのが現状である。

このような背景を踏まえ、仮想通貨取引における安全対策基準についての調査・研究が必要と考える。

5.4.1. 仮想通貨取引における安全対策基準についての調査・研究

暗号技術の安全性評価に関しては、前述の通り、暗号技術の安全性評価はアメリカ国立標準技術研究所（NIST）や日本においても CRYPTREC が実施している。また ISO TC68（金融サービス）の下部組織 SC2 セキュリティにおいても、金融取引への暗号技術の適用に関する標準化が議論されている。いずれの組織においても、既存の情報システムへの暗号技術の適用が論点となっており、中央管理者の存在が前提となっている基準である。

一方、仮想通貨取引においては、非中央集権という管理者不在のシステムであるため、秘密鍵の安全な保管手段や、ブロックチェーン上に実装されている暗号技術が適切に取り扱われているかは、利用者（仮想通貨交換事業者や一般のユーザー）の判断に任されているのが現状である。

このような現状から、今後の調査・研究として以下のテーマが考えられる

「仮想通貨における暗号技術の安全対策基準作成に向けた調査・研究」

- 既存の暗号技術に関する安全性評価に関する調査
- 仮想通貨取引に利用されるウォレットの安全性評価、および安全対策基準作成に向けた論点整理
- 仮想通貨で利用されている暗号技術の安全性評価、および安全対策基準作成に向けた論点整理

その他、上記の活動においては、既存の安全性評価を実施している団体との連携に加え、今後組成が望まれる仮想通貨交換事業者の自主規制団体との連携も必要と考える。

5.4.2. 秘密鍵の預託制度についての調査・研究

前述の通り、仮想通貨交換事業者のシステムの脆弱性により、仮想通貨が盗まれる事態が国内外で発生している。また仮想通貨取引の利用拡大に伴い、多くの資産を仮想通貨交換事業者に預けている利用者もいる状況である。

利用者保護の観点から仮想通貨交換事業者の情報セキュリティ対策を強化することは当然のことながら、仮想通貨取引における秘密鍵の第三者への預託制度を検討するべきとの指摘も見られる。このような指摘の是非を検討するとともに、仮に第三者への預託を推奨する場合には以下のようなテーマでの調査・研究を行うことが考えられる。

「仮想通貨秘密鍵の第三者預託制度検討に向けた調査・研究」

- 預託する秘密鍵の生成、保管方法に関する調査・研究（保管者の遵守事項の洗い出し）
- 利用者からのトランザクションへの署名指図時の脆弱性に関する調査・研究
- 秘密鍵の第三者預託制度設計に向けた論点整理

6. 参考資料

6.1. 調査対象論文のリスク詳細

6.1.1. Double Spending or Race attack

6.1.1.1. 概要

攻撃者自身のビットコインを支払わずに売手から商品を得ることを目的とした攻撃。攻撃者は攻撃対象である売手に将来的に無効となるトランザクションを承認させることにより、目的を達成する。この攻撃は、ブロックの正当性を十分に確認せずにトランザクションを承認する fast payment を承認している売手 V が攻撃対象となる。

この攻撃を成功させるためには、攻撃者 A が攻撃対象である売手 V のビットコインアドレスと IP アドレスを知っている必要がある。一方、攻撃対象の秘密鍵や計算機（パソコンやモバイルデバイス）へのアクセスは必要としない。

攻撃の手順は以下の通り [15]。（図 6-1 Double spending or Race attack 概要図参照）

① A は、V への支払のトランザクション TR_V と同じ BTC を持つトランザクション TR_A を作成する。ここで、送金先は TR_V では V となっているが、TR_A では A の制御下にあるアドレスに書き換えておく。

② A は、以下の要件が満たされるように TR_V と TR_A を送信する。

(1) V は TR_A よりも先に TR_V を受信する。

(2) TR_A がブロックチェーンネットワークで承認される。したがって TR_V は棄却される。

上記要件（1）、（2）を満たすための手法として、以下の手順が考えられる。

要件（1）を満たすため、P2P ネットワークにおいて A は V の隣接ノードとして接続する [16]。さらに、A は 1 つ以上の協力者ノード H を用意し、H は V に直接接続しないようにする。このような接続環境を構築した上で、A は V に TR_V を送信し、その後 H に TR_A を送信する。これにより、V は先に TR_V を受信し、TR_V と TR_A の送信タイミングの時間差分とそれぞれが P2P ネットワーク中で V に伝播するまでの時間差分の合計の時間経過後に TR_A を受信する。

さらに、要件（2）が満たされる確率を高めるために A は（i）TR_V よりも先に TR_A を送信することや、（ii）TR_A をより早くネットワーク内に拡散させるために複数の協力者ノードを利用するといった手段を活用する。（i）について、要件（1）を同時に満たすためには、TR_V と TR_A が P2P ネットワーク中で V に伝播するまでの時間差分だけ TR_V よりも先に TR_A を送信することができる。

この攻撃は、ビットコインネットワークにおいてトランザクションがノードからノードへ伝播するまでに数秒かかるため、入力が同一で出力が異なる 2 つのトランザクションが異なる経路で拡散すると、V がその不正なトランザクションを検知するのが遅れてしまうという脆弱性を突いている。

[15] G. O. Karame, E. Androulaki, and S. Capkun, “Two Bitcoins at the Price of One? Double-spending Attacks on Fast Payments in Bitcoin”

[16] ビットコインのプロトコル仕様においては、V は内向き（外部から自分への接続）の最大接続数に達しない限り、常に他のノードからの接続要求を受け入れるため、A は V の IP アドレスを知っていれば V への直接接続を試すことができる。

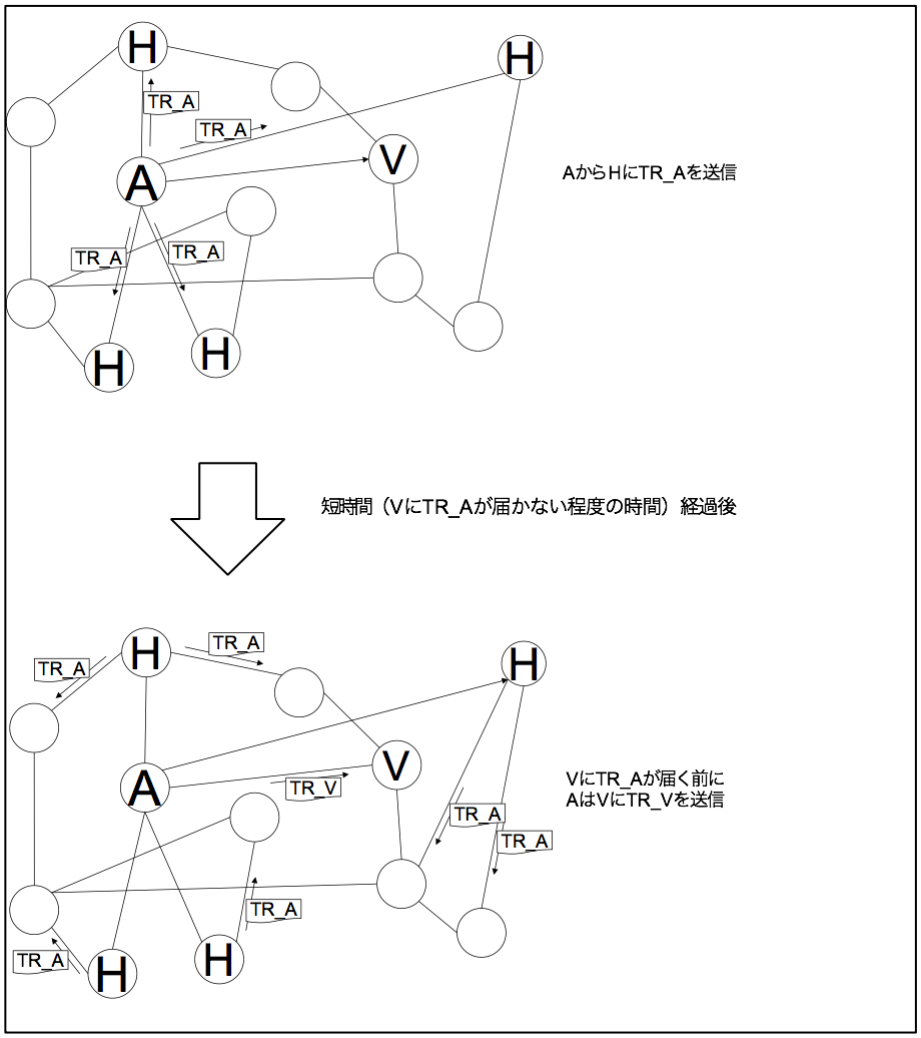


図 6-1 Double spending or Race attack 概要図

6.1.1.2. 対応策

この攻撃に対して、以下の 3 通りの対策が提案されている。

- listening period を使用

  V が A に商品を提供する前に、数秒間の listening period を設ける。V はその間に、受信した全てのトランザクションを監視し、コインが二重使用されようとしていないかを確認する。
  ただしこの方法は、A が先に TR_V を送信し listening period 以上の時間経過後に TR_A を送信することで回避される。この場合、listening period が長くなるほど、TR_V がブロックチェーンネットワークで確認され、TR_A が棄却される確率が高くなるため、A は攻撃を成功させるために H を増やさなければならない。したがって、listening period を長くするほどこの攻撃の成功確率は低下するが、顧客を待たせることになるため、利便性は低下する。この対策は技術的な改良を必要としないため、V は容易に導入可能である。

- ネットワーク内に observer ノードを導入

  受信した全てのトランザクションを V に転送する observer ノードを導入する。V は observer ノードから TR_A を受信することで、数秒以内に二重使用を検知することができる。確実に二重使用を検知するためには、多数のノードに接続する observer ノードを相当数用意する必要があり、V にとっては導入にコストがかかる。

- アラートメッセージの伝搬

  TR_V と TR_A のような、共通の入力に対し異なる出力を持つトランザクションを受信したとき、ビットコインのノードがアラートをブロードキャストするメカニズムを導入する。ビットコインには元々目的は異なるが全てのクライアントにアラートメッセージを送信する仕組みが導入されているため、V の追加コストは不要であり、A はこの対策を回避できない。A が V や V の observer ノードが TR_A を受信することを妨害しても、相当数のビットコインのノードが TR_A と TR_V の両方を受信する。これらのノードは直ちにアラートメッセージをブロードキャストするため、数秒以内にアラートメッセージが V に到達する。

6.1.1.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：3（低）
- 評価値（金融取引システムへの影響度×発生確率）：1（低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：大　・・・　誰でも攻撃対象になり得る
  - 深刻度：低　・・・　主に Fast payment での支払時に限られるため被害は少額
- 金融取引システムへの影響度
  - 影響範囲：小　・・・　ブロックチェーンのフォークによる影響は小さい
  - 深刻度：低　・・・　ブロックチェーンのフォークによる影響は小さい
- 発生確率
  - 攻撃容易性：低　・・・　Fast payment での支払時という限られた状況
  - インセンティブ：低　・・・　Fast payment での支払であるため利益は少額

6.1.2. Finney attack

6.1.2.1. 概要

二重使用攻撃の一種で、攻撃者自身のビットコインを支払わずに売手から商品を得ることを目的とした攻撃。攻撃者は攻撃対象である売手に将来的に無効となるトランザクションを承認させることにより、目的を達成する。特に売手である V が待つブロックの承認回数が 1 回以下の場合、ブロックの正当性を十分に確認せずに支払完了とするため、攻撃対象となり易い。

この攻撃を成功させるためには、攻撃者 A は事前に自分の支配下にあるノード間でのトランザクションを含むブロックをマイニングできる能力が必要である。

攻撃の手順は以下の通り[17] 。(図 6-2 Finney attack 概要図参照)

① A は事前に自分の支配下にあるノード間のトランザクション(A から A への送金でもよい)TR_AA を含むブロックをマイニングし、ブロードキャストせずに保管しておく。

② A は同じコインを用いて、V へのトランザクション TR_AV を作成する。

③ TR_AV が V に受理されるのを待つ。TR_AV はブロックチェーン B に含まれる。

④ V から商品を受け取ったら、A はあらかじめマイニングしておいたブロックをネットワークに送信し、ブロックチェーン B のフォーク B'を発生させる。

⑤ 次にマイニングされたブロックがブロックチェーン B'に繋げられれば、B'が最長になり、B は無視されるため、TR_AV は無効となる [18]。

⑥ ブロックチェーン B'に含まれる TR_AA により、A は V に支払うべきコインを取り戻す。

この攻撃は、特に少額の取引において、V が顧客サービス上、ビットコインネットワークでのブロックの承認を十分に待たずに支払を受理しなければならない場合の脆弱性を突いている。

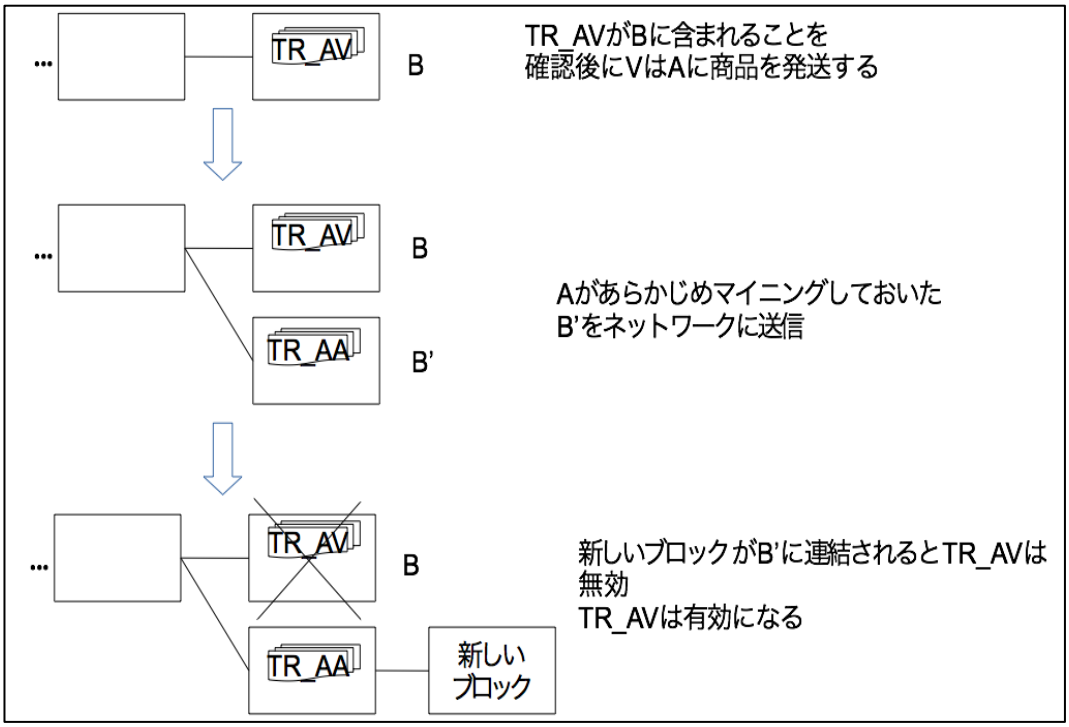


図 6-2 Finney attack 概要図

[17] M. Conti, S. Kumar E, C. Lal, S. Ruj, "A Survey on Security and Privacy Issues of Bitcoin"

[18] TR_AA も TR_AV も A 以外のマイナーから見れば有効な取引であるため、次のブロックがブロックチェーン B に繋げられるか、B' に繋げられるかは運次第。

6.1.2.2. 対応策

この攻撃に対しては、V が商品を提供する前に複数段階の承認を確認する、すなわち TR_AV が含まれるブロックチェーンBの後に複数のブロックチェーンが繋がれる状態になるまで待つことが必要である。この対策は技術的な改良を必要としないため、V は容易に導入可能である。ただし、この対策技術は完全ではなく、特に次節で述べる Finney attack の発展形により対策は回避される可能性がある。

6.1.2.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）:3（低）
- 評価値（金融取引システムへの影響度×発生確率）:1（低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲:大　・・・　誰でも攻撃対象になり得る
  - 深刻度:低　・・・　主に Fast payment での支払時に限られるため被害は少額
- 金融取引システムへの影響度
  - 影響範囲:小　・・・　ブロックチェーンのフォークによる影響は小さい
  - 深刻度:低　・・・　ブロックチェーンのフォークによる影響は小さい
- 発生確率
  - 攻撃容易性:低　・・・　Fast payment での支払時という限られた状況
かつ攻撃者にマイニング能力が必要
  - インセンティブ:低　・・・　Fast payment での支払であるため利益は少額

6.1.3. Brute force attack

6.1.3.1. 概要

Finney attack の発展形で、攻撃者自身のビットコインを支払わずに売手から商品を得ることを目的とした攻撃。Finney Attack では、攻撃者は自らに都合のよい取引を含むブロックを一つだけマイニングするが、Brute-force attack では攻撃者は次のブロック以降のブロックもマイニングすることで攻撃の成功確率を上昇させる。これにより、攻撃者は攻撃対象である売手に将来的に無効となるトランザクションを承認させることができ、目的を達成する。

この攻撃を成功させるためには、攻撃者 A は売手 V が必要とする承認の段階数分のブロックを短時間でマイニングできる十分なハッシュパワーが必要である。

攻撃の手順は以下の通り。（図 6-3 Brute-force attack 概要図参照）

① A は事前に自分の支配下にあるノード間のトランザクション（A から A への送金でもよい）TR_AA を含むブロックをマイニングし、さらにそれに続くブロックをマイニングし続け、ブロードキャストせずに保管しておく。

② A は同じコインを用いて、V へのトランザクション TR_AV を作成する。

③ TR_AV が V に受理されるのを待つ。V は受理するまでに x 回の承認を待つとすると、TR_AV が含まれるブロックの後にさらに x-1 個のブロックが繋がれる。

④ A が V から商品を受け取り、x+1 個以上のブロックを秘密裏にマイニングできるとネットワークに送信し、ブロックチェーン B のフォーク B'を発生させる。

⑤ ここで B'が最長になり、B は無視されるため、TR_AV は無効となる。

⑥ ブロックチェーン B'に含まれる TR_AA により、A は V に支払うべきコインを取り戻す。

この攻撃は、V が支払を受理するまでにビットコインネットワークで複数ブロックの承認を待つ場合でも、A が十分なハッシュパワーを有していれば、承認された A から V への送金を含むブロックをオーファンブロックとしてしまうことができるという脆弱性を突いている。

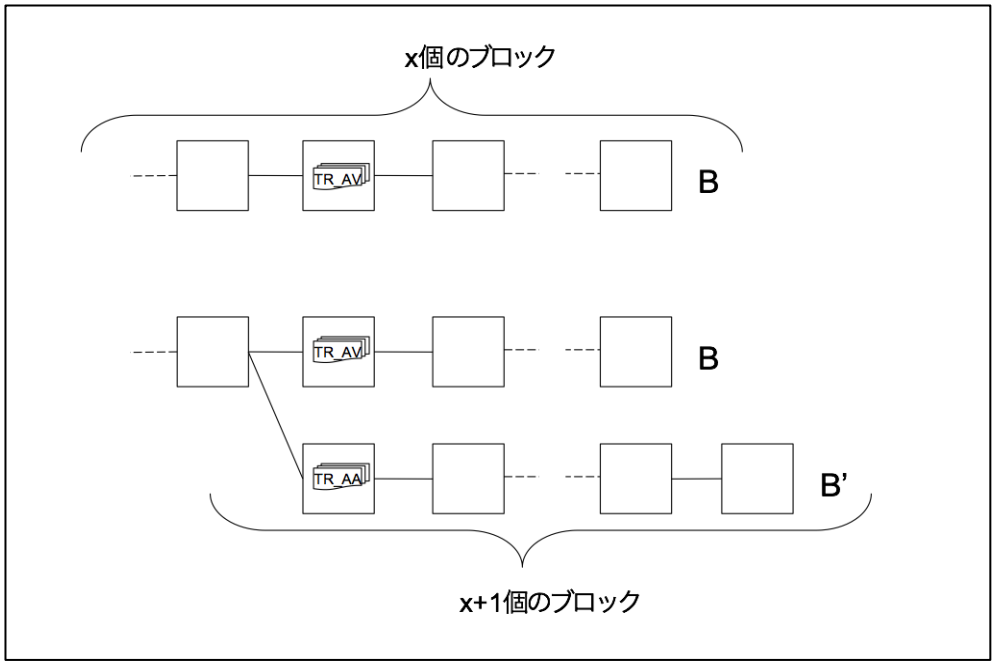


図 6-3 Brute-force attack 概要図

6.1.3.2. 対応策

この攻撃の成功確率は、A のハッシュパワーのネットワーク全体に対する割合が低いほど、また V が待つ承認の回数が多いほど、低くなる。ビットコインネットワークを単純化したモデルを使用した

理論的な計算によると、例えば A のハッシュパワーの割合が 10%で V が 6 承認を待つ場合は、攻撃の成功確率は 0.1%以下である [19]。ただし、6 承認を待つためには約 1 時間かかるため、V がこの対策を導入できる状況は限られる。

6.1.3.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：6（中）
- 評価値（金融取引システムへの影響度×発生確率）：4（中）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：大　　・・・　誰でも攻撃対象になり得る
  - 深刻度：中　　・・・　被害額は限定的任意
- 金融取引システムへの影響度
  - 影響範囲：中　　・・・　大規模フォークが起こる可能性あり
  - 深刻度：中　　・・・　大規模フォークが起こる可能性
- 発生確率
  - 攻撃容易性：低　　・・・　攻撃者に相当の能力が必要
  - インセンティブ：中　・・・　利益は限定的中程度

[19] Meni Rosenfeld, "Analysis of hashrate-based double-spending", Figure 4

6.1.4. Vector 76 or one-confirmation attack

6.1.4.1. 概要

Race attack と Finney attack の組み合わせで、無効なトランザクションを使用して攻撃対象から不正にコインを出金することを目的とした攻撃。攻撃対象は、取引所、ビットコインミキサーなど、オフチェーンでの会計サービスの提供者である。

この攻撃を成功させるためには、攻撃者 A がマイニングでき、攻撃対象のアドレスを知っていることが必要である。

攻撃の手順は以下の通り [20]。（図 6-4 Vector 76 or one-confirmation attack 概要図参照）

① 攻撃者 A は、攻撃対象 V のノードのみに接続するフルノード N_A を保持している。A はまた、1 つ以上のノードと接続しているフルノード N_B も保持している。

② A は同一のコインを使用する 2 つのトランザクションを作成する。1 つ目は V における A の口座への入金のトランザクション TR_dep で、2 つ目は A 自身のウォレットへの支払のトランザクション TR_AA である。どちらのトランザクションもこの時点ではネットワークに送信しない。

③ A は 1 つ目のトランザクションを含むようブロックをマイニングする。マイニングが完了したら、ブロックを公開する代わりに以下を同時に実行する。

- N_A に対し 1 つ目のトランザクション TR_dep を送信する。
- N_B に対し 2 つ目のトランザクション TR_AA を送信する。

④ あらかじめマイニングしたブロックを N_A に送信する。

⑤ V は A の口座への入金のトランザクションを見て、A のウォレットに該当する金額を振り込む。A はただちにその金額を引き出す。

⑥ N_A は繋がっているノードが少ないので、ネットワーク全体を見ると、大部分のノードにおいて N_B に送られたトランザクション TR_AA が受理される。そのため、結果的にネットワーク内では N_A に送られたトランザクション TR_dep は棄却される。。

この攻撃は、一承認のみでの支払を受理している攻撃対象に対して直接接続し二重使用のトランザクションを含むブロックを送信した場合、攻撃対象はそのブロックが不正であることを直ちに検知できないという脆弱性を突いている。

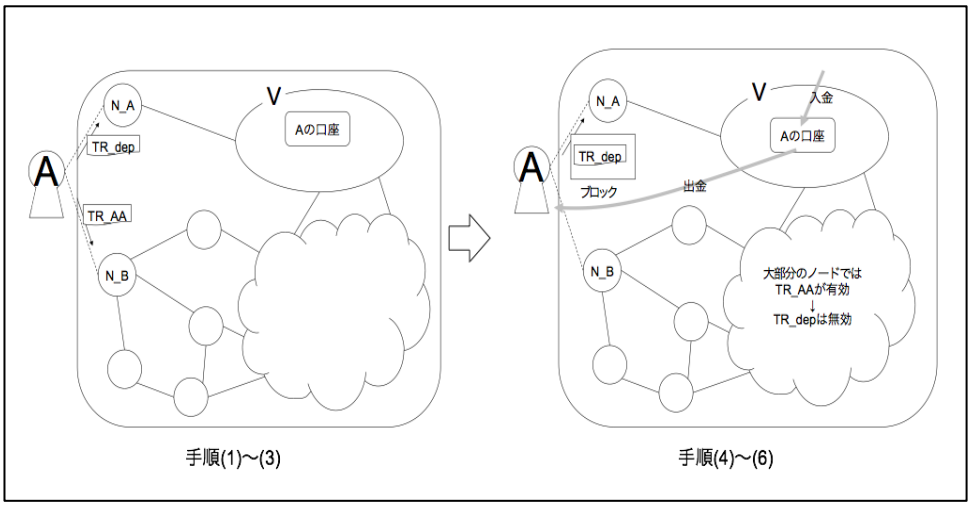


図 6-4 Vector 76 or one-confirmation attack 概要図

[20] sgornick, "Vector76 Double Spend Attack?" https://www.reddit.com/r/Bitcoin/comments/2e7bfa/vector76_double_spend_attack/（後半、The attack would be carried out as follows. 以降）

6.1.4.2. 対応策

この攻撃に対して、以下の対策が提案されている。

- 1 承認のみで支払を受理しない。ただし、2 承認以上を待つためには約 20 分以上かかるため、V がこの対策を導入できる状況は限られる。
- 内向きに接続されたノード[21]からのトランザクションは使用しない。この対策はビットコインクライアントの改良が必要となるが、導入は容易であると考えられる。
- 静的 IP アドレスを使用しない。この対策は V のネットワーク設定を変更するだけで可能であるため、導入は容易であると考えられる。

6.1.4.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）:62（中低）
- 評価値（金融取引システムへの影響度×発生確率）:42（中低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲:小 ・・・ 攻撃対象は取引所やビットコインミキサーなど、オフチェーンでの会計サービスの提供者
  - 深刻度:小 ・・・ 攻撃対象は取引所やビットコインミキサーなど、オフチェーンでの会計サービスの提供者
- 金融取引システムへの影響度
  - 影響範囲:小 ・・・ 攻撃対象となる取引所の数が少ない
  - 深刻度:低 ・・・ システム自体への被害が少ない
- 発生確率
  - 攻撃容易性:低 ・・・ 攻撃者に相当の能力が必要
  - インセンティブ:中 ・・・ 利益は限定的

[21] V から見て内向きに接続されたノードは N_A のように A が V に直接接続したノードである可能性がある。

6.1.5. >50% hashpower or Goldfinger（Majority attack）

6.1.5.1. 概要

ネットワーク全体のハッシュパワーの過半数を持つことで、コインの二重使用やマイニングの独占等を目的とした攻撃。攻撃対象は、マイナー、取引所、ユーザーなどブロックチェーンネットワーク全体である。

攻撃者 A はネットワーク全体のハッシュパワーの過半数を手に入れる必要がある。そのための方法として、マイニングプールの寡占や結託、政府や大企業による大規模な資本投入が考えられる。A がハッシュパワーの過半数を手に入れると、不正なトランザクションを含むブロックの後に多数のブロックをつなげることで二重送金や正常な取引の妨害、また、マイニングを独占し報酬を全て入手することが可能となる。

この攻撃は、ビットコイン取引の合意形成アルゴリズムがハッシュパワーの多数決をもとにしているという点を突いている。

6.1.5.2. 対応策

ブロックチェーンが Proof of Work のルールを採用している限り、この攻撃に対する対策はないと考えられている。ただし、ビットコインでは各ブロックがどのマイナーやマイニングプールによってマイニングされたのかが公開されているため、この攻撃が行われた場合、直ちに一般に知られることとなる。その結果、ビットコインのシステムの信頼性が低下し、ビットコインの価値が暴落し、A にとっても不利益となるため、A がビットコインネットワークの破壊を望む以外に攻撃を行う動機はほぼないと考えられている。

また、Proof of Work の代わりに Proof of Stake と呼ばれるアルゴリズムを用いることで、この攻撃のリスクがさらに軽減されると考えられている。Proof of Work はハッシュパワーが大きいほどブロック承認の成功率が高くなるのに対して、Proof of Stake はコインの保有量が多いほど、また保有期間が長いほどブロック承認の成功率が高くなる。したがって、攻撃の成功確率を上げるためには多くのコインを保有しなければならないため、コストがかかる。また、攻撃が成功しても、保有しているコインの価値が低下してしまう。以上のことから、Proof of Stake はこの攻撃に対して頑健なアルゴリズムであると考えられている[22] 。

6.1.5.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：3（低）
- 評価値（金融取引システムへの影響度×発生確率）：3（低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：大　・・・　誰でも攻撃対象になり得る
  - 深刻度：中　・・・　被害額は限定的任意
- 金融取引システムへの影響度
  - 影響範囲：大　・・・　（Goldfinger attack では特に）システムそのものを無効にする可能性
  - 深刻度：大　・・・　（Goldfinger attack では特に）システムそのものを無効にする可能性
- 発生確率
  - 攻撃容易性：低　・・・　ハッシュパワーを半分以上入手するためには、攻撃者に相当のリソースが必要
  - インセンティブ：低　・・・　攻撃が露見するとビットコインの価値が低下するため、攻撃者の利益は縮小

[22] Proof of Stake はコインの保有量が多いほど、また保有期間が長いほどブロック承認の成功率が高まるため、通貨を溜め込む人が多くなり流動性が損なわれる、また、ハッシュパワーを必要とせず通貨を保管しておくだけで報酬が得られるため、最長のブロックチェーンにブロックを連結するインセンティブが低くなりブロックチェーンの収束が起こりにくいということや、初期ユーザーがより多くのコインを手に入れられる仕組みであるため、不公平であるという問題点もある。

6.1.6. Block discarding or Selfish mining

6.1.6.1. 概要

一般のマイナーやマイニングプールの計算資源を無駄にすることで、攻撃者によるブロックの検証結果が採用され易くし、通常よりも多くの報酬を得ることを目的とした攻撃。一般のマイナーには公開しないマイニング済みブロック(*1)を確保しておき、一般のマイナーがブロックをマイニングした直後などに、*1 を公開することで、自分がマイニングしたブロックの採用確率を高める攻撃である。

攻撃対象は一般のマイナーやマイニングプールである。

攻撃者 A の計算能力が大きいほど通常よりも多く利益を得る確率が大きくなるため、A は通常単一のノードではなく、マイニングプールである。

攻撃の手順は以下の通りである。(図 6-5 Block discarding or Selfish mining 概要図参照)

A は一般のマイナーやマイニングプールが使用する公開ブロックチェーンとは別に、秘密のブロックチェーンを保持し、常に秘密のブロックチェーンに対してマイニングを行い、自分が発見したブロックを直ちに公開せずに秘密にしておく。A は下記の通り、ネットワークの各状態に応じて戦略を決定する。

状態 0:初期状態で、A の秘密ブロックチェーンは公開ブロックチェーンに等しい。ネットワーク全体に対する A のハッシュパワーの割合を□とすると、以下の 2 通りの事象が起こり得る。

- 確率 α で A がブロックを発見し、状態 1(A の秘密ブロックチェーンが 1 ブロック分長い状態)に遷移する。
- 確率 1-α で他のマイナーがブロックを発見し、公開ブロックチェーンが 1 ブロック分長くなると、A は公開ブロックチェーンを自分の秘密ブロックチェーンとしてコピーする。

状態 1:A の秘密ブロックチェーンが 1 ブロック分長い状態である。ここでは以下の 2 通りの事象が起こり得る。

- 確率 α で A がブロックを発見し、状態 2(A の秘密ブロックチェーンが 2 ブロック分長い状態)に遷移する。
- 確率 1-α で他のマイナーがブロックを発見し、公開ブロックチェーンの長さが A の秘密のブロックチェーンの長さと等しくなると、状態 0'に遷移する。

状態 0':A は直ちに秘密ブロックチェーンを公開する。したがって、公開ブロックチェーンに公開のブロックと A のブロックの1ブロック分のフォークが発生した状態である。この時、競合する 2 つのブロックのうち、A のブロックに対してマイニングを行うノードの合計のハッシュパワーの全体に対する割合を γ とすると、ここでは以下の3つの事象が起こり得て、いずれの場合もシステムは状態 0 に戻る。

- 確率 α で A がブロックを発見すると、A はさらにそのブロックを公開する。ここで公開ブロックチェーンにおいて、A が発見した 2 ブロックのフォークの方が長くなるため、それらのブロックが採用され、A は 2 ブロック分の報酬を得る。
- 確率 γ (1-α)で他のマイナーが A のブロックに続くブロックを発見し、他のマイナーと A はそれぞれ 1 ブロック分の報酬を得る。
- 確率(1-γ)(1-α)で他のマイナーが公開のブロックに続くブロックを発見し、他のマイナーは 2 ブロック分の報酬を得る。

状態 2:A の秘密ブロックチェーンが 2 ブロック分長い状態である。ここでは以下の 2 通りの事象が起こり得る。

- 確率 α で A がブロックを発見し、状態 3(A の秘密ブロックチェーンが 3 ブロック分長い状態)に遷移する。
- 確率 1-α で他のマイナーがブロックを発見し、公開ブロックチェーンの長さが A の秘密のブロックチーンよりも 1 ブロック分短くなると、A は秘密の 2 ブロックを公開する。A が公開した 2 ブロックが採用されるため、A は 2 ブロック分の報酬を得る。システムは状態 0 に遷移する。

状態 n(n>2):A の秘密ブロックチェーンが n ブロック分長い状態である。ここでは以下の 2 通りの事象が起こり得る。

- 確率 α で A がブロックを発見し、状態 n+1(A の秘密ブロックチェーンが 3 ブロック分長い状態)に遷移する。A は 1 ブロック分の報酬を得る。
- 確率 1- α で他のマイナーがブロックを発見すると、システムは状態 n-1 に遷移する。

この攻撃は、ビットコインにおいてマイナーが発見した正当なブロックを直ちに公表せずに隠し持っておくことができるという脆弱性を突いている。

参考:
https://bitcoinmagazine.com/articles/selfish-mining-a-25-attack-against-the-bitcoin-network-1383578440/

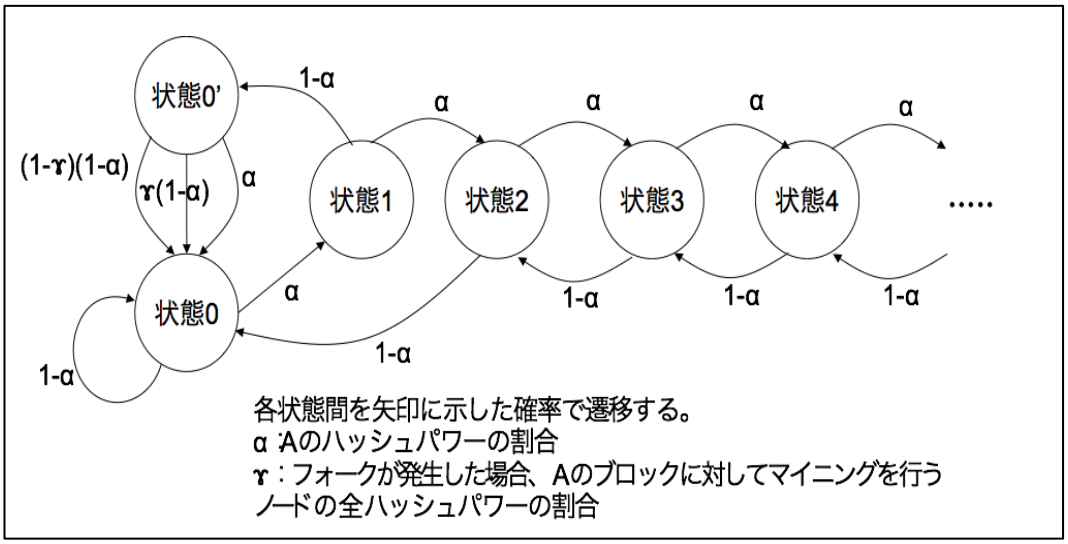


図 6-5 Block discarding or Selfish mining 概要図

6.1.6.2. 対応策

γ =1 の場合、α がどのような値であっても、A は上記の戦略をとることにより通常よりも多く報酬を得ることができる。γ=0 の場合は、A が通常よりも多く報酬を得るためには α は 1/3 以上でなければならず [23]、攻撃の難易度が高くなるが、公開ブロックチェーンにフォークが発生した場合に一般のマイナーはどちらが A によるブロックなのか識別できないため、γ=0 とすることは難しい。現実的に可能な対策として提案されている案は、一般のマイナーが競合する 2 つのブロックを受信した場合に、それらの両方を拡散し、どちらのブロックに対してマイニングするかを一様ランダムに選択するというアルゴリズムに変更するということである。これにより γ=1/2 となり、A が通常よりも多く報酬を得るためには γ は 1/4 以上でなければならなくなる 9。この対策はマイニングのアルゴリズムを変更するのみであるため、導入は比較的容易であると考えられる。

[23] I. Eyal and E. G. Sirer,“Majority is not enough: Bitcoin mining is vulnerable”, Fig 3 参照

6.1.6.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：2（低）
- 評価値（金融取引システムへの影響度×発生確率）：4（中）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：小　・・・　マイニングプール内の正直なマイナーが対象
  - 深刻度：小　・・・　マイニングプール内の正直なマイナーマイニングプール内の正直なマイナーが対象で一般の利用者への影響はないが対象
- 金融取引システムへの影響度
  - 影響範囲：小　・・・　マイニングは成功するのでシステム自体に影響はない
  - 深刻度：中　・・・　本来マイニング成功で得られる報酬が得られない
- 発生確率
  - 攻撃容易性：低　・・・　攻撃者にマイニング能力が必要
  - インセンティブ：中　・・・　マイニング成功で得られる報酬は限定的中程度

6.1.7. Block withholding

6.1.7.1. 概要

マイニングして発見したブロックを隠し持っておくことで、マイニングプールに損害を与える、不当に利益を得る、等を目的とした攻撃。攻撃対象はマイニングプールの運営者や参加者。攻撃に必要とされる条件としては、攻撃者 A はマイニングプールに参加している必要がある。また、A のハッシュパワーが大きいほど攻撃の影響が大きい。

この攻撃はその手順により”Sabotage”と”Lie in wait”の二種類に分けられる。

Sabotage の攻撃手順は、単純に A はマイニングしたブロックをマイニングプールの運営者に提出しない、というものである。これにより、マイニングプールの報酬方式が PPLNS 型（Pay Per Last N Shares）[24] である場合は、マイニングプールの運営者には損害がないが、Aも含めマイニングプールの参加者の報酬は A のハッシュパワーのプール全体に対する割合の分だけ減少する。また、マイニングプールの報酬方式が PPS 型（Pay Per Share）[25]である場合は、ブロックが発見されなくてもマイニングの貢献度に応じて各参加者に報酬が支払われるため、運営者が損害を被る。

これに対して、Lie in wait は A が利益を得ることができる戦略である。攻撃の手順としては、まず A は PPLNS 型の報酬方式を採用している複数箇所のマイニングプールで同時にマイニングを行う。そしてあるマイニングプールでブロックを発見した場合、そのブロックは隠し持っておく。また、そのマイニングプール以外でのマイニングを中止し、そのマイニングプールに全てのハッシュパワーを集中させる。ブロック発見からある時間 T 経過後にブロックを提出することで、本来は他のマイニングプールに割り当てていたハッシュパワーの分だけ得られる報酬が増加する。ただし、T が長すぎると、マイニングプール内の他の参加者が先にブロックを発見する確率が高くなり、得られる報酬の期待値は低下する。得られる報酬の期待値が最大となる T は、ブロックを発見する平均時間を T_0、A がマイニングするマイニングプールの数を m とすると、$T=(m-1)/(2m-1)T_0$ である[26] 。

この攻撃は、ビットコインにおいてマイナーが発見した正当なブロックを直ちに公表せずに隠し持っておくことができるという脆弱性を突いている。

6.1.7.2. 対応策

この攻撃に対して、マイニングプールにおいてマイナーが発見したブロックが、マイニングプールの運営者に提出されるまでは有効かどうかが識別されない（マイナーがブロックを隠し持つ判断ができなくなる）ように、ビットコインのプロトコルを改良する下記の技術が提案されている。

① 各ブロックに SecretSeed, ExtraHash, SecretHash の 3 つのフィールドを追加する。

② ExtraHash は SecretSeed のハッシュ値とする。

③ ExtraHash はブロックヘッダーの一部とし、ブロックハッシュ計算に使われるフィールドの一つとする。

④ SecretHash はブロックハッシュと SecretSeed の連結のハッシュ値とする。

⑤ ブロックが有効であるための要件を、ブロックハッシュが 2256/（232D）以下であること[27]から、ブロックハッシュが 2256/232 以下かつ SecretHash が 2256/D 以下であることに変更する[28]（ここで D は難易度を表す）。

⑥ プール運営者は SecretSeed を選び、これを秘密に保つ。プール運営者は ExtraHash を計算し、他のフィールドと共にマイナーに提供する。マイナーはブロックハッシュを計算し、2256/232 以下かどうか確認し、そうであればシェアとして提出する。マイナーはこれが有効なブロックかどうか分からない。SecretSeed を知っているプール運営者は SecretHash を計算し、これが 2256/D 以下なら有効なブロックとしてネットワークに送信する。

[24] ブロック発見前の一定の計算量に対するプール参加者の貢献度に応じて報酬を決定する方式
[25] 単純にマイナーのハッシュパワーとマイニングした時間に応じて報酬を決定する方式
[26] M. Rosenfeld, “Analysis of bitcoin pooled mining reward systems”, 6.2.2 章参照
[27] ブロックのハッシュ値は 256 ビットで、難易度 D=1 の時に 2^{32} 回のハッシュ計算に 1 回の割合でブロックがマイニングされるよう設定されているため。
[28] 変更前と変更後でマイニングプールがブロックをマイニングする頻度は変わらない。

この対策はビットコインのソフトフォークが必要であり導入の難易度は比較的高いと考えられる。

6.1.7.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：42（低中）
- 評価値（金融取引システムへの影響度×発生確率）：4（中）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：小　・・・　マイニングプール内の正直なマイナーや
マイニングプールの運営者が対象
  - 深刻度：低　・・・　マイニングプール内の正直なマイナーや
マイニングプールの運営者が対象で一般の利用者への影響はない
- 金融取引システムへの影響度
  - 影響範囲：小　・・・　マイニングは成功するのでシステム自体に影響はない
  - 深刻度：中　・・・　本来マイニング成功で得られる報酬が得られない
- 発生確率
  - 攻撃容易性：低　・・・　攻撃者にマイニング能力が必要
  - インセンティブ：中　・・・　マイニング成功で得られる報酬は限定的

6.1.8. Bribery attacks

6.1.8.1. 概要

短期的に大量の計算資源を手に入れ、二重使用や block withholding attack を成功させることを目的とした攻撃。攻撃対象は一般のマイナーや売手である。

攻撃に必要とされる条件としては、攻撃者 A はマイナーに賄賂を渡す必要があるため、そのための資金が必要である。

攻撃の手順は、下記の 3 通りの方法でマイナーに賄賂を渡し、短期的に大量の計算資源を入手し、二重使用や block withholding attack の攻撃を行う。

① 仮想通貨または法定通貨により直接賄賂を支払う。これはクラウドマイニング業者を利用することで、簡単に実現できる。

② 報酬を市場より高めに設定したプールを作成し、マイナーを誘致する。

③ 攻撃者が延長したいブロックチェーンのフォークに対し、賄賂を設定してマイニングさせる。これは、例えば単純に手数料が高いトランザクションを A が延長したいブロックチェーンのフォークにブロードキャストすることで実現することができる。

この攻撃は、ビットコイン取引の合意形成アルゴリズムがハッシュパワーの多数決をもとにしているという点を突いている。

6.1.8.2. 対応策

この攻撃を防ぐためには、ブロックのマイニングの報酬を、そのブロックに含まれる全トランザクションの合計以上にしなければならないが、これは現実的ではない。

対策技術の代わりに、この攻撃の可能性が軽減される以下の要因が挙げられている。

- マイナーが賄賂やより高い報酬を提示されても計算資源の貸与/プールの変更ができない、またはする意思がないといった可能性や、賄賂が設定されたブロックチェーンのフォークに気がつかないといった可能性がある。ただし、マイナーが経済合理的に行動し、技術的に高度になるほど、自身にとってより利益のある行動をとる傾向があるため、この可能性は低くなる（したがって賄賂を受け取る可能性が上がる）。
- A がこの攻撃で利益を得るためには大きな額のトランザクションを作成する必要があり、そのための資金が必要である。また、攻撃が失敗した場合、必ずしもこのトランザクションを失うことはないが、賄賂は返却されない。
- 二重使用により商品を手に入れようとする場合、返品させられる可能性がある。また、二重使用を行う場合に取引相手に手数料[29]を支払う必要がある可能性があるか、または賄賂の相対費用を無視できる程度に二重使用による利益を得ることができない可能性がある。
- トランザクションの金額が大きくなるほど、受取人はより多くのブロックの承認を要求すると考えられ、A はより多くの賄賂が必要となる。ただし、A は少額のトランザクションを多数作成し、それらの全てを二重使用しようとすると考えられるため、この要因は攻撃リスクを大きく軽減させるものではないと考えられる。
- 攻撃対象者が、攻撃を防ぐために、逆にマイナーに賄賂を渡す可能性がある（賄賂合戦になる可能性）。
- この攻撃が行われようとしている場合、その意図は他のマイナーにすぐに伝わり、もしも攻撃が成功した場合はビットコインの価値は低下する。したがって、マイナーが賄賂を受け取り、短期的な利益とはなっても、長期的には不利益となるため、賄賂を受け取る可能性は低い。

[29] マイナーに支払う手数料ではなく、ビットコイン外の一般的な取引手数料。

6.1.8.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：4（中）
- 評価値（金融取引システムへの影響度×発生確率）：4（中）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：中　　・・・　マイナーや売手が攻撃対象
  - 深刻度：中　　・・・　二重使用（2.4.1～2.4.3 の攻撃）や block withholding attack と同程度
- 金融取引システムへの影響度
  - 影響範囲：中　　・・・　二重使用によりフォークが発生する可能性もある
  - 深刻度：中　　・・・　二重使用によりフォークが発生する可能性もある
- 発生確率
  - 攻撃容易性：低　　・・・　攻撃者に資金力が必要
  - インセンティブ：中　・・・　二重使用（2.4.1～2.4.3 の攻撃）や block withholding attack と同程度

6.1.9. Refund attacks

6.1.9.1. 概要

払戻を活用して、取引履歴を隠し不正な利益を得ることを目的とした攻撃。この攻撃の種類として、BIP70（Bitcoin Improvement Proposal 70）における認証の脆弱性に着目した Silkroad attack と、既存の支払処理の払戻規約を悪用した Marketplace Trader attack の 2 種類が考案されている 。[30]

攻撃に必要とされる条件は特にない。それぞれの攻撃の手順は下記の通りである。

6.1.9.1.1 Silkroad attack:（図 6-6 Silkroad attack 概要図参照）

この攻撃は、例えば違法な商品を業者 T から購入したい顧客が、第三者である売手 V を利用して自分の購入履歴を隠蔽することを目的としたものである。ここで、攻撃者は顧客 A で、攻撃対象は売手 V である。

① 顧客 A が、業者 T のウェブサイトから（T のアドレス AT、金額 B、T の公開鍵 σ T を含む）支払要求のメッセージをダウンロードする。

② A は違法な商品と同額かそれ以上の商品を販売している売手 V を探し、その商品の購入手続きを行い、（V のアドレス AV、金額 B、V の公開鍵 σ V を含む）支払要求メッセージをダウンロードする。

③ A のウォレットで支払のトランザクションを承認し、T の支払アドレス AT を払戻アドレスとして支払要求メッセージに追加する。

④ A が V から支払の承認メッセージを受信した後、A は注文をキャンセルし V に払戻を要求する。

⑤ V が BIP70 にしたがっている場合、払い戻されたコインは T に送信される[31]。

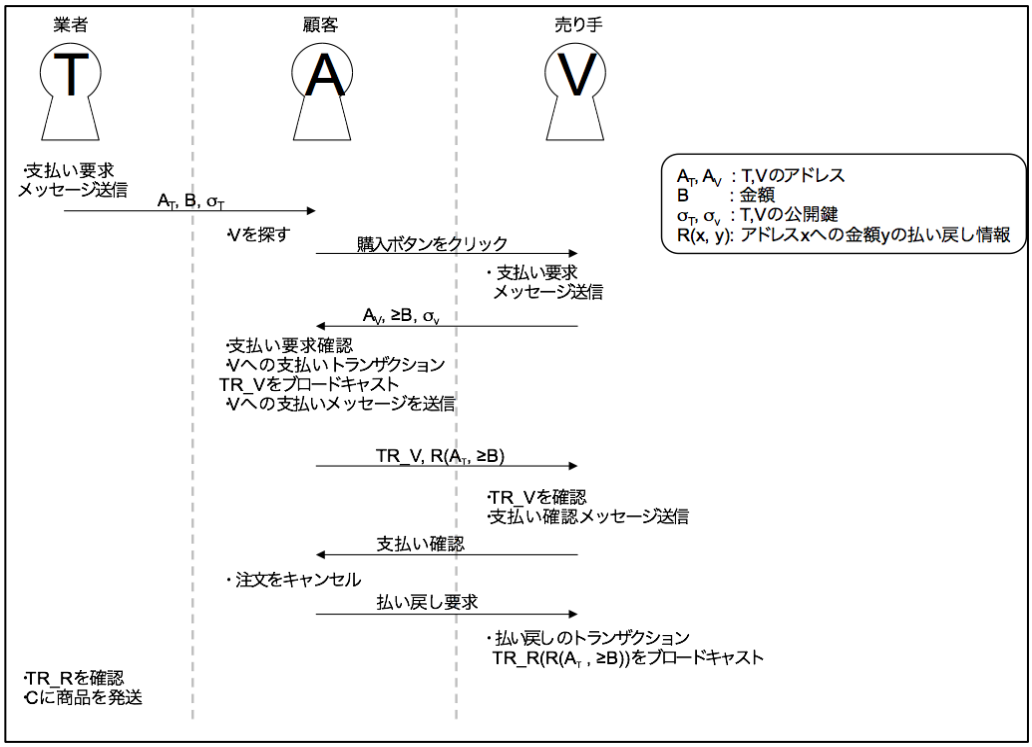


図 6-6 Silkroad attack 概要図

[30] P. McCorry, S. F. Shahandashti, and F. Hao, “Refund attacks on Bitcoin's Payment protocol”

[31] BIP70 で導入された支払いプロトコルでは、支払い時に顧客から売り手に払い戻し用のビットコインアドレスが通知されるが、売り手はそのアドレスが本当に顧客のものかどうか確認することができないため。

6.1.9.1.2 Marketplace Trader attack:（図 6-7 Marketplace Trader attack 概要図参照）

この攻撃では、攻撃者である業者 A が最新の製品などを安く販売する Web サイトを構築し、その支払方法は大手の信頼できる小売業者を通じて行われると宣伝することで顧客を安心させる。顧客が騙されてその Web サイトで購入した後に、A がその注文をキャンセルし、顧客が支払ったビットコインを小売業者から A に払戻しさせることで利益を得る。ここで、攻撃者は業者 A で、攻撃対象は顧客 V である。

① 違法な業者 A は、商品を市価より安く販売する Web サイトを構築し、CeX のような信頼できる小売業者 T を通じた取引を行うことで、自身を信頼できる業者のように装う。

② 顧客 V が A の商品の購入を決め、A の Web サイトで支払ボタンをクリックすると、A は T の Web サイトから（T のアドレス AT、金額 B、T の公開鍵 σ_T を含む）支払要求メッセージを自動的に取得し、V に転送する。

③ V のウォレットでは正規の支払メッセージに、支払額と共に信頼できる T の名前が表示されるため、V は A を信用し、T へ支払メッセージを送信し T への支払トランザクション TR_T をブロードキャストする。

④ A の Web サイトはネットワーク内の支払トランザクション TR_T を特定すると、V の Web ブラウザを更新して偽の確認ページを表示する。

⑤ A は T に対して V の注文がキャンセルとなったことを連絡し、T にメールで払戻のアドレス AA と金額 B を送信する。

⑥ メールによる認証を許可する規約により、A の払戻のアドレス A_T にコイン฿が送信される。

この攻撃は、ビットコインでの取引において払戻を行う際に、第三者が商品の購入者を装い払戻を受けることができるという点を突いている。

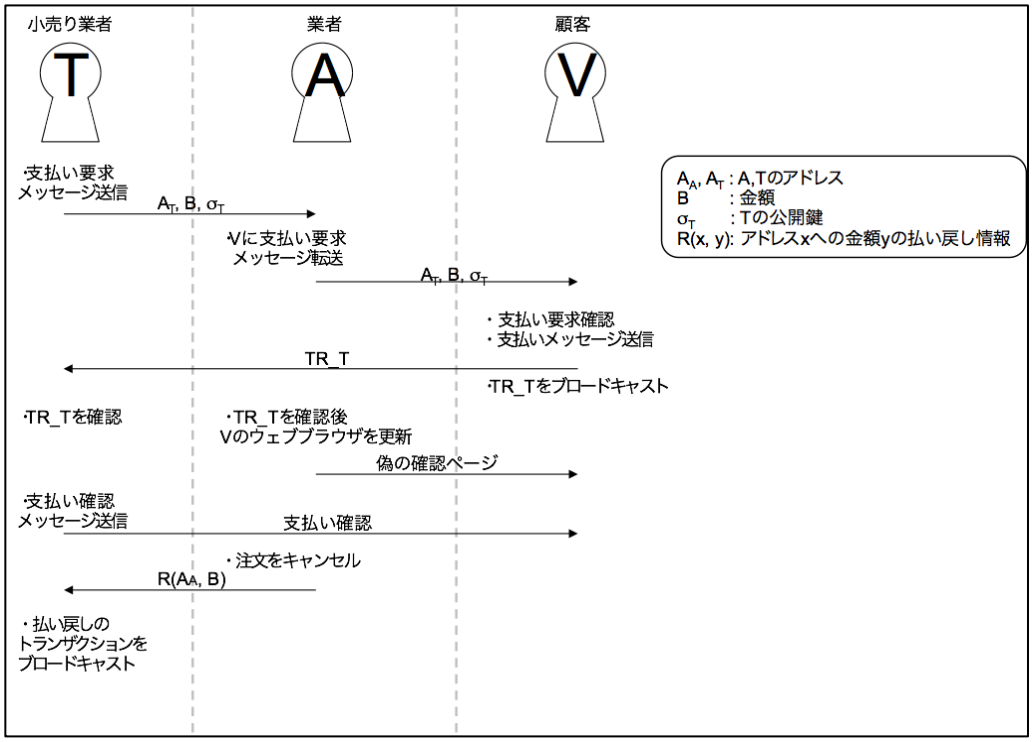


図 6-7 Marketplace Trader attack 概要図

6.1.9.2. 対応策

この攻撃に対して、売手に、受信した払戻のアドレスが、支払を承認した買手と同一人物によって承認されたことを暗号学的に証明できる公式に検証可能な証拠を提供することが提案されている。買手がトランザクションを認証したそれぞれの鍵で、買手自身の払戻のアドレスを承認するようにすることで、この攻撃を防止する。

この対策の導入は比較的容易であると考えられ、一部の取引所で既に実装されている。

6.1.9.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：6（中）
- 評価値（金融取引システムへの影響度×発生確率）：2（低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：大　　・・・　誰でも攻撃対象になり得る
  - 深刻度：中　　・・・　被害額は限定的
- 金融取引システムへの影響度
  - 影響範囲：小　　・・・　システム自体には影響なし
  - 深刻度：低　　・・・　システム自体には影響なし
- 発生確率
  - 攻撃容易性：低　　・・・　BIP70 という限られたプロトコルが対象
  - インセンティブ：中　・・・　利益は限定的

6.1.10. Punitive and Feather forking

6.1.10.1. 概要

例えば資産を凍結するために、攻撃対象のトランザクションをブラックリストに登録することを目的とした攻撃。攻撃対象は顧客や売手など、一般のユーザー。

攻撃に必要とされる条件としては、攻撃者 A はマイニングの能力を有している必要がある。また、A のハッシュパワーが大きいほど攻撃の影響が大きい。

攻撃の手順は、攻撃者 A がブラックリストに登録したい攻撃対象者からのトランザクションを含むブロックチェーンに対してマイニングをせずに、フォークを発生させそのフォークに対してマイニングを行うことと、そのような行動を取るということを公言することである。A のハッシュパワーが小さい場合、A が単独で攻撃を行ってもこの攻撃が成功する確率は低い。ただし、一定のハッシュパワーを有している A がこの攻撃を行うことを公言すると、他のマイナーはメインのブロックチェーンに対してマイニングしても報酬が得られない可能性が大きくなることが分かるため、A が発生させたフォークに対してマイニングするマイナーが増加し、この攻撃が成功する確率は高くなる。

この攻撃は、ビットコイン取引の合意形成アルゴリズムがハッシュパワーの多数決をもとにしているという点を突いている。

6.1.10.2. 対応策

この攻撃に対する対策技術は提案されていない。

6.1.10.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：3（低）
- 評価値（金融取引システムへの影響度×発生確率）：1（低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：大　　・・・　誰でも攻撃対象になり得る
  - 深刻度：高　　・・・　利用者の仮想通貨が永久に使用できなくなる
- 金融取引システムへの影響度
  - 影響範囲：小　　・・・　システム自体は持続させる
  - 深刻度：低　　・・・　システム自体は持続させる
- 発生確率
  - 攻撃容易性：低　　・・・　攻撃者にマイニング能力が必要
  - インセンティブ：低　・・・　攻撃者にとって直接の利益はない

6.1.11. Wallet theft

6.1.11.1. 概要

ユーザーのウォレットに侵入することで、不正に仮想通貨を取得することを目的とした攻撃。

攻撃対象は仮想通貨のユーザー。

攻撃者 A は何らかの手段を用いてユーザーの秘密鍵またはウェブウォレット等のログイン ID、パスワードを入手する必要がある。

ウォレットはホットウォレットとコールドウォレットに大別される。

ホットウォレットはネットワークに接続されているタイプで、ウェブウォレット、デスクトップウォレット、モバイルウォレットなどがこれに当たる。それに対してコールドウォレットはネットワークに接続していないタイプで、秘密鍵とアドレスを紙媒体などに記録して保管するペーパーウォレット、専用の USB 端末などに保存するハードウェアウォレットなどがある。

A がコールドウォレットからユーザーの秘密鍵とアドレスのデータを入手する手段は、物理的またはデジタル的な窃盗以外にはほとんどない。

ウェブウォレットは秘密鍵の管理を仮想通貨交換業者等のサービス提供者が行い、ユーザーは ID とパスワードを入力してログインしてウォレットを利用する。したがって、A はハッキングやウイルス等で ID とパスワードを入手できる可能性がある 。デスクトップウォレットやモバイルウォレットは、ユーザーが自分の PC 端末やモバイル端末で秘密鍵を管理する方式であるため、ウェブウォレットと比較すると安全性は高いが、ハッキングやウイルス等で秘密鍵を入手できる可能性がある[32]。また、ハッシュ関数に対する衝突攻撃で秘密鍵を入手できる可能性もある。

この攻撃は、ビットコインに対する脆弱性ではなく、一般のコンピュータセキュリティや暗号技術の脆弱性を突いている。

6.1.11.2. 対応策

この攻撃に対して、下記のような対策技術が提案されている。

米国政府はウォレットの保護のために指紋認証を導入した多要素セキュリティを用いた独自のビットコインネットワークを立ち上げている[“Biometric tech secures bitcoin wallet,” Biometric Technology Today, vol. 2015, no. 6, 2015.]。

また、ビットコインの秘密鍵を保護するため、トランザクションの認証に使用するハードウェアトークン「BlueWallet」が提案されている。BlueWallet はトランザクションを作成する端末と BLE（Bluetooth Low Energy）[33]で通信し、ユーザーが署名する前にそのトランザクションの内容（支払先のアドレス、支払金額、手数料）を表示するため、ユーザーはそのトランザクションが正しいものかチェックすることができる。BlueWallet 内に保管された秘密鍵はユーザーが正しい PIN を入力した時にのみアンロックされる[T. Bamert, C. Decker, R. Wattenhofer, and S. Welten, BlueWallet: The Secure Bitcoin Wallet. Springer International Publishing, 2014, pp. 65‑80.]。

ビットコインでは 1 つのアドレスに複数の秘密鍵を割り当てるマルチシグネチャ（マルチシグ）と呼ばれるサービスが提供されている。これにより、仮に 1 つの秘密鍵が盗まれたとしても、他の秘密鍵も盗まれない限りコインが盗まれることはない。

これらの対策の導入は比較的容易であると考えられ、一部で既に実装されている。

---

[32] 香港の暗号通貨取引所である Bitfinex では、マルチシグと呼ばれる認証に複数の秘密鍵が必要とされる安全性の高いシステムを使用していたが、2016 年 8 月、ハッキングの被害に遭い約 12 万 BTC（当時のレートで約 66 億円）が盗難された。

[33] 近距離無線通信技術である Bluetooth の一部で、バージョン 4.0 から追加になった低消費電力の通信モード。

6.1.11.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：9（高）
- 評価値（金融取引システムへの影響度×発生確率）：3（低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：大　　・・・　誰でも攻撃対象になり得る
  - 深刻度：高　　・・・　個人の資産が完全に盗まれる
- 金融取引システムへの影響度
  - 影響範囲：小　　・・・　システム自体に影響はない
  - 深刻度：低　　・・・　システム自体に影響はない
- 発生確率
  - 攻撃容易性：中　　・・・　ウォレットの安全性次第
  - インセンティブ：高　・・・　攻撃対象の資産を自由に入手できる

6.1.12. Transaction malleability

6.1.12.1. 概要

ユーザーや取引所から何度も同じトランザクションを送信させることで不正に利益を得ること、またはネットワークを混乱させることを目的とした攻撃。後述するトランザクションを識別するハッシュを書き換えることができる脆弱性をついた攻撃である。

攻撃対象はトランザクションの内容を確認せずにハッシュのみでトランザクションの識別を行っている一般のユーザーや取引所。

攻撃者 A は攻撃対象 V の秘密鍵を知っている必要はなく、攻撃対象からのトランザクションを書き換えるためにも特別な条件は必要ない。したがって、攻撃者 A に求められる特別な条件はない。

攻撃の手順は以下の通りである。（図 6-8 Transaction malleability 概要図参照）

① A は V（典型的には V は取引所など）に B に送金するトランザクション TR_VB を作成させる。ここで、A が利益を得る目的である場合は、B は A の支配下にあるアドレスである必要があるが、A が単にネットワークを混乱させることが目的である場合は、B は A の支配下である必要はない。

② A は、V が作成した TR_VB がネットワークに送信されるのを待つ。

③ A は受信した TR_VB をコピーし、電子署名の正当性を保ったままハッシュ値が異なるように改ざんしたトランザクション TR_VB'を作成する。TR_VB'を作成する方法としては、例えば A が V の秘密鍵を知っている場合は新しく異なる署名を作成するという方法や、A が V の秘密鍵を知らなくても、TR_VB のスクリプト部分をスクリプトの結果が変わらない範囲で変更するという方法などが挙げられる。

④ A は改ざんされたトランザクション TR_VB'をネットワークに送信する。

⑤ TR_VB'が承認されることで攻撃が成功する。V がハッシュのみでトランザクションの識別を行っている場合、TR_VB のハッシュがブロックチェーンに含まれていないため、V は B への送金が失敗したと誤認し再度同じ内容のトランザクションを送信してしまう。

この攻撃は、ビットコインにおいて、意味（支払元、支払先、金額）が同一でも記述方法を変更することで、異なるハッシュ値を有するトランザクションを作成できるという脆弱性や、そのようなトランザクションを異なる取引と識別してしまうビットコインソフトウェアの脆弱性を突いている。

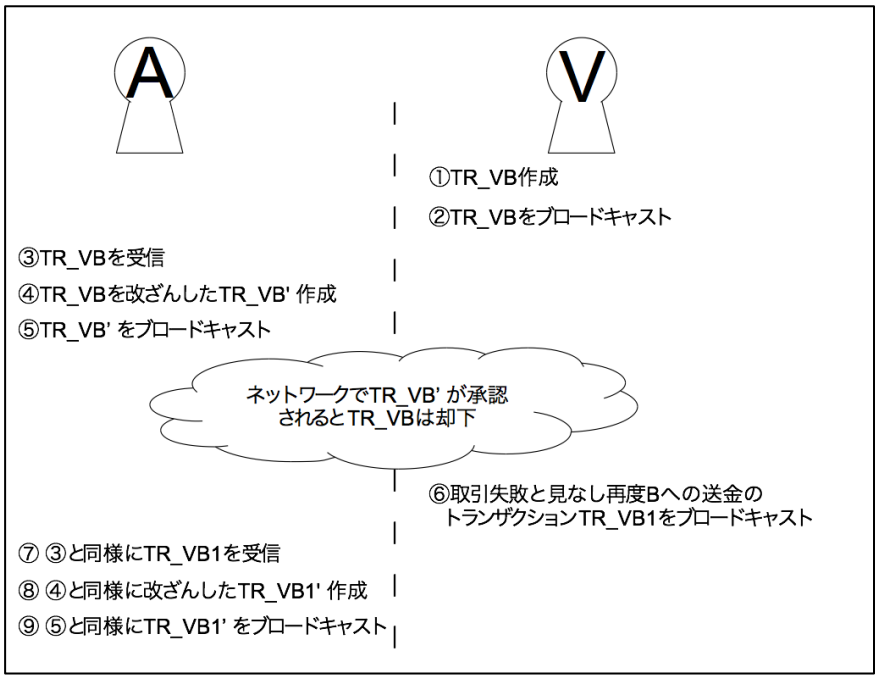


図 6-8 Transaction malleability 概要図

6.1.12.2. 対応策

この攻撃に対して、下記のような対策技術が提案されている。

- ビットコインのスクリプト言語の文法を厳格化し、異なる記法で同じ結果にならないようにする。ただし、この提案は暫定的で経験則的に決められたものであり、公式に議論されていないため、全ての文法に対して対策できているかは不明である。
- 現在はトランザクション全体からハッシュ値を計算しているが、トランザクションから入力スクリプトを除外した部分からハッシュ値を計算する方式に変更する。
- ビットコインスクリプト言語に新たな命令コード OP_CHECKLOCKTIMEVERIFY を導入することで、トランザクションの出力が将来のある時点まで使用できないようにし、A がすぐに出金できないようにする。この対策は 2015 年 10 月に導入済みである。
- 2017 年 8 月にビットコインでアクティベートされた SegWit（Segregated Witness）[34]により、従来のトランザクション全体のハッシュ値を計算する方式から、電子署名を除いたインプットとアウトプットのデータのみのハッシュ値を計算する方式に変更された。これにより、上記手順（3）のように電子署名の正当性を保ったままハッシュ値が異なるようにトランザクションを改ざんすることは不可能となった。この対策と上記の OP_CHECKLOCKTIMEVERIFY の導入により、今後はこの攻撃は発生しないと考えられる。

6.1.12.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：6（中）
- 評価値（金融取引システムへの影響度×発生確率）：3（低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：中　・・・　攻撃対象はハッシュのみでトランザクションの識別を行っている一般のユーザーや取引所
  - 深刻度：中　・・・　被害額は限定的
- 金融取引システムへの影響度
  - 影響範囲：小　・・・　システム自体に影響はない
  - 深刻度：低　・・・　攻撃対象となる取引所の数が少ない
- 発生確率
  - 攻撃容易性：高　・・・　トランザクションを書き換えるだけ
  - インセンティブ：中　・・・　利益は限定的

[34] その後ブロックチェーンネットワークにおける 1 つ 1 つのブロックのサイズを 1MB から 2MB に変更する SegWit2X が提案されたが、採用は中断された。その理由の 1 つは、SegWit2X はハードフォークを必要としリプレイアタック（ハードフォークしたブロックチェーン側で特有の署名を採用していない場合に、ユーザーが意図せずともハードフォーク側のトランザクションが実行されてしまう攻撃）の対策がなされていないことである。ここでの SegWit はハードフォークではなくソフトフォークにより実装されているため、リプレイアタックの懸念はない。

6.1.13. Time jacking

6.1.13.1. 概要

攻撃のゴールは、ブロックチェーンのネットワーク時間を不正に操作し二重使用、他のマイナーの計算資源の浪費、トランザクションの承認スピードの低下、などの攻撃を成功させることであるを目的とした攻撃。

攻撃対象は一般のユーザーやマイナー。

攻撃者 A は攻撃対象 V の IP アドレスを知っている必要がある。

ビットコインネットワークでは、各ノードはネットワーク時間を表すカウンタを内部に保持している。初めて接続するノードに対しては、この値は各ノードのカウンタの中央値がバージョンメッセージで送信される。ただし、その中央時間がシステム時間から 70 分を超えた誤差がある場合、ネットワーク時間カウンタはシステム時間にセットされる。

このネットワーク時間は新しいブロックを検証するために使用される。 ブロックのタイムスタンプが現在のネットワーク時間から 120 分時間を超えて進んでいる場合、ノードはそのブロックを拒否する。 過去 11 ブロックのタイムスタンプの中央値よりも前のタイムスタンプのブロックも拒否される。この検証により、ブロックタイムスタンプの許容範囲に上限と下限が設定される。

攻撃の手順は以下の通りである[35]。（図 6-9 Time jacking 概要図参照）

① A は複数のノードをネットワークに接続させ、それらのノードから誤ったタイムスタンプを報告させる。これにより、V の時間は実際の時間から最大 70 分遅らせ、それ以外の大部分のノードは最大 70 分進ませることができる。ここで、A の支配下にある端末数が少ない場合でも、例えば Tor を使用し発信元の IP アドレスを様々に偽装することで、接続先ノードの時間を変更すのに十分な数のバージョンメッセージを送信することができる。

② A は実際の時間よりも 190 分早いタイムスタンプを設定したブロック B を作成する。

③ V のノードからはブロック B のタイムスタンプは 260 分進んでいるように見えるため、B を拒否する。

④ 一方、他の大部分のノードからはブロック B のタイムスタンプは 120 分進んでいるように見えるため、B を承認する。

⑤ V のノードはネットワークの通常のトランザクション処理から孤立する。他のマイナーが作成したブロックは V からは 140 分進んでいるように見えるため、V はそれらのブロックを全て棄却してしまう。これにより、V に対する二重使用[36]や V の計算資源の浪費[37]などの攻撃が容易となる。

この攻撃は、ビットコインネットワークにおいて、各ノードでのネットワーク時間は、ネットワークへの接続時に隣接ノードから送信されるバージョンメッセージのタイムスタンプから決められるため、不正なバージョンメッセージを送信することにより任意のノードのネットワーク時間を進めたり遅らせたりすることが可能であるという脆弱性を突いている。

[35] corbixgwelt, “Timejacking and bitcoin”, http://culubas.blogspot.jp/2011/05/timejacking-bitcoin_802.html
[36] 二重使用されたトランザクションが他の大部分のマイナーにより承認されても、V はそのブロックを棄却し、二重使用に気がつかないため。
[37] V からは、攻撃を受けている期間中にネットワークの大部分で承認されたブロックは無効に見えるため、V はそれらのブロックの前に連結するためのブロックをマイニングしてしまう。

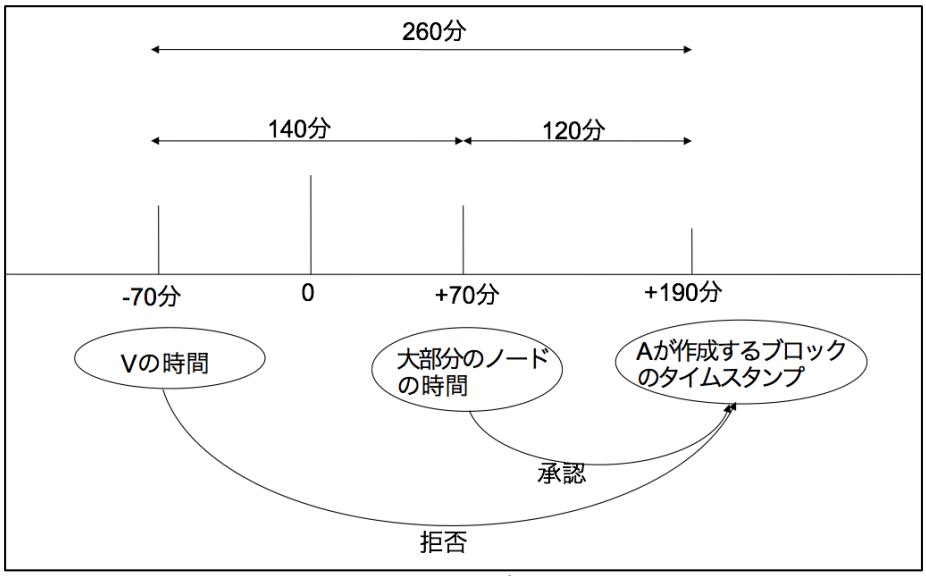


図 6-9 Time jacking 概要図

6.1.13.2. 対応策

この攻撃に対して、下記のような対策技術が提案されている。

- ブロックのタイムスタンプと、その上限を決定するために、ネットワーク時間の代わりにノードのシステム時間を使用する。ただし、これには定期的な時刻合わせが必要であり、ノード間の時刻のずれが数秒程度であっても、ネットワークの分割やノードの分離などの攻撃を受ける可能性がある
- ノードのネットワーク時間のずれの許容範囲を狭くする。現在の許容範囲である 70 分から 30 分に変更することができるが、この攻撃を完全に防止できるわけではない。
- 信頼できるノードのみを使用する。ただし、この場合、少数の信頼できるノードが攻撃対象となる可能性があり、これらのノードの安全性が低下する。これは分散システムのメリットに反する。
- ネットワーク状態を監視し、疑わしい行動があったらシャットダウンする。これは有効な対策であるが、この攻撃に対して自動的に解決するものではない。
- トランザクションを受理する前に多くの承認を要求する。ただし、これは二重使用を防ぐことはできるが、他のマイナーの計算資源の浪費など他の攻撃が成功する可能性は残っている。
- ブロックを検証する際、ブロックタイムスタンプの下限と同様に、上限にも 120 分ではなく過去のブロックのタイムスタンプの中央値から計算した値を使用する。これはこの攻撃に対する完全な解決策であり、ビットコインクライアントのルールの変更とソフトウェアの改良が必要であるが、導入は比較的容易であると考えられる。
- タイムスタンプが進んでいるブロックをメモリ内に保存しておき、後で確認する。この対策もビットコインクライアントのルールの変更とソフトウェアの改良が必要であるが、導入は比較的容易であると考えられる。

6.1.13.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：3（低）
- 評価値（金融取引システムへの影響度×発生確率）：6（中）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：小　　・・・　マイナーが攻撃対象
  - 深刻度：低　　・・・　マイナーが攻撃対象
- 金融取引システムへの影響度
  - 影響範囲：中　　・・・　システム内の時刻が非同期になる
  - 深刻度：中　　・・・　システム内の時刻が非同期になる
- 発生確率
  - 攻撃容易性：高　　・・・　時刻の不正確なブロックを生成するだけ
  - インセンティブ：中　・・・　利益は限定的

6.1.14. Sybil

6.1.14.1. 概要

攻撃者が増加させたIDを使用してtime jacking, DDoS, double spendingを実行することを目的とした攻撃。

攻撃対象は一般のユーザーやマイナー。

攻撃者 A はネットワークに多数のノードを配置する必要があるため、多数の端末あるいは協力者が必要である。

この攻撃では、攻撃者 A がネットワーク上に複数の協力者のノードを配置したり、複数の ID を作成することで、攻撃対象をネットワークから隔離し、攻撃対象によって作成されたトランザクションを切断したり、攻撃対象に、A が管理するブロックのみを選択させる。

この攻撃は、ビットコインにおいて、同一人物が複数 ID を作成することや複数の協力者ノードと共謀することが可能であるという脆弱性を突いている。

6.1.14.2. 対応策

この攻撃に対して、匿名のノード同士のペア間でコインを交換する Xim というミキシングのプロトコルが提案されている。このプロトコルでは、ノード同士のペアを作成する際に双方のノードからマイナーに手数料が支払われるため、A が複数の ID を大量に作成するとペアの作成にコストがかかり、攻撃が抑制されると考えられる。

6.1.14.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）:6（中）
- 評価値（金融取引システムへの影響度×発生確率）:6（中）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲:大 ・・・ 最悪の場合は誰でも攻撃対象になり得る
  - 深刻度:中 ・・・ 攻撃の目的に依存するが被害額は限定的
- 金融取引システムへの影響度
  - 影響範囲:大 ・・・ 最悪の場合 DDoS 攻撃によりシステムが機能しなくなる
  - 深刻度:高 ・・・ 最悪の場合 DDoS 攻撃によりシステムが機能しなくなる
- 発生確率
  - 攻撃容易性:中 ・・・ 攻撃に十分な資源が必要
  - インセンティブ:中 ・・・ 利益は限定的

6.1.15. DDoS

6.1.15.1. 概要

ネットワーク上に不正なデータを大量に流し、サービスを停止させることを目的とした攻撃。

攻撃対象は取引所、マイニングプール、e ウォレットなどネットワーク全体。

攻撃者 A は大量のデータを短時間に送信するために多数の端末を有している必要がある。

この攻撃では、攻撃者 A が多数のクライアントからネットワーク上に偽のブロックやトランザクションのような不正なデータを大量に送信することで、ネットワークのリソースを枯渇させ、ユーザーがネットワークにアクセスすることを阻害し、マイナーに通常のユーザーからのブロックを棄却させる。

この攻撃は、ビットコインネットワークにおいて、誰でも低コストで不正なデータを送信できるという脆弱性を突いている。

6.1.15.2. 対応策

この攻撃に対して堅牢な Proof of Activity（PoA）アルゴリズムが提案されている[38]。PoA では、 N 人のステークホルダーがブロックを作成する。はじめにマイナーは Proof of Work（PoW）により空のブロックヘッダーを生成し、コインの所有量に応じてランダムにステークホルダーを選択する follow-the-satoshi と呼ばれるサブルーチンを実行する。この方法で選択された N-1 人のマイナーは、空のブロックヘッダーのハッシュに署名し、それをブロードキャストする。最後に、N 番目のステークホルダーは、トランザクションを空のブロックヘッダーに追加し、このブロック全体に署名する。このアルゴリズムでは、ブロックに N 人のステークホルダーの署名が含まれているかが簡単に検証でき、不正なブロックは受信を拒否できるため、DDoS 攻撃を軽減することができる。ただし、不正なトランザクションの受信を拒否することはできないため、完全な対策法ではない。

6.1.15.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：6（中）
- 評価値（金融取引システムへの影響度×発生確率）：9（高）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：中　・・・　ノードに接続するユーザーに影響
  - 深刻度：低　・・・　システムは使えなくなるが資産が失われる訳ではない
- 金融取引システムへの影響度
  - 影響範囲：大　・・・　システムが機能しなくなる
  - 深刻度：高　・・・　システムが機能しなくなる
- 発生確率
  - 攻撃容易性：高　・・・　ツールさえあれば攻撃可能
  - インセンティブ：中　・・・　利益は限定的

[38] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, “Proof of activity:Extending bitcoin’s proof of work via proof of stake [extended abstract] y,” SIGMETRICS Perform. Eval. Rev., vol. 42, no. 3, pp. 34–37,Dec. 2014.

6.1.16. Eclipse or netsplit

6.1.16.1. 概要

攻撃対象者の全ての送信/受信コネクションを独占し、ネットワークから孤立させることを目的とした攻撃。

攻撃対象は一般のマイナーやユーザー。

攻撃者 A は多数の IP アドレスが必要であるため、ボットネットや多数の端末を有している必要がある。また、A は攻撃対象 V のブロックチェーンのクライアントを DDoS 攻撃などにより再起動させる能力が必要である。この能力がない場合、A は V のブロックチェーンのクライアントが再起動するタイミングを待って攻撃を行う必要がある。

この攻撃の目的は、A が V のルーティングを制御することによって、V のデータフローを支配し、A の意図するデータのみが承認されるようにすることである。攻撃は以下の手順で行われる。（図 6-10 Eclipse or netsplit attack 概要図参照）

① A は V の tried テーブル（コネクションが確立した IP アドレスのテーブル）に A の支配下にあるノードの IP アドレスを追加する。ビットコインではコネクションが確立したノードの IP アドレスは自動的に tried テーブルに格納されるため、A は単に A の支配下にあるノードから V に接続するだけでよい。

② A は V の new テーブル（コネクションが成功していない IP アドレスのテーブル）に、ビットコインネットワークに含まれない IP アドレスを上書きする。ビットコインでは、他のノードから受信する ADDR メッセージに記載されている 1000 個の IP アドレスを無条件に new テーブルに追加する。したがって、A は（1）で接続した V に対して、偽の IP アドレスが記載された ADDR メッセージを送信することで、これを実現する。

③ A は何らかの手段で V のクライアントを再起動させるか、再起動するまで待機する。

④ V のクライアントが再起動すると、tried テーブルと new テーブルから接続する IP アドレスを選択するが、new テーブルに格納されているのは架空の IP アドレスのみであるため、接続する IP アドレスは tried に格納されている A の支配下にあるノードのもののみとなる。

⑤ A は V のノードの残りの受信コネクションを独占する。

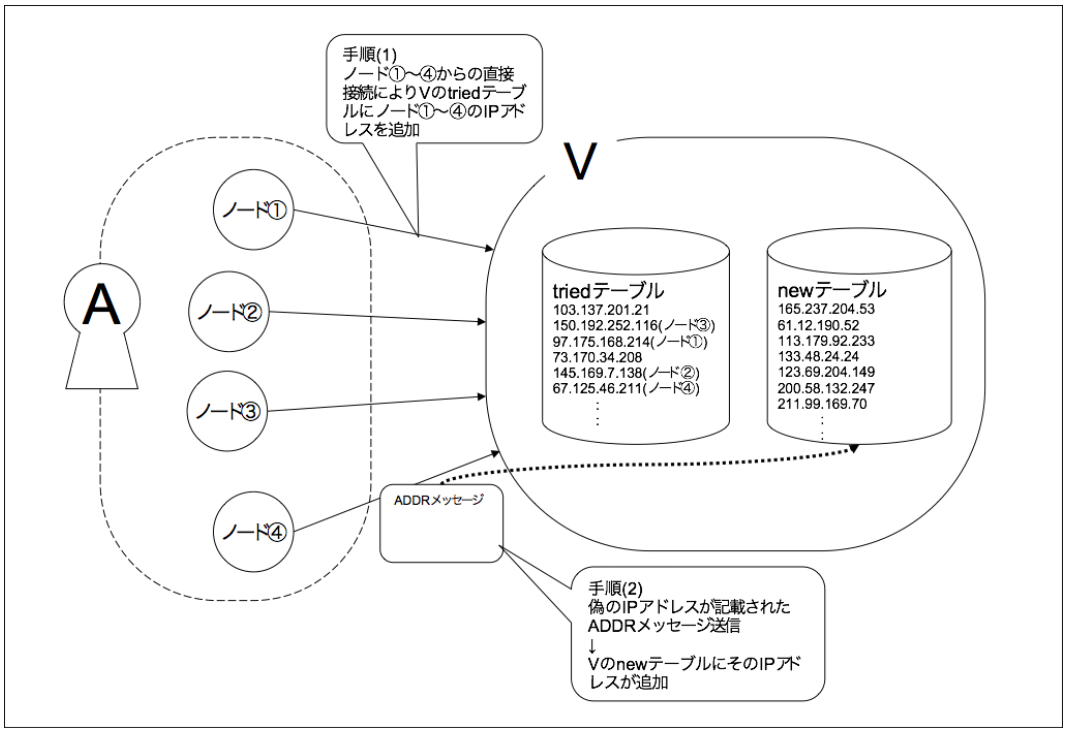


図 6-10 Eclipse or netsplit attack 概要図

6.1.16.2. 対応策

この攻撃に対して、下記のような対策技術が提案されている。これらの対策のうち、(a)、(b)、(f)は既にビットコインに導入済みである。

(a) 現行のアルゴリズムでは、ノード間のコネクションが確立した際に tried テーブルに空きがない場合、tried テーブルに元々格納されている IP アドレスからランダムに 1 つが選択され、接続相手の IP アドレスと交換される。したがって A は同じ IP アドレスで何度も V に接続することで V の tried テーブルを A の IP アドレスで満たすことができる。これを防ぐために、接続相手の IP アドレスの tried テーブルでの格納場所が、ランダムではなくその IP アドレスのハッシュを元に一意に決められるようにする。

(b) 現行のアルゴリズムでは tried テーブルに格納されている IP アドレスのタイムスタンプが新しいほど接続先として選択されやすくなっているが、これを tried テーブルと new テーブルから IP アドレスをランダムに選択する方式に変更する。

(c) tried テーブルに格納されている IP アドレスを上書きする際に、古い IP アドレスに対して接続テストを行い、それが失敗した時のみ新しい IP アドレスで上書きする。

(d) new テーブル内の IP アドレスに接続し、接続できたらその IP アドレスを tried テーブルに移動し、接続できなかったら new テーブルから削除する。

(e) 現在接続している IP アドレスとそのアドレスに最初に接続した時間を記録するための anchor テーブルを新たに追加し、再起動後には anchor テーブル内の最古の 2 つの IP アドレスに接続する。

(f) 最も有効な対策の一つは、tried テーブルのサイズを増加させることである。これにより、tried テーブル中の A の支配下にあるノードの IP アドレス数の割合を増やすために、A はより多くのが必要となる。ただし、元々 tried テーブル中の正当な IP アドレスの数が少ない場合、この対策を行っても A は自分の支配下にあるノードの IP アドレス数の割合を増やすことができる。したがって、この対策は、tried テーブル中の正当な IP アドレスの数を増やすための他の対策と合わせて行う必要がある。

(g) ノードの外向きの接続数を増加させることで、A が必要とするボットネットや端末数を増加させる。

(h) 10 件を超えるアドレスを含む ADDR メッセージの受信を拒否し、new テーブルに含まれる IP アドレスが少ない場合にのみ外向きに接続しているノードから ADDR メッセージを要求するようにする。

(i) 現在のビットコインのノードは、全ての内向きの接続を同一の IP アドレスとすることができ、これにより A は V の内向きの接続を独占することが容易となる。したがって、同一の IP アドレスからの接続数を限定させる。

(j) 様々な IP アドレスからの連続する一時的な TCP 接続、無効な IP アドレスを含む大きなサイズの ADDR メッセージ、ビットコインネットワークへのノードの接続数の急激な増加などの異常を検出するシステムを導入する。

6.1.16.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：4（中）
- 評価値（金融取引システムへの影響度×発生確率）：4（中）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：中　・・・　攻撃者から直接接続される Public IP を使用している利用者が対象
  - 深刻度：中　・・・　攻撃対象のデータが正しく流れないが被害額は限定的
- 金融取引システムへの影響度
  - 影響範囲：中　・・・　攻撃者から直接接続される Public IP を使用しているマイナーも攻撃対象
  - 深刻度：中　・・・　マイナーのデータも正しく流れないが被害額は限定的
- 発生確率
  - 攻撃容易性：低　・・・　ルーティングテーブルの改ざんは困難
  - インセンティブ：中　・・・　被害額は限定的

6.1.17. Tampering

6.1.17.1. 概要

特定のノードからのトランザクションとブロックの伝搬を遅らせ、DDoS 攻撃に利用したり、攻撃者のマイニング能力を相対的に向上させたり、二重使用攻撃に利用することを目的とした攻撃。

攻撃対象は一般のマイナーやユーザー。

攻撃者 A は攻撃対象 V に直接接続するため、V の IP アドレスを知っている必要がある。

この攻撃では、ビットコインの帯域最適化とネットワーク遅延と輻輳の対策を悪用することで、攻撃者 A は、ネットワーク構成を変えることなく、トランザクションやブロックが攻撃対象 V のノードに伝搬するのを遅らせることができる。攻撃の手順は下記の通り。（図 6-11 Tampering attack 概要図参照）

① 攻撃者 A は、トランザクションまたはブロックを、検証せず直ちに V に転送する。一方、周りのノードはトランザクションまたはブロックを検証しているので、A は V へそのデータを送信する最初のノードとなる。ビットコインのノードは、ネットワークの帯域消費を最小化するために同じデータを単一のノードからしか要求しないので、A の近隣ノードは、他のノードからの要求を一定時間受け付けない。そのデータが A によって作成されたものである場合、A はそのデータをネットワークにブロードキャストする前に V に送信すればよい。

② また、ビットコインでは、データの本体を送信する前にデータのハッシュを inv メッセージで送信し、受信側がそのデータを保有していない場合のみ getdata メッセージでそのデータ本体の送信を要求する。同じデータについての inv メッセージが複数のノードから送信された場合、受信側では各ノードを FIFO（First In First Out）でバッファに格納し、各ノードからのデータの受信を一定のタイムアウト毎に繰り返す。この攻撃においては、V が A からデータを受信する際の、V が A に getdata メッセージを送信してからそのデータを受信するまでの待ち時間が長いほど、攻撃の成功確率が高くなる。そのために、A は V に inv メッセージを繰り返し送信する。トランザクションの送信要求タイムアウトは 2 分であるため、A が x 回 inv メッセージを送信すると、タイムアウトは 2x 分となる。

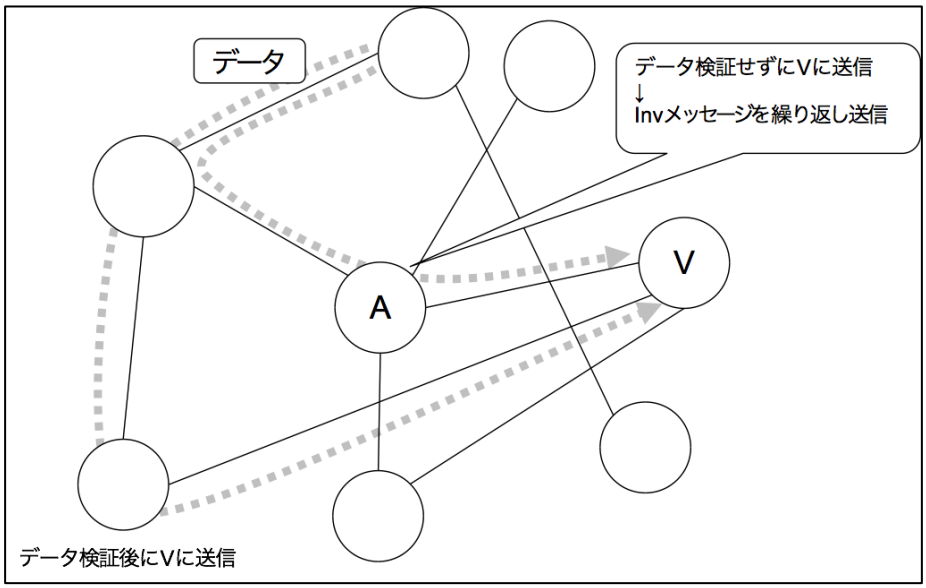


図 6-11 Tampering attack 概要図

6.1.17.2. 対応策

この攻撃に対して、下記のような対策技術が提案されている。

- タイムアウト時間を、各ノードがメッセージのサイズなどに応じて動的に設定する方式に変更する。これにより、同じノードからの複数の inv メッセージを検出することができる。
- inv メッセージを廃止し、代わりにブロックのヘッダーのみを通知する方式に変更する。
- トランザクション通知の際、各 IP アドレスからは一つの inv メッセージのみを受信する、または getdata メッセージに対して応答がなかった場合に次の送信元の同時接続数を 1 つずつ増やしてランダムに選べるようにする。
- getdata メッセージに対して常にデータ送信を遅延するノードにはペナルティを課し、ネットワークから切断する。

6.1.17.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）:4（中）
- 評価値（金融取引システムへの影響度×発生確率）:4（中）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲:中　・・・　攻撃対象ノードを利用するユーザーに影響
  - 深刻度:中　・・・　攻撃対象のデータが正しく流れないが、被害額は限定的
- 金融取引システムへの影響度
  - 影響範囲:中　・・・　攻撃対象ノードのマイナーに影響
  - 深刻度:中　・・・　マイナーのデータも正しく流れないが、被害額は限定的
- 発生確率
  - 攻撃容易性:低　・・・　攻撃者はフルノードである必要あり
  - インセンティブ:中　・・・　被害額は限定的

6.1.18. Deanonymization

6.1.18.1. 概要

IP アドレスとビットコインウォレットまたは秘密鍵アドレスをリンクさせ、ユーザーのプライバシーを侵害することを目的とした攻撃。攻撃対象はユーザー。

攻撃者 A はビットコインネットワーク内の多数のサーバに接続する必要があるため、ボットネットや多数の端末を有している必要がある。この攻撃では、ユーザーの公開鍵のハッシュ値と IP アドレスを結びつけ、そのユーザーの支払や受け取りの履歴などの情報を得ることを目的とする。攻撃の手順は下記の通りである。

① A はビットコインネットワークで Tor を使用不能にするか、Tor を使用していないユーザーのみを攻撃対象とする。Tor を使用不能にするために、例えば Tor ノードから不正なメッセージをビットコインネットワークに送信し、ビットコインからそのノードを 24 時間接続禁止にさせる。

② A は既知のノードに GETADDR メッセージを送信し、その応答の ADDR メッセージに記載されているアドレスを収集することで、ビットコインネットワークの全てのサーバのリスト S を取得する。このリストは定期的に更新される。

③ A はこの攻撃で公開鍵のハッシュ値を入手したいビットコインクライアントの IP アドレスのリスト C を構成する。

④ A は S に含まれるサーバに接続しておき、C に含まれるクライアント V がビットコインネットワークに接続した時に送信する V 自身のアドレスを収集することで、V のアドレスと V が接続しているノードの組の組み合わせ E'P を特定する。

⑤ A は、S に含まれるサーバからの inv メッセージを収集し、ある同一 inv メッセージを転送してきた複数ノードのアドレスの組 RT を取得する。

⑥ E'P と RT を比較し、V の IP アドレスとそれに対応するトランザクションの組を得る。

6.1.18.2. 対応策

この攻撃に対して、それぞれのトランザクションごとにクライアントの IP アドレスを変更し、トランザクション後にランダムな遅延を加えるという対策技術が提案されている。これにより、A は IP アドレスとトランザクションを関連づけられなくなるが、V の ISP が知られてしまうというリスクを軽減することはできない。また、複数のトランザクションをミックスした後で最終的な送金先に再配分するミキシングサービスを利用することで、この攻撃を防ぐことができると考えられる。

6.1.18.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：3（低）
- 評価値（金融取引システムへの影響度×発生確率）：1（低）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：大　・・・　誰でも攻撃可能
  - 深刻度：低　・・・　金銭的被害はない
- 金融取引システムへの影響度
  - 影響範囲：小　・・・　システムに影響なし
  - 深刻度：低　・・・　システムに影響なし
- 発生確率
  - 攻撃容易性：低　・・・　攻撃者はサーバの応答を盗聴する必要あり
  - インセンティブ：低　・・・　攻撃者に直接の利益はない

6.1.19. Compromise of underlying cryptographic algorithms

6.1.19.1. 概要

ブロックチェーンのアーキテクチャでは、使用されている暗号アルゴリズムに関してアルゴリズムの移行が考慮されておらず、ブロックチェーンで使用されている暗号アルゴリズムが危殆化した時に、それを基盤とする金融取引の全ての価値が失われるリスクがある。例えば、電子署名はブロックチェーンを構成するトランザクションの真正性（トランザクションの内容が改ざんされていないことを保証する性質）を確保するために付与されるが、電子署名アルゴリズムが危殆化すると、トランザクションの改ざんが可能となり、トランザクションの真正性が確保できなくなる。これにより、トランザクションに記録された取引情報の信頼性が失われ、金融取引の価値がなくなる。

暗号アルゴリズムの危殆化には二種類の顕在化パターンが存在する。ひとつは、計算機能力の発展に伴い暗号アルゴリズムが計算量的に解読される場合である。もうひとつは、暗号アルゴリズムに対して効率的な解読方法が発見された場合である。前者は数十年ごとに必ず発生するリスクである一方、後者は今すぐにでも発生するリスクである。二種類の顕在化パターンの例について、付録 A で具体例を示す。

したがって、ブロックチェーンを用いた金融取引において長期間に渡り安定した運用を行うためには、暗号アルゴリズム危殆化のリスクに対応できるよう、暗号アルゴリズムを変更可能なブロックチェーンを利用する必要がある。暗号アルゴリズムを変更可能にすることにより、過去に利用していた暗号アルゴリズムが危殆化した場合でも、過去から現在におけるブロックチェーンの真正性を確保することができる。

6.1.19.2. 対応策

暗号アルゴリズム危殆化のリスクに対し、長期署名技術を応用し、古い暗号アルゴリズムが危殆化する前に新しい暗号アルゴリズムに切り替えるという対応策が提案されている。古い暗号アルゴリズムと新しい暗号アルゴリズムの利用期間が重なっている間に、それまでの金融取引の有効性を新しい暗号アルゴリズムを用いて保証する。

調査対象論文[39]において、二通りの実現方式が提案されている。

- 元のチェーンをそのまま繋げる方式（論文中の Fig. 6）
- サポートチェーンを利用する（論文中の Fig. 7）

[39] Masashi Sato and Shin'ichiro Matsuo, "Long-term public blockchain: Resilience against Compromise of Underlying Cryptography," ICCCN 2017.

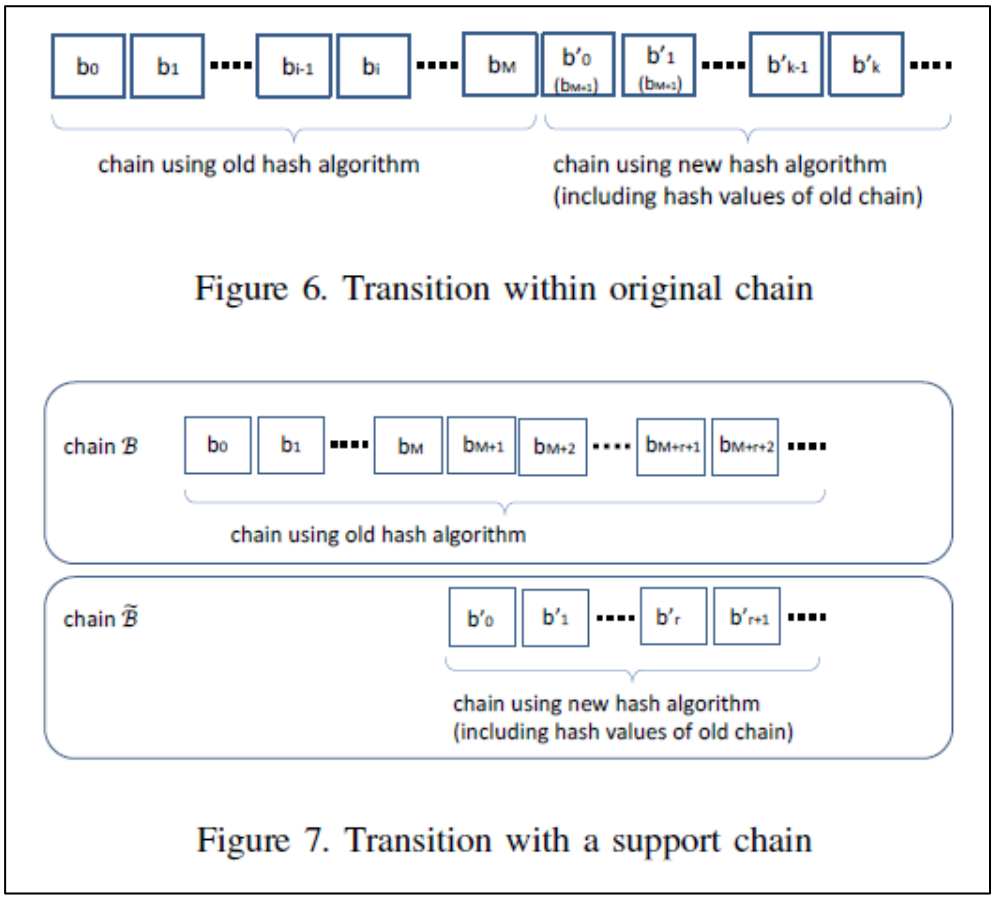


図 6-12 長期署名技術を用いたブロックチェーンの実現方法（調査対象論文[36]より抜粋）

長期署名技術を用いたブロックチェーンを実装する場合、以下の点に注意する必要がある。

(1) クライアントのバージョン管理が必要

- 古い暗号アルゴリズムのみのバージョン
- 新しい暗号アルゴリズムを追加したバージョン（まだ古い暗号も安全）
- 新しい暗号アルゴリズムのみを受け付けるバージョン（古い暗号を使ったブロックは追加の検証が必要）

(2) サポートチェーン方式において、トランザクションをどのように一致させるか
（マイナーの報酬はどちらかのみに入れる？）

(3) ソフトフォークで実現しようとする場合、ブロックヘッダーを変える必要がある（追加したフィールドを無視する仕組みが古い実装には存在しない）

(4) 古い実装と新しい実装が共存した際、新しい実装で作られたブロックを古い実装では検証できない。検証できないブロックを伸ばすモチベーションをマイナーに与えられるか。実装の移行をどのようにするかが問題。

付録 A. 暗号アルゴリズムの危殆化に関する具体例

(1) 計算機能力の発展による危殆化

計算機能力の発展による危殆化の事例として、DES Challenge および AES（Advanced Encryption Standard）の選定が挙げられる。DES（Data Encryption Standard）は 1977 年に米国国立標準技術研究所（NIST: National Institute of Standards and Technology）にて FIPS（Federal Information Processing Standards）として発行された、鍵長 64bit のブロック暗号アルゴリズムである。DES の安全性については 5 年ごとに見直しが行われていたが、線形解読法の発見や計算機能力の発展により DES の安全性の低下が本格的になった。1997 年以降 DES Challenge と呼ばれる DES 解読コンテストが開催され、1999 年には 22 時間 15 分で解読されるに至った。これに対し NIST は、DES を 3 回繰り返す Triple DES（3DES）を提案して DES の安全な使用を促すとともに、DES に代わるブロック暗号アルゴリズム標準を策定するプロジェクトを開始した。2001 年、新たな 128bit ブロック暗号アルゴリズムである AES が FIPS として発行された。2005 年、DES の仕様を定めた FIPS が廃止され、DES の使用が事実上禁止された。

(2) 解読方法の発見による危殆化

解読方法の発見による危殆化の事例として、MD5 の解読とそれに伴う APOP（Authenticated Post Office Protocol/Automatic Processing Options Protocol）の使用禁止が挙げられる[40][41]。MD5 は 1991 年に開発された 128bit 出力のハッシュ関数である。MD5 を利用して電子メール受信時の認証を実現したプロトコルが APOP である。APOP のパスワード認証方法を図 6-13 APOP のパスワード認証に示す。

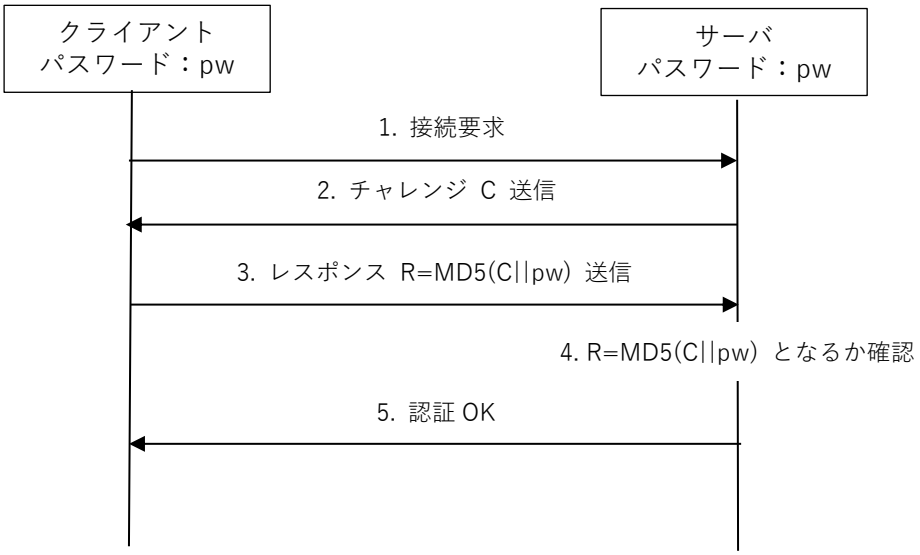


図 6-13 APOP のパスワード認証

2004 年に MD5（Message Digest Algorithm 5）に対する実時間で実行可能な衝突発見攻撃が提案された。この攻撃を APOP に適用すると、実用上 31 文字までのパスワードを復元可能であることが 2007 年に報告された。この攻撃により、APOP にプロトコル上の脆弱性が見つかったことになり、APOP の使用を避けるよう勧告された。

また、ブロックチェーンの暗号アルゴリズムが解読された事例として、IOTA で使用されていたハッシュ関数 Curl が破られた事例が挙げられる[42]。この事例では、IOTA において安全性が十分に議論されたハッシュ関数ではなく、独自に設計したハッシュ関数を使用していたことが解読の一因であると考えらえる。この脆弱性の発見により、IOTA はハッシュ関数 Curl をハッシュ関数 KECCAK（SHA-3: Secure Hashing Algorithm 3）に置き換えるハードフォークを実施した。これにより、仮想通貨取引が約 3 日間停止した。

[40] IPA「脆弱性関連情報取扱い：APOP 方式におけるセキュリティ上の弱点（脆弱性）の注意喚起について」、https://www.ipa.go.jp/security/vuln/200704_APOP.html
[41] IPA「MD5 の安全性の限界に関する調査研究報告書」、2008 年 7 月、 https://www.ipa.go.jp/files/000013897.pdf
[42] IOTA Vulnerability Report: Cryptanalysis of the Curl Hash Function Enabling Practical Signature Forgery Attacks on the IOTA Cryptocurrency , https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md

6.1.19.3. 評価

本リスクを以下の通り評価した。

- 評価値（利用者への影響度×発生確率）：9（高）
- 評価値（金融取引システムへの影響度×発生確率）：9（高）

評価根拠は以下の通りである。

- 利用者への影響度
  - 影響範囲：大　　・・・　誰でも攻撃対象になり得る
  - 深刻度：高　　・・・　資金価値がなくなる
- 金融取引システムへの影響度
  - 影響範囲：大　　・・・　システムそのものを無効にする可能性あり
  - 深刻度：高　　・・・　システムそのものを無効にする可能性あり
- 発生確率
  - 攻撃容易性：低　　・・・　現時点では安全な暗号アルゴリズムが使用されている
  - インセンティブ：高　・・・　トランザクション改ざんにより利益を得られる

6.2. 用語集

fast payment

主に一般の店舗での少額の取引において、トランザクションがブロックチェーンに取り込まれ、さらにその後に複数のブロックが連結されるまで待たずに支払いを受理する仕組み。店舗側がトランザクションを受信し、署名の検証や、支払いに使うコインが未使用であることの確認ができた時点で、支払いが完了したとみなす。

アドレス

ビットコインアドレスとは、ビットコインを使用するうえでの口座番号のようなもの、このアドレスは 1 か 3 から始まる 27~34 文字の英数字となっている。公開鍵から生成される。

P2P

Peer to Peer の略であり、サーバなどを介さず、目的の端末同士で通信を行うアーキテクチャ。

ノード

ビットコインネットワークに参加しているプログラム 1 つ 1 つのこと。さらにマイニングを主体とするノード、ウォレット機能を主体とするノード、軽量化ウォレット（SPV）機能を主体とするノードなどの種類がある。

IP アドレス

IP と呼ばれるネットワーク上で、通信を送受信する機器を判別するための番号である。

ブロードキャスト

あるデータを不特定多数に同時に送信すること。

ビットコインの取引において、取引を行うとまず取引情報がネットワーク上の全ノードにブロードキャストされ、そのノードによって取引が承認されると、承認された取引データ(ブロック)が再びネットワーク上の全ノードに向けてブロードキャストされる

フォーク

同時に複数ブロックがマイニングされることなどにより、複数のブロックが同じブロックの後に加えられブロックが分岐すること。

ビットコインミキサー

ビットコインを使用するうえで匿名性を高める技術

オフチェーン

本来ブロックチェーン上での送金や取引を行うものをブロックチェーン外で行うこと

静的 IP アドレス

固定された IP アドレスのこと

マイニングプール

複数のマイナーで協力してマイニングを行う仕組み

ブロック承認

ビットコインのある取引がブロックチェーンのブロックに載る状態のこと

クラウドマイニング

マイニングを行っているサービスに投資を行うこと

BIP70

売り手と顧客のビットコインを使った決済プロセス中の中間者攻撃を防ぐために定義されたプロトコルのこと

秘密鍵

公開鍵暗号方式で使用される一対の鍵の組のうち、一般に公開されない鍵

ソフトフォーク

ブロックチェーンのプロトコルに規定された検証規則をより厳密なものに変更することによって発生するブロックチェーンの分岐で、以前の通貨と互換性があるアップデート

ハードフォーク

ブロックチェーンのプロトコルに規定された検証規則を緩和することによって発生するブロックチェーンの分岐で、互換性のない 2 つの通貨に別れるアップデート

ミキシング

ビットコインをいつ手に入れたものなのかを他の人たちに追跡されないようにするためのシステム、仮想通貨の匿名性を高める技術。3 世代に分けられていると言われており、「中央集権によるミックシングサービス」「P2P ミキサー」「匿名アルトコイン」の 3 つが存在する

ボットネット

サイバー犯罪者がトロイの木馬やその他の悪意あるプログラムを使用して乗っ取った多数のゾンビコンピュータで構成されるネットワークのこと

DDoS 攻撃

多数の端末から 1 つのサービスへ DoS 攻撃を行うこと

DoS 攻撃

Denial of Service attack、ウェブサービスを稼働しているサーバやネットワークなどのリソース（資源）に意図的に過剰な負荷をかけたり脆弱性をついたりすることでサービスを妨害する攻撃

ルーティング

宛先となるホストまでパケットを送信する時に最適な経路を選択して転送すること

TCP 接続

TCP (Transmission Control Protocol)、IP と同様にインターネットにおいて標準的に利用されているプロトコルで行う接続のこと

内向き/外向きの接続

ビットコインネットワークにおける接続方法、自ら他のノードへ接続を行うことを外向き、他のノードから自らへ接続を行われることが内向き

ハッシュ値

ハッシュは、メッセージを特定するための暗号化技術である。受信者がメッセージを受け取ったときに、通信経路上で改ざんされていないか、受け取ったデータが壊れていないかを確認するために使う

Tor

暗号化と複数のノードをプロキシ接続することによって匿名での通信を行うための技術、あるいはそのような技術を実現するためのソフトウェアの名称

公開鍵

公開鍵暗号方式で使用される一対の鍵の組のうち、一般に公開される鍵

公開鍵暗号方式

暗号化と復号に別個の鍵（手順）を使い、暗号化のための鍵を公開できるようにした暗号方式

衝突攻撃

主にハッシュ関数を使用した暗号化方式において、異なるデータを暗号化して同一のハッシュ値が出力される（値が衝突する）ことを利用した攻撃方法のこと

原像攻撃

特定のハッシュ値を持つメッセージを探索する攻撃。与えられたハッシュ値 h に対して、hash(m)=h となるようなメッセージ m を探索する第一原像攻撃と、与えられたメッセージ m1 に対して、hash(m2)=hash(m1)となるような別のメッセージ m2 を探索する第二原像攻撃の二種類がある。

デジタル署名

書面上の手書き署名のセキュリティ特性を模倣するために用いられる公開鍵暗号技術の一種

仮想通貨

インターネットを通じて不特定多数の間で物品やサービスの対価に使用でき、中央銀行などの公的な発行主体や管理者が存在しないもの

分散データベース

1 つのデータベース管理システム（DBMS）が複数の CPU に接続されている記憶装置群を制御する形態のデータベース

サプライチェーン

製品の原材料が生産されてから消費者に届くまでの一連の工程

メモリプール

ソフトウェア（ここではビットコインクライアント）で予め割り当てたメモリ領域

6.3. 調査対象論文一覧表

6.3.1. 対象論文の選定方法

本研究の対象とした論文は以下の方法により選定しております。

(1) 項番 0 のサーベイ論文から現存する攻撃手法（リスク要因）および該当論文をリストアップ。

(2) 上記(1)に加え、暗号研究分野における金融に関する研究が盛んな Financial Cryptgraphy とその派生ワークショップ、および国際暗号学会（IACR）で主催される主要な国際会議（CRYPTO・EUROCRYPT・ASI ACRYPT 等）の発表論文から関連した論文を調査対象に追加。

(3) 最終的に項番 0 のサーベイ論文を含め 27 の論文を対象として選定。

対象論文については、関心を持たれた方が自由に確認できることを考慮し、Web 上に無料公開されているものを選定し、有料論文は対象外としております。

表 6-1 調査対象論文一覧表

項番	分類	リスク要因	該当論文			
			著者およびタイトル	発表媒体	掲載先 URL	発表媒体種類
0	サーベイ論文	―	・Mauro Conti, Sandeep Kumar E, Chhagan Lal, Sushmita Ruj, "A Survey on Security and Privacy Issues of Bitcoin"	・arXiv	https://arxiv.org/pdf/1706.00916	論文
1	二重使用攻撃	Double spending or Race attack	・G. O. Karame, E. Androulaki, and S. Capkun, Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin”	・2012 ACM Conference on Computer and Communications Security	https://eprint.iacr.org/2012/248.pdf	論文
2		Finney attack	・H. Finney, ckt.iacr.org/2012/248.pdfransaction acceptance - how high is the risk?”	・Bitcoin Forum	https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384	Web サイト
3		Brute force attack	・Meni Rosenfeld, "Analysis of hashrate-based double-spending"	・arXiv	https://arxiv.org/pdf/1402.2009.pdf	論文
4		Vector 76 or one-confirmation attack	・Vector67, or one-confirmatio・sgornick, "Vector76 Double Spend Attack?"	・Bitcoin Forum ・reddit.com	https://bitcointalk.org/index.php?topic=36788.msg463391#msg463391 https://www.reddit.com/r/Bitcoin/comments/2e7bfa/vector76_double_spend_attack/	Web サイト
5		>50% hashpower or Goldfinger (Majority attack)	・J. A. Kroll, I. C. Davey, and E. W. Felten, “The economics of bitcoin mining, or bitcoin in the presence of adversaries”	・WEIS 2013	http://www.thebitcoin.fr/wp-content/uploads/2014/01/The-Economics-of-Bitcoin-Mining-or-Bitcoin-in-the-Presence-of-Adversaries.pdf	論文

項番	分類	リスク要因	該当論文			
			著者およびタイトル	発表媒体	掲載先 URL	発表媒体種類
6	マイニングプール攻撃	Block discarding or Selfish mining	・N. T. Courtois and L. Bahack, ningt/uploads/2014/01/The-Economics-of-Bitcothholding attack in bitcoin digital currency” ・L. Bahack, ois and L. Bahack, ningt/uploads/2014/01/The-Economics-of-Bitcothholding attack in bit・I. Eyal and E. G. Sirer, ack, ningt/uploads/2014/01/The-Economics-of-Bitcothhol	・CoRR	https://allquantor.at/blockchainbib/pdf/courtois2014subversive.pdf https://eprint.iacr.org/2013/868.pdf https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf	論文
7	マイニングプール攻撃	Block withholding	・M. Rosenfeld, ingornell.edu/~ie53/publications/btcProcFC.pdfff.ed	・arXiv	https://bitcoil.co.il/pool_analysis.pdf	論文
8		Bribery attacks	・J. Bonneau, “Why buy when you can rent? Bribery attacks on Bitcoin-style consensus”	・BITCOIN 2016	http://fc16.ifca.ai/bitcoin/papers/Bon16b.pdf	論文
9		Refund attacks	・P. McCorry, S. F. Shahandashti, and F. Hao, “Refund attacks on Bitcoin's Payment protocol”	・Cryptology ePrint archive	https://eprint.iacr.org/2016/024.pdf	論文
10		Punitive and Feather forking	・A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction" ・A. Miller, "Feather-forks: enforcing a blacklist with sub-50% hash power"	・Princeton University Press ・Bitcoin Forum	https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf https://bitcointalk.org/index.php?topic=312668.0	・書籍 ・Web サイト
11	クライアント側のセキュリティ脅威	Wallet theft	・Securing your wallet	・Bitcoin Wiki	https://en.bitcoin.it/wiki/Securing_your_wallet	Web サイト
12	ビットコインプロトコルやネットワークインフラへの攻撃	Transaction malleability	・M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "On the Malleability of Bitcoin Transactions" ・Christian Decker and Roger Wattenhofer, "Bitcoin transaction malleability and MtGox"	・Financial Cryptography and Data Security 2015 ・ESORICS 2014	https://ai2-s2-pdfs.s3.amazonaws.com/c276/84f2fe5a85fe2871f693edc46061d0ecb20d.pdf https://www.tik.ee.ethz.ch/file/7e4a7f3f29917847860372854876f5c/malleability.pdf	論文
13		Time jacking	・corbixgwelt, tik.ee.ethz.ch/file/7e4a7	・culubas	http://culubas.blogspot.jp/2011/05/timejacking-bitcoin_802.html	Web サイト
14		Sybil	・J. R. Douceur, blogspot.jp/2011/0	・First International Workshop on Peer-to-Peer Systems (IPTPS) 2001	http://www.divms.uiowa.edu/~ghosh/sybil.pdf	論文

項番	分類	リスク要因	該当論文			
			著者およびタイトル	発表媒体	掲載先 URL	発表媒体種類
15		DDoS	・M. Vasek, M. Thornton, and T. Moore, "Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem"	・Financial Cryptography and Data Security 2014	http://fc14.ifca.ai/bitcoin/papers/bitcoin14_submission_17.pdf	論文
16		Eclipse or netsplit	・E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, sion_17.pdttacks on bitcoin's peer-to-peer network"	・USENIX Conference on Security Symposium (SEC) 2015	https://eprint.iacr.org/2015/263.pdf	論文
17		Tampering	・A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin"	・ACM CCS 2015	https://scalingbitcoin.org/hongkong2015/presentations/DAY1/3_block_propagation_2_gervais.pdf	論文
18		Deanonymization	・P. Koshy, D. Koshy, and P. McDaniel, "An Analysis of Anonymity in Bitcoin Using P2P Network Traffic" ・A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network"	・Financial Cryptography and Data Security 2014 ・ACM CCS 2014	https://pdfs.semanticscholar.org/c277/62257f068fdbb2ad34e8f787d8af13fac7d1.pdf https://orbilu.uni.lu/bitstream/10993/18679/1/Ccsfp614s-biryukovATS.pdf	論文
19		Compromise of underlying cryptographic algorithms	・I. Giechaskiel, C. Cremers, and K. B. Rasmussen, "On Bitcoin Security in the Presence of Broken Cryptographic Primitives" ・Masashi Sato and Shin'Ichiro Matsuo, "Long-term Public Blockchain: Resilience against Compromise of Underlying Cryptography"	・ESORICS2016 ・ICCCN2017	(ICCCN_Blockchain_matsuo_cameraready_8p.pdf)	論文

6.4. 実証実験準備の詳細

6.4.1. BSafe.network 版ビットコインへの暗号アルゴリズム危殆化対応策の実装

BSafe.network 版ビットコインに対し、以下の改修を加えた。

6.4.1.1. archiveHash の実装

archiveHash の計算および検証機能を実装した。実装方針は以下の通りである。

- コンセンサスルールを更新し、ハードフォークを行うこととした（これにより、archiveHash をサポートしてないノードは元のチェーンに取り残される）
- コンセンサスルールに追加するパラメータは以下の 3 つとした。
  - archiveHash を付加し始めるチェーンの高さ
  - 1 つの archiveHash に何ブロック分のアーカイブを行うか
  - archiveHash をいくつ付加するか
- SPV ノードをサポートするために、以下の実装を行うこととした。
  - archiveHash を archiveHeaderHash と archiveTransactionHash に分離
  - segwit と整合性のために archiveTransactionHash を archiveWitnessHash に分離

archiveHash の実装箇所は以下の通りである。

- ブロックヘッダーに上記 3 つのハッシュ値（archiveHeaderHash, archiveTransac-tionHash, archiveWitnessHash）を追加（これによりブロックヘッダーが 96 バイト増加する）
- ブロックヘッダーのバージョンフィールドに archiveHash を持っているかどうかのフラグを追加
- コンセンサスルールに archiveHash 設定を持っているノードが新規ブロックを受け取った場合、ブロックの高さが archiveHash 設定の範囲外の場合は archiveHash 付加フラグが立っていないことを検証。範囲内の場合は立っていることを検証し、さらに対応するアーカイブ対象ブロックをブロックチェーンから読み出し、3 つのハッシュ値（SPV ノードの場合は archiveHeaderHash のみ）を計算して新規ブロック内のハッシュ値と一致するかを検証。
- それぞれのハッシュ値の計算方法は以下の通り
  - archiveHeaderHash
    - アーカイブ対象の各ブロックヘッダーのバイト列表現と最後のブロックヘッダーのハッシュ値を結合して、ハッシュを計算する。
  - archiveTransactionHash
    - アーカイブ対象の各ブロックに含まれるトランザクションのハッシュ値を Merkle tree により計算する。
    - 将来的に SPV ノードでは、partial Merkle tree を使ってアーカイブ対象のトランザクションを検証することが望ましい（本実証実験では未実装）
  - archiveWitnessHash
    - アーカイブ対象の各ブロックに含まれるトランザクションの witness のハッシュ値を、Merkle tree により計算する

6.4.1.2. ハッシュ関数 SHA-512 の実装

Proof of Work およびトランザクションの Merkle tree で用いられるハッシュ関数 SHA-256 を SHA-512 に変更する実装を行った。実装方針および留意点は以下の通りである。

- 現状のビットコインの実装では、ハッシュ関数のビット長を変えることを全く想定しておらず、ハッシュ値が 256bit であることに依存したコードとなっている。そのため、ビット長が変わるようなハッシュ関数の変更はコードへの影響範囲が大きく、実装が困難であった。そのため本実証実験では、新しいハッシュ関数である SHA-512 の結果を 256bit に切り詰めることとした。
- ブロックヘッダーのハッシュ値 (Proof of Work) 、トランザクションの Merkle tree、および archiveHash の計算で利用するハッシュ関数を新しいものに変更する。
- トランザクション ID の計算は古いハッシュ関数 (RIPEMD-160) のままとする。
- コンセンサスルールを更新し、ハードフォークを行う（新しい Proof of Work をサポートしていないノードは元のチェーンに取り残される）
- コンセンサスルールに、新しい Proof of Work に移行するチェーンの高さを新たにパラメータとして追加する。

実装箇所は以下の通りである。

- ビットコインの実装に既に含まれている SHA-512 を利用し、その結果を 256bit に切り詰めることで新しいハッシュ関数を実装した。
- ブロックヘッダーのバージョンフィールドに新しい Proof of Work を使うかどうかのフラグを追加した。
- ブロック生成時にチェーンの高さがコンセンサスルールに設定された高さ以上になる場合、新しい Proof of Work のフラグを立てたブロックを作成し、Merkle tree も新しいハッシュ関数で計算することとした。
- ブロックヘッダーのハッシュ値を計算するときにフラグを見てハッシュ関数を切り替えることとした。
- 以下の通りブロックの検証として実装した。
  - コンセンサスルールに新しい Proof of Work の設定を持っているノードが新規ブロックを受け取った場合、チェーンの高さが設定値未満であった場合はブロックにフラグが立っていないことを検証し、チェーンの高さが設定値以上であった場合は立っていることを検証する。
  - ブロックの検証の残りの部分は、フラグによってハッシュ関数を切り替える以外は既存のものを使用する。

6.4.1.3. ECDSA の鍵長の変更

ECDSA の鍵長を現在の 256bit から 384bit に変更する実装を行った。これにより、利用する曲線も変更となる。実装方針は以下の通りである。

- 現状の署名アルゴリズムは曲線 secp256k1 を用いた ECDSA のみだが、曲線 secp384r1 も利用可能にする。
- secp384r1 の実装は OpenSSL を用いる。
- ウォレット内の残高を新しいアドレスに移行するコマンドを追加する。

ECDSA 鍵長変更の実装箇所は以下の通りである。

- 秘密鍵のバイト数を32バイト固定から32または48バイトの可変長にし、フラグで管理することとした。
- 秘密鍵の保存は現状のビットコインと同様に DER 形式で行うこととした。
- ノードの再起動時に現状のビットコインと同じ形式でウォレットをロードし、それが失敗した場合は秘密鍵を 48 バイトとしてロードを行うこととした。
- 公開鍵のバイト数は現在の 33 バイト（圧縮表現）または 65 バイト（非圧縮）に加え、49 バイト（圧縮表現）を追加した。
- 公開鍵の先頭のバイトで署名アルゴリズムの種類の判定を行うため、新しくフラグを追加した。
- 署名は現状と同様に DER 形式で記録することとした。
- 署名を行う際には秘密鍵のフラグをチェックし、48 バイトの秘密鍵の場合は OpenSSL の ECDSA ライブラリを呼び出して署名を生成することとした。
- 署名の検証は、公開鍵のバイト数に応じて現状のビットコイン方式または OpenSSL の ECDSA ライブラリを呼び出して検証を行うこととした。
- 署名を検証する際、DER 形式を現状のビットコイン方式でパースして、成功した場合は現状のビットコイン方式で検証を行い、失敗した場合は OpenSSL の ECDSA ライブラリを呼び出して検証を行うこととした。
- ビットコインアドレスは現状と同様、公開鍵のハッシュ値 160bit を使った P2PKH のままとした。
- コマンドを追加して、新しい方式で鍵を生成しウォレットに追加できるようにした。さらにそのアドレスをデフォルトの受け取り先に設定することとした。これにより、新しいアドレスに全額送金することで、ウォレット内の新たな鍵長の ECDSA 鍵に移行することが可能になる。
- お釣りを受け取るアドレスについては、その都度生成する方式から、デフォルトで設定した受け取り先を使うように変更した。

6.4.2. BSafe.network への配備

BSafe.network への配備は以下のように行った。配備したノードは次の 2 種類である。

- フルノード
- SPV ノード

この 2 種類のノードを、4.3.3 に示した 5 つのシナリオそれぞれに対応するネットワークに対して立ち上げた。

6.4.2.1. フルノードの配備

BSafe.network 版ビットコインのフルノード実装に対し、4.1 に示した機能を実装したプログラム を、各フルノードにインストールした。インストール手順は下記の通りである。

(1) ソースコードを GitHub から取得して展開

```
$ git clone https://github.com/BSafe-network/LongTermBlockchain
$ cd LongTermBlockchain
$ git checkout bsafe-long-term-20180130 #  シナリオ 1～3 に対応するタグ:
bsafe-long-term-20180130、シナリオ 4～5 に対応するタグ: bsafe-long-term-20180215
$ ./autogen.sh
$ ./configure
$ make
$ make install # optional
```

(2) シナリオごとにディレクトリを作成する。

```
$ mkdir bsafenetlt1
```

(3) 作成したディレクトリに bitcoin.conf を設置する。

```
$ vi bsafenetlt1/bitcoin.conf
bsafenetlt1=1    # or bsafenetlt12 or bsafenetlt3 or bsafenetlt4 of bsafenetlt5
dnsseed=0
upnp=0

server=1
rpcallowip=0.0.0.0/0

rpcuser=user
rpcpassword=password
seednode=202.16.211.119 #  シードノードの IP アドレス（環境に応じて変更する）
```

(4) bitcoind を起動する。

```
$ LongTermBlockchain/src/bitcoind -datadir=bsafenetlt1
```

(5) マイニングを行うために bitcoin-cli でコマンドを実行する。

```
$ LongTermBlockchain/src/bitcoin-cli -datadir=bsafenetlt1 generate 1
```

6.4.3. 実証実験用データ

実証実験において、シナリオ 1～3 ではデータとしてブロックを流して実験を行った。ブロックはマイニングを行うコマンドである bitcoin-cli を実行することで生成できる。

一方、シナリオ 4～5 ではデータとしてトランザクションを流して実験を行った。これは、ECDSA がトランザクションに対して付与される署名であるため、ECDSA の鍵長変更の影響を観察するためにはトランザクションを流す必要があるからである。

本実証実験では、実際のビットコインネットワーク (mainnet) から流れているトランザクションと同程度のトランザクションを BSafe.network に流すために、mainnet に流れているトランザクションを BSafe.network に合わせて変換して流すためのツール (txrelay.jar) を開発した。トランザクションの変換にあたり、送金先の公開鍵と金額はそのまま利用することとした。また、送金元は BSafe.network の単一のウォレットから全て支出するものとした。

txrelay.jar の詳細を説明する。まず、mainnet のノードに接続し、受け取ったトランザクションに対して次の通り変換を施す。

- 送金先アドレスのマジックナンバー（ネットワーク識別子）を mainnet のものから BSafe.network のものに変更
- 送金額はそのまま利用

実装は以下の通りである。

```
Map<String, Double> amounts = new HashMap<String, Double>();
for (TransactionOutput output: transaction.getOutputs()) {
        Address address = output.getAddressFromP2PKHScript(params);
        if (address == null) {
                address = output.getAddressFromP2SH(params);
        }
        if (address != null) {
                if (address.isP2SHAddress()) {
                        address = new Address(dstParams, dstParams.getP2SHHeader(),
                        address.getHash160());
                } else {
                        address = new Address(dstParams, dstParams.getAddressHeader(),
                        address.getHash160());
                }
                amounts.put(address.toBase58(), (double)output.getValue().getValue() /
        Coin.COIN.getValue());
        }
}
```

できあがった送金先と金額のリストに対し、BSafe.network のノードに RPC コマンドを送信し、このノードのウォレットから送金を行う。実装は以下の通りである。

```
JSONObject json = sendmany(amounts);
System.err.println(json.toJSONString());

post.setEntity(new StringEntity(json.toJSONString(), StandardCharsets.UTF_8));

try (CloseableHttpResponse response = client.execute(post)) {
    System.err.println("response: " + response);
    System.err.println("response: " + EntityUtils.toString(response.getEntity()));
}
```

6.5. 実証実験結果詳細

今回の実験ではフルノードのみ測定を行っている。SPV ノードは計測にかかわる機能がなかったため、計測には使うことができず、フルノードとの疎通確認を行うのみとなった。

6.5.1. 通信データ量への影響

6.5.1.1. ブロックごとの送受信バイト数

ブロックごとの送受信バイト数を測定した。

- シナリオ 1 (bsafenetlt1)
- シナリオ 2 (bsafenetlt2)
- シナリオ 3 (bsafenetlt3)

の平均送受信バイト数を比較する。

node と拠点の対応は次の通り。

- node1: Toho University
- node2: Keio University（日本）
- node3: University of British Columbia

今回は、たまたま node1 に対しその他ノード（node2, node3）が接続するスター型のネットワーク構成となっている。そのため node1 に通信が集中する偏りのある測定結果となる。今回はノード数が 3 つと少ないため偏りが発生しているが、ノードが多いと理想的な P2P ネットワーク構成となり、送受信データも平準化されると想定される。

また、実験データの投入やマイニングは node1 で実施しているため、node1 からその他ノードへのデータ送信が主になる。

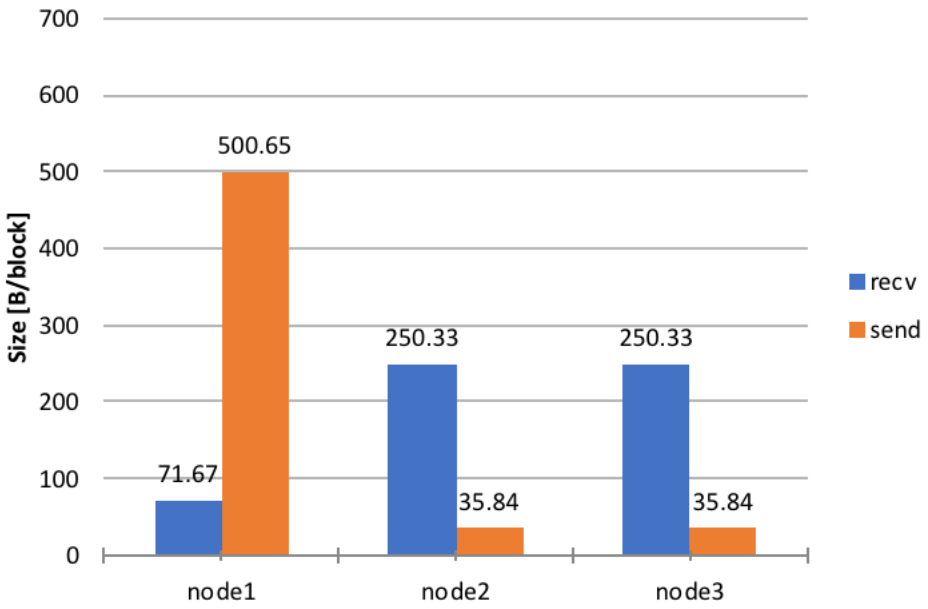


図 6-14 平均送受信バイト数 (シナリオ 1)

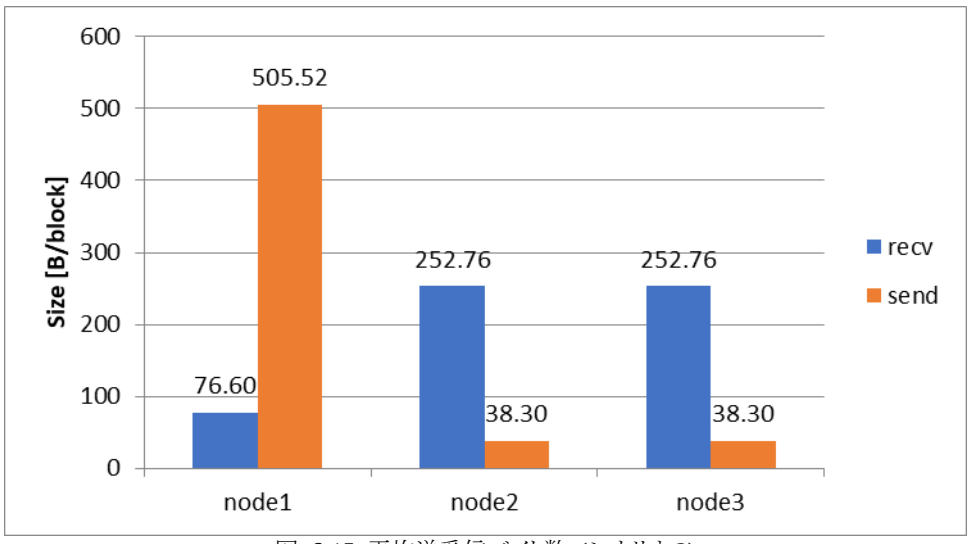
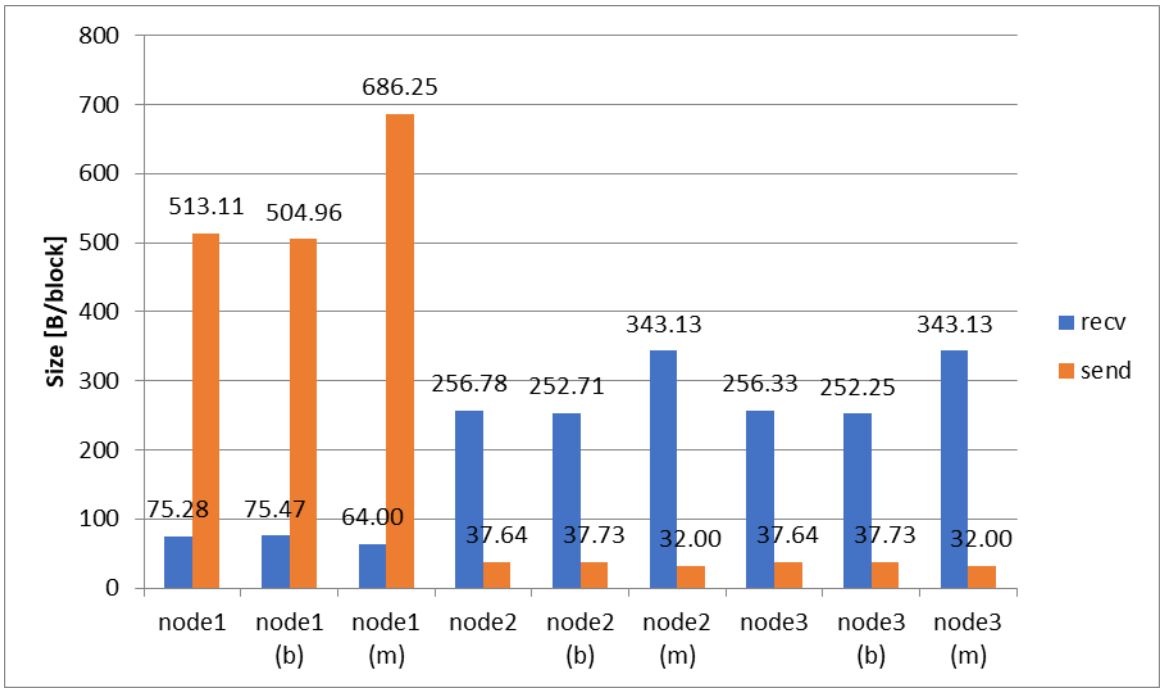


図 6-15 平均送受信バイト数 (シナリオ 2)



凡例:(無印) 全体平均 (b) archiveHash 前 (m) archiveHash 中

図 6-16 平均送受信バイト数 (シナリオ 3)

シナリオ 1～3 ではトランザクションを流さずに、ブロックのみを流している。そのためブロックサイズへの影響が直接結果に現れる。

シナリオ 1 およびシナリオ 2 では、送受信バイト数は接続ピア数あたり 250 バイト程度となった。シナリオ 1 とシナリオ 2 の差分は Proof of Work で使われるハッシュ関数の変更となるが、出力サイズを変更していない(参照:4.1.2 ハッシュ関数 SHA-512 の実装)ため、send/recv に大きな差異は見られない。

シナリオ 3 では、シナリオ 2 に対しブロックの途中から archiveHash を有効化しているため、archiveHash 分の差分が見られる。archiveHash 前の送受信バイト数は接続ピア数あたり 250 バイト程度でありシナリオ 1、シナリオ 2 と差異は見られない。archiveHash 中の送受信バイト数は接続ピア数あたり 340 バイト程度となった。

シナリオ3のarchiveHash中のsend/recvでは、archiveHashの追加によりヘッダーが96バイト増加したため、その分のバイト数が増えていることが分かる。また、送受信されるブロック全体サイズの割合としては、マイナーへの報酬を記録するトランザクションとP2Pレイヤーのプロトコルのオーバーヘッドが大きく、ブロック全体サイズで比較すると3割程度の増加となった。

次に、シナリオ4およびシナリオ5の接続ピア数あたりの送受信バイト数を比較する。シナリオ4は修正前のBSafe.network 版ビットコインであり、シナリオ 5 はそれに加え、ハッシュ関数変更、archiveHash 有効化、ECDSAのパラメータ変更(鍵長増加)を行っている。

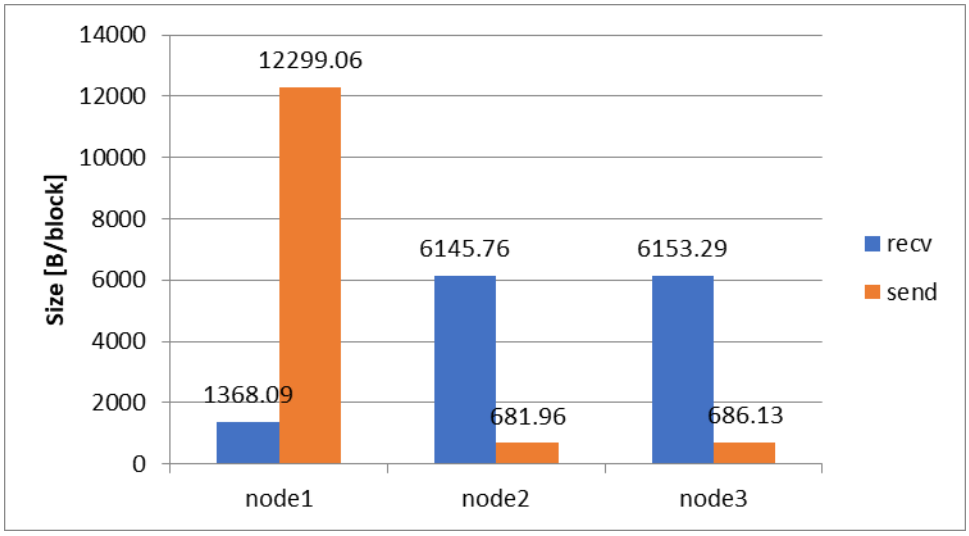


図 6-17 平均送受信バイト数 (シナリオ4)

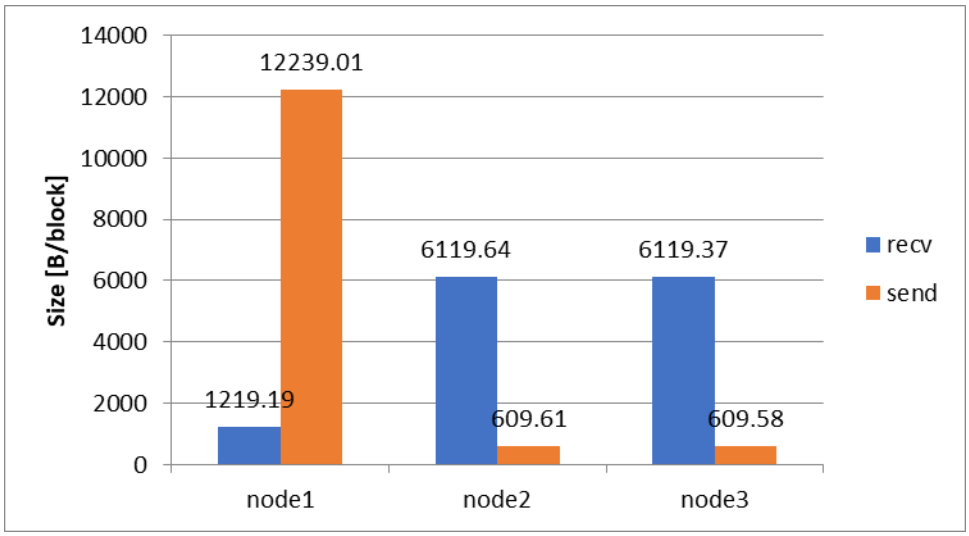


図 6-18 平均送受信バイト数 (シナリオ5)

シナリオ1～3ではブロックのみを流していたが、シナリオ4～5では、ブロックとトランザクションの両方を流している。

シナリオ4およびシナリオ5では、平均送受信バイト数は接続ピア数あたり6,000バイト程度となった。シナリオ1とシナリオ4を比較すると、送受信バイトのうち96%程度がトランザクション部分だと分かる。

これらにより、ブロックサイズに影響を与える修正を行っているものの、計測結果に差分はほぼ見られない。これは、トランザクションを流している実運用に近い環境ではトランザクションがネットワークの送受信の多くを占めるため、署名パラメータ変更伴う署名サイズ増加分はネットワークにほぼ影響しないことが分かる。

6.5.1.2. 通信データ量への影響の考察

想定通りの影響が計測できたものは以下の通りである。

- archiveHash の影響でブロックサイズが増加すること
- トランザクション自体が通信量の多くを占めること

一方、想定通りの影響が計測できなかったものは以下の通りである。

- 署名（ECDSA）パラメータ変更により通信データ量が増加すること
  - 今回の実験条件が良くなく、仮説は正しいと推測される。
  - 実験ではウォレットの移行の際に一個のアドレスを生成し、そこへ全部送金することで新しい署名パラメータに移行させた。
  - そのため、トランザクションに含まれる送金元のアドレスが一個になってしまい、トランザクションあたりの署名の数が一個と少なくなった。（トランザクションあたりの平均署名数は 2）
  - ビットコインの実環境では、ウォレットには複数のアドレスが含まれており、トランザクション中の送金元アドレスもウォレットの中から複数選ばれ、その数だけ署名が必要となる。
  - そのため、トランザクション中の署名数が実情とあっておらず通信データ量に結果として現れていないと推測する。
  - また、計測に使用したトランザクションはランダムで選択したものの、平均 500byte と言われているトランザクションサイズに比べ、5000byte と非常に大きなトランザクションが計測に用いられてしまい、署名パラメータ変更の増加分の影響が少なくなってしまった。

6.5.2. データ量への影響

6.5.2.1. ブロックごとのディスク使用量

ディスク使用量として、以下をノード単位で計測する。

- ディスク使用量の前回のブロックからの増加分
  - "du -s --time $DIR" コマンドで計測

ただし、ビットコインの以下の実装を考慮して、計測結果を確認する必要ある。

- ビットコインの実装では、はじめにある程度大きなディスク領域を割り当てる
- 割り当てが足りなくなった時点で再度大きなディスク領域を一度に確保する方式
- ディスク領域はまとめて確保されるため、たまに上向きのピークが発生する
- アンドゥー情報を時々まとめて消すため、たまに下向きのピークが発生する

ディスク使用量の前回のブロックからの増加分を示すグラフを以下に示す。いずれのグラフも縦軸の単位はキロバイトである。

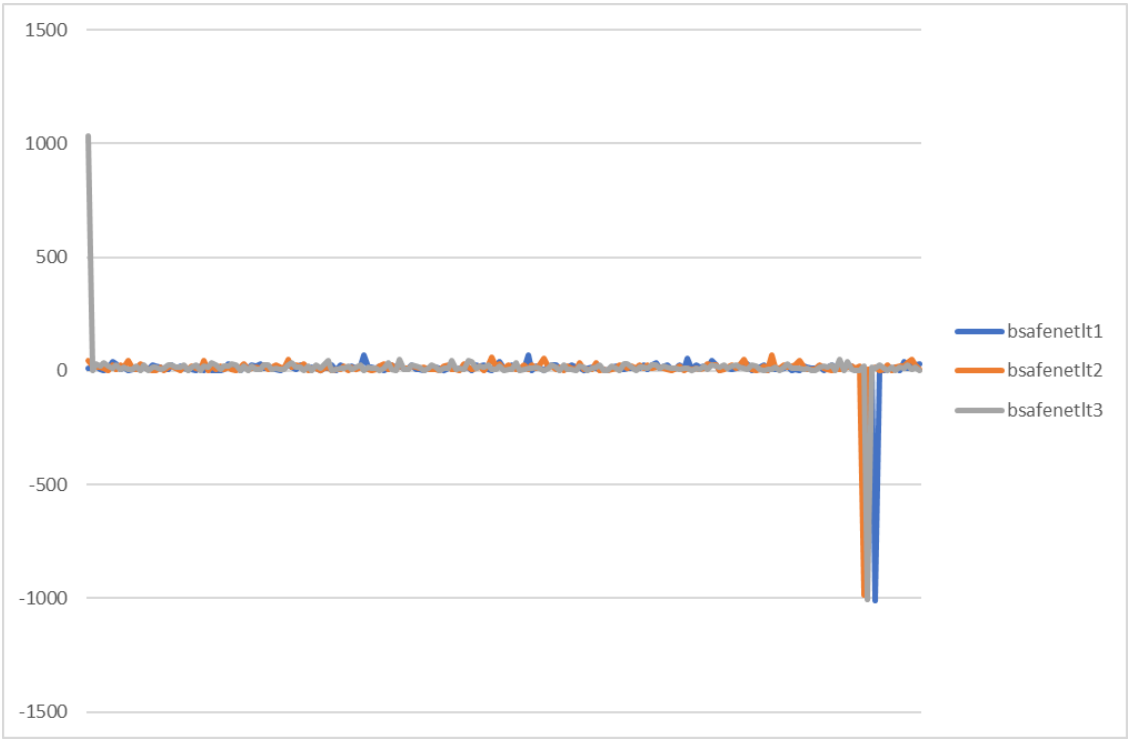


図 6-19 ディスク使用量の推移 (node 1)

図 6-20 ディスク使用量の推移 (node 2)

図 6-21 ディスク使用量の推移 (node 3)

シナリオ 1〜3 では、ビットコインの実装の影響（はじめにある程度大きなディスク領域を割り当てる）により、ディスクサイズの大きな変化が見られなかった。

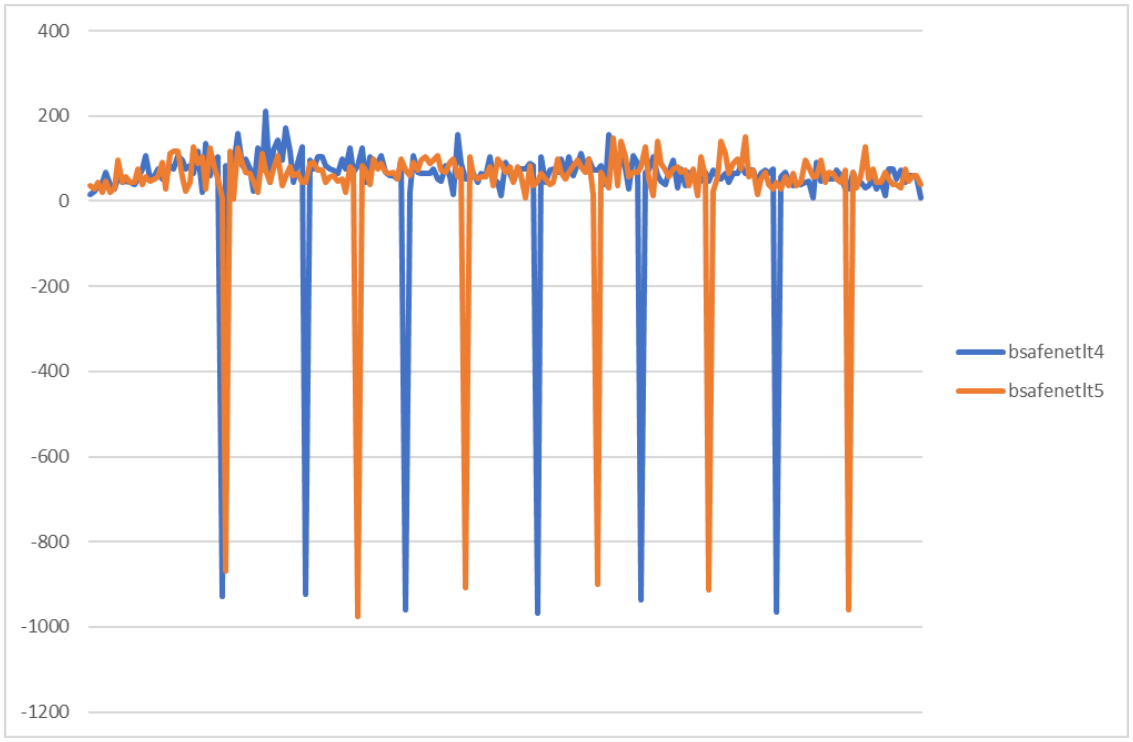


図 6-22 ディスク使用量の推移 (node 1)

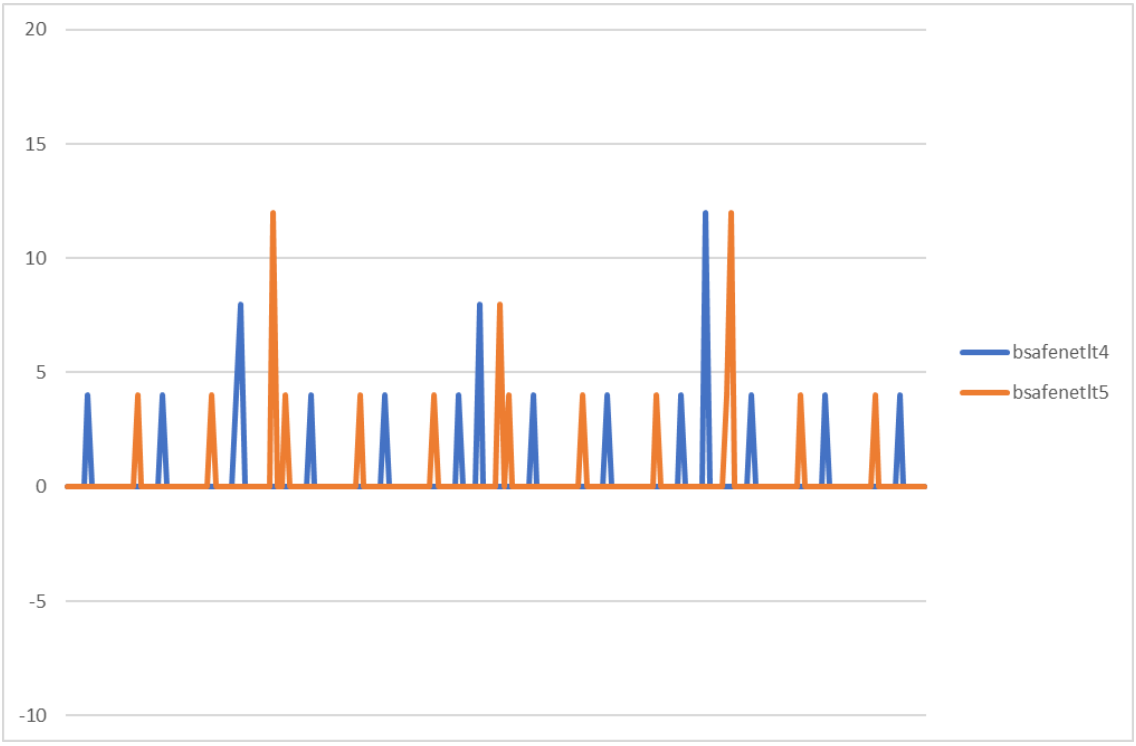


図 6-23 ディスク使用量の推移 (node 2)

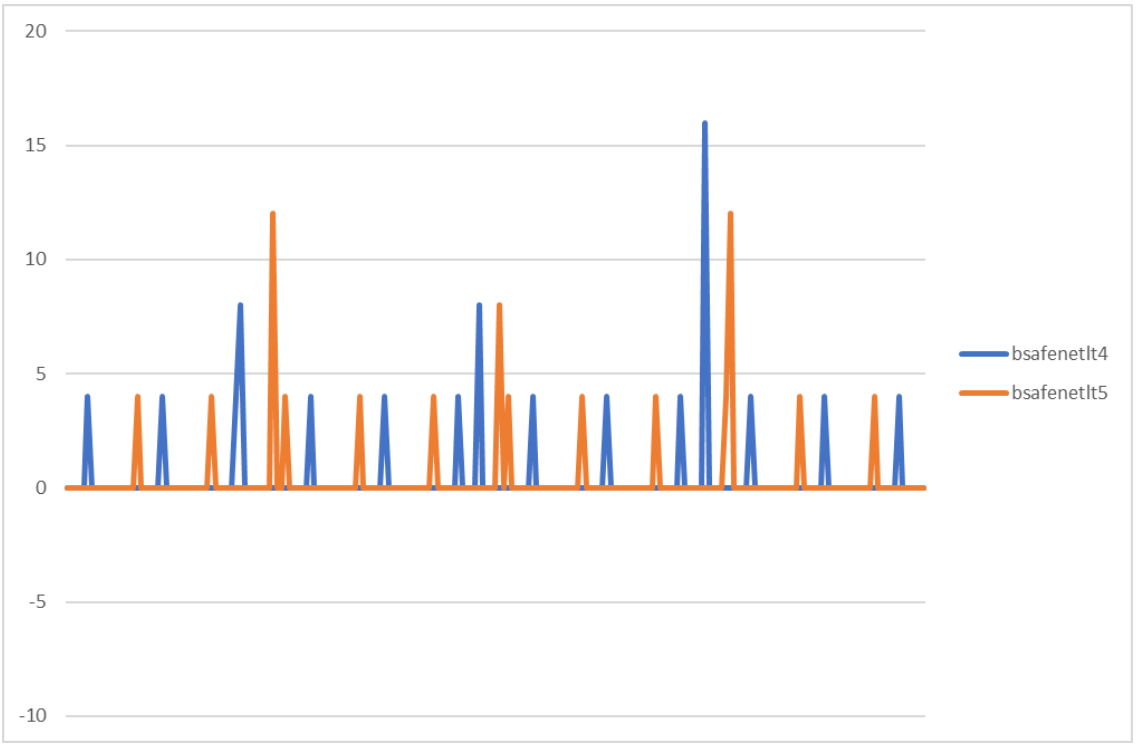
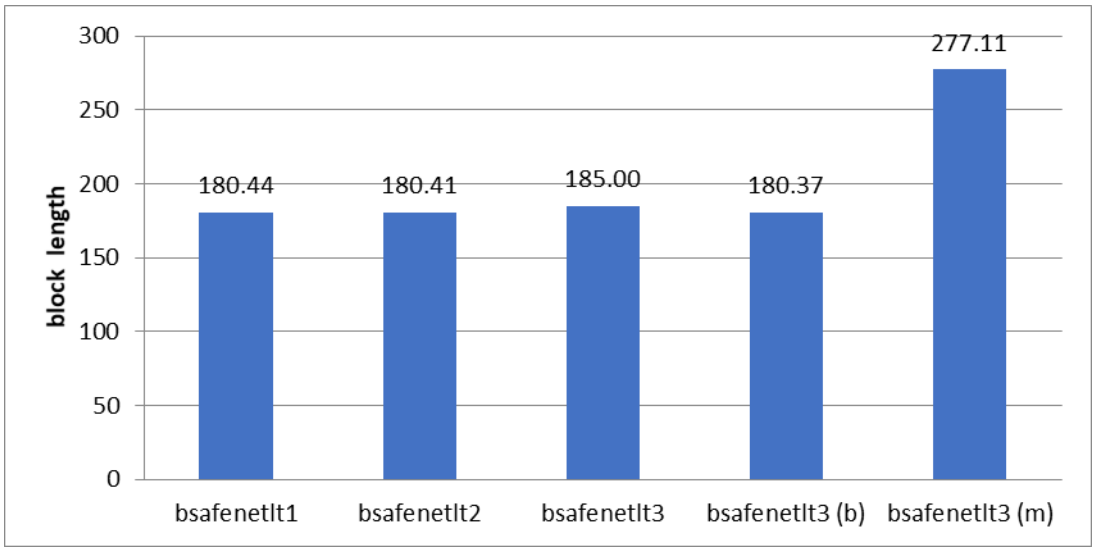


図 6-24 ディスク使用量の推移 (node 3)

シナリオ 4、シナリオ 5 との比較でも、ディスク使用量の推移に差分が見られなかった。

6.5.2.2. ブロックごとのブロックサイズ

ブロックサイズに関する実験結果を示す。各シナリオの平均ブロックサイズを図 13 および図 14 に示す。図中で、bsafenetIt1=シナリオ 1、bsafenetIt2＝シナリオ 2、bsafenetIt3＝シナリオ 3、bsafenetIt4＝シナリオ 4、bsafenetIt5＝シナリオ 5 にそれぞれ対応する。



凡例: (b) archiveHash 前　(m) archiveHash 中

図 6-25 平均ブロックサイズ (シナリオ 1, シナリオ 2, シナリオ 3)

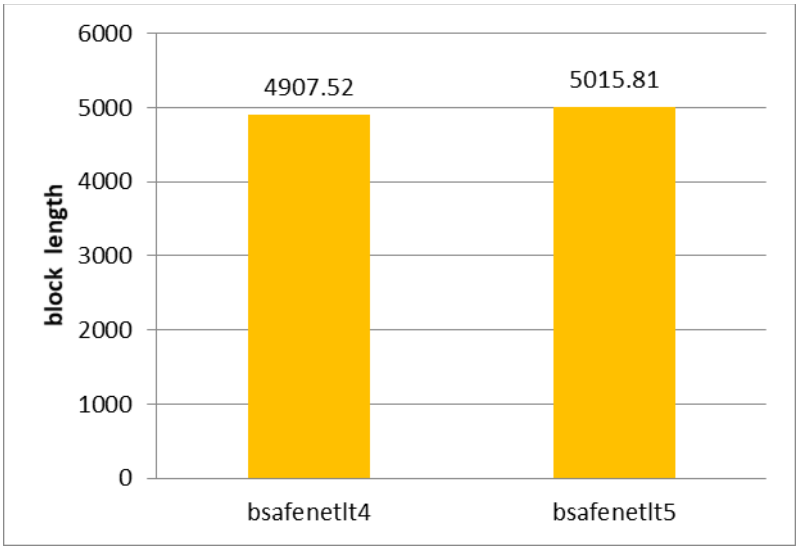


図 6-26 平均ブロックサイズ (シナリオ 4, シナリオ 5)

シナリオ 1(bsafenetlt1), シナリオ 2(bsafenetlt2), およびシナリオ 3(bsafenetlt3)では、archiveHash 前と archiveHash 中のブロック長が 97byte 分増加していることが分かる。これは、ヘッダー長の増加分とほぼ一致する。

シナリオ 4(bsafenetlt4)およびシナリオ 5(bsafenetlt5)では、今回のトランザクションの流量では 100byte 程度増加した。これは署名のビット変更（256bit→384bit）による署名サイズ増加が影響している。署名のサイズ増加に伴い 1 つあたり 48byte 増加する。（詳細は、3.1 目的のデータ量の試算を参照）また、今回は 1 トランザクションあたり、2 つの署名が付与されていたため、合計で 100byte 程度の増加となった。

署名サイズ自体は 5 割ほどサイズが増加（256bit→384bit）したが、トランザクションに占める署名の割合が低いため、トランザクション全体としては 2%程度の増加（シナリオ 4：4907.52→シナリオ 5：5015.81）になった。

次に、各シナリオにおけるブロックサイズの推移を図 6-27 および図 6-28 に示す。

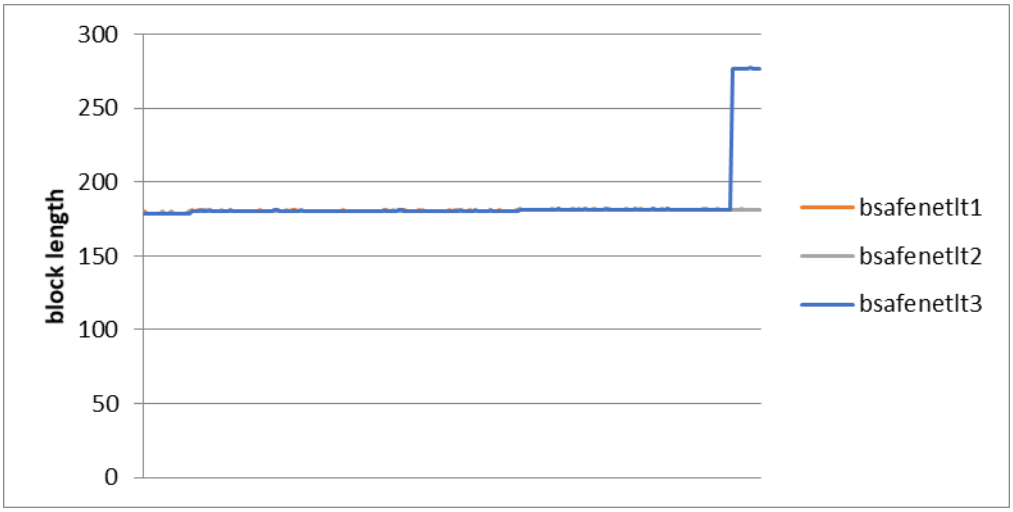


図 6-27 ブロックサイズの推移 (シナリオ 1, シナリオ 2, シナリオ 3)

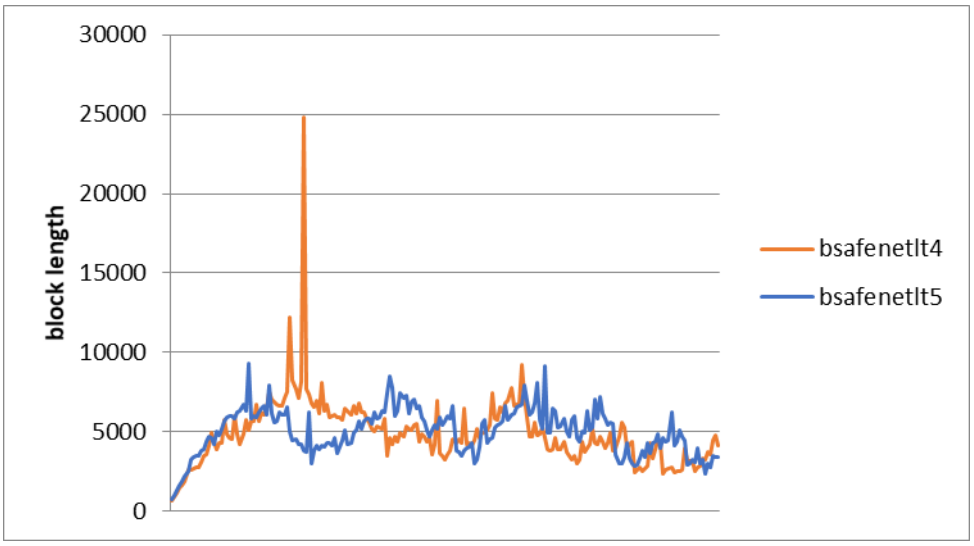


図 6-28 ブロックサイズの推移 (シナリオ 4, シナリオ 5)

6.5.2.3. データ量への影響の考察

想定通りの影響が計測できたものは以下の通りである。

- ブロック単位では、archiveHash によりブロックサイズが増加すること
- ブロック単位では、長期署名技術（archiveHash）による増加の影響が少ないこと

一方、想定通りの影響が計測できなかったものは以下の通りである。

- ブロック単位では、署名（ECDSA）パラメータ変更によりブロックサイズが増加すること
  - 署名（ECDSA）パラメータ変更の影響はより大きいと考えられる。（詳細は、5.1.2 通信データ量への影響の考察　参照）
- archiveHash および署名パラメータの変更が、全体としてのディスク使用量の影響がないこと
  - ビットコインの実装の特性上、まとめてディスク確保を行うが、今回の実験期間では、十分にディスク使用量を使うことができなかったため。
- ディスクへの影響を計測するためには 1GB 程度ディスクを使用する実験を行う必要があると考えられる。
  - 上記のブロック単位の考察と同じく、署名（ECDSA）パラメータ変更の実験時の条件が適切ではなかったことが影響していると考えられる。

6.5.3. 実行速度への影響

6.5.3.1. ハッシュ関数の計算速度

ハッシュ関数の変更による計算速度への影響を計測した。計測したのは以下の 3 種類である。なお、いずれもブロックヘッダーのハッシュ値の計算時間のみであり、merkle tree の計算時間は含まれない。

- BlockHash
  - 古い Proof of Work ハッシュ関数でブロックヘッダーのハッシュ値を計算したときの時間 [マイクロ秒/block]
- BlockHashNew
  - 新しい Proof of Work ハッシュ関数でブロックヘッダーのハッシュ値を計算したときの時間 [マイクロ秒/block]
- BlockHashArchive
  - 新しい Proof of Work ハッシュ関数で archiveHash つきブロックヘッダーのハッシュ値を計算したときの時間 [マイクロ秒/block]

測定結果を表 2 に示す。

表 6-2 ハッシュ関数の計算時間[マイクロ秒/block]

ハッシュ関数	測定回数	最小値	最大値	平均値
BlockHash	1179648	0.846	0.983	0.888
BlockHashNew	1310720	0.733	0.983	0.809
BlockHashArchive	917504	0.539	1.222	1.120

以上の測定結果より、ハッシュ関数変更（BlockHash → BlockHashNew）による計算速度への影響は小さい。

なお、BlockHashArchive は、archiveHash 生成時のみ発生する計算で、ハッシュレートの検証への影響は少ないため、ブロックチェーンのハッシュレート計算全体としては、影響は少ないと考える。

6.5.3.2. archiveHash の計算速度

新しいハッシュ関数を用いて 1MB のバイト列のハッシュ値を計算するのにかかった時間を計測した。これにより、archiveHash の計算速度を測定することができる。

測定結果を表 3 に示す。

表 6-3 archiveHash の計算時間[マイクロ秒/block]

	測定回数	最小値	最大値	平均値
HashArchive	448	1237.076	2797.548	2584.982

6.5.3.3. ECDSA の計算速度

ECDSA の鍵長変更による計算速度への影響を計測した。計測したのは ECDSA 256bit および ECDSA 384bit の署名生成処理である。

測定結果を表 4 に示す。

表 6-4 ECDSA の計算時間[マイクロ秒]

	測定回数	最小値	最大値	平均値
ECDSA 256bit	24576	21.223	46.720	41.296
ECDSA 384bit	1792	282.156	598.073	575.776

これにより、ECDSA の鍵長変更により署名生成速度が 10 倍程度になることが分かった。速度低下の原因としては、鍵長増加よりも、鍵長変更のための手段として外部ライブラリを用いたこと(4.1.3 参照)の影響が大きいと推測される。

1 回の署名生成時間は小さいため、支払い処理のような軽量な利用では問題とならないが、フルノードでのトランザクション署名(ECDSA)の検証には大きな影響が出ると考えられる。

6.5.3.4. 実行速度への影響の考察

想定通りの影響が計測できたものは以下の通りである。

- archiveHash により計算速度には、影響がないこと
- 署名アルゴリズム(ECDSA)を外部ライブラリとする影響で、実行速度に影響があること

一方、想定通りの影響が計測できなかったものは以下の通りである。

- (特になし)