情報処理学会論文誌 Vol.60 No.1 130–146 (Jan. 2019)

トランザクションの本人性を確認できる分散台帳技術の提案

加賀 陽介[1,a)] 藤尾 正和[1] 長沼 健[1] 高橋 健太[1] 村上 隆夫[2] 大木 哲史[3] 西垣 正勝[3]

受付日 2018年4月17日, 採録日 2018年10月2日

概要：中央集権的組織に依存せずに仮想通貨やスマートコントラクトを実現する分散台帳技術が近年注目されている．分散台帳技術が今後安全かつ信頼できる社会基盤として普及するためには，取引の健全性，クレデンシャルの非中央集権的管理，承認権限の独占困難性，という 3 つの特徴を備えることが必要となる．しかしながら，現実に運用されている分散台帳ではこれらの特徴すべてを備えるものはなく，安全かつ信頼できる台帳管理を行うことは依然として困難であった．そこで我々は，これらの特徴をすべて備える基盤として，ユーザと不可分な秘密情報をトラストアンカとしてトランザクションの生成・承認を行う Human Trusted Distributed Ledger（HTDL）を提案し，その要件を整理した．さらに，生体情報を秘密鍵として用いて署名生成を行う Fuzzy signature と，有向非循環グラフに基づき合意形成を行う方式を活用した，HTDL の構成方法を提案し，その構成が HTDL の要件を満たすことを示した．

キーワード：分散台帳技術，ブロックチェーン，バイオメトリクス，Fuzzy signature

Human Trusted Distributed Ledger Technology

Yosuke Kaga[1,a)] Masakazu Fujio[1] Ken Naganuma[1] Kenta Takahashi[1] Takao Murakami[2]
Tetsushi Ohki[3] Masakatsu Nishigaki[3]

Received: April 17, 2018, Accepted: October 2, 2018

Abstract: Distributed ledger technology that realizes cryptocurrency and smart contract without depending on centralized organizations has received attention in recent years. In order for the distributed ledger technology to become popular as a secure and reliable social infrastructure in the future, it is necessary to have three characteristics: transaction soundness, decentralized credential management, and distributed authority. However, any practically operated distributed ledger do not have these three characteristics, and it is still difficult to manage secure and reliable ledgers without a trusted third party. Therefore, for satisfying the characteristics, we propose Human Trusted Distributed Ledger (HTDL), which creates and approves transactions with secret information that is indivisible to users. Moreover, we define the HTDL's requirements for realizing secure and reliable distributed ledger. For the practical construction of the HTDL, we propose a HTDL construction method which utilizes fuzzy signature which can generate a digital signature from user's biometric feature and a consensus method based on a directed acyclic graph, and show that the construction meets HTDL's requirements.

Keywords: distributed ledger technology, blockchain, biometrics, fuzzy signature

1. はじめに

Nakamoto によってビットコイン [1] が 2009 年に提案され，仮想通貨として大きく普及した．このビットコインにおいて，非中央集権的な台帳管理に用いられた技術がブロックチェーンである．ブロックチェーンは，仮想通貨にとどまらず，契約の締結や執行を自動化するスマートコントラクト [2] などを支える技術として注目を集めてい

[1] 株式会社日立製作所
Hitachi, Ltd., Yokohama, Kanagawa 244–0817, Japan
[2] 国立研究開発法人産業総合技術研究所
The National Institute of Advanced Industrial Science and Technology, Koto, Tokyo 135–0064, Japan
[3] 国立大学法人静岡大学
Shizuoka University, Hamamatsu, Shizuoka 432–8561, Japan
[a)] yosuke.kaga.dc@hitachi.com

© 2019 Information Processing Society of Japan

る．このブロックチェーンを一般化した概念であり，より汎用的に分散環境で台帳管理を行う技術を，分散台帳技術（Distrubuted Ledger Technology; DLT）と呼ぶ．このブロックチェーンに代表される分散台帳技術は，電子署名に基づくトランザクション生成と，トランザクションの有効性に関する合意形成を行うコンセンサスから構成される [3]．トランザクション生成では，ユーザが自分の秘密鍵を使って取引内容に対して電子署名を付与することで，トランザクションを生成する．トランザクション生成に用いる秘密鍵は，ユーザやサーバが安全に管理する必要がある．さらに，コンセンサスでは，生成されたトランザクションの有効性および他のトランザクションとの整合性を検証し，改ざんできない形で管理する．たとえば，ブロックチェーンでは，複数のトランザクションから構成されるブロックを生成し，そのブロックをチェーン上につなげていくことで，過去のトランザクションの改ざんや 2 重取引などの不正行為を防止している．

仮想通貨を例にすると，ユーザが生成した任意の秘密鍵をトラストアンカとして取引が行われるため，資産がユーザとは直接結びついていない．このため，秘密鍵の盗難や紛失が発生すると，そのまま資産の盗難や紛失につながる．この仕組みの特徴は，誰がどの資産を所持しているかが紐づけられず，盗難や紛失が発生しても回復できない，現金に近い．一方で，より信頼性が高い資産管理方法として，銀行口座のように最初に個人を特定したうえで，個人に資産を紐づけるものがある．このような資産は，国や銀行が保証を行うことで，盗難や紛失が発生するリスクが低くなっている．分散台帳技術が広範な社会システムを支える基盤として普及するためには，銀行口座のように資産が個人と直接紐づき，資産の盗難や紛失が発生するリスクが低い基盤を，銀行のような中央集権的組織なしで実現することが望ましい．

このような資産が個人と紐づく分散台帳技術を実現するためには，取引の健全性，クレデンシャルの非中央集権的管理，承認権限の独占困難性の 3 つの特徴を満たす必要がある．しかしながら，実際に仮想通貨やスマートコントラクトとして利用されているシステムでは，資産が匿名性の高い秘密鍵に紐づいており，資産が個人と紐づく分散台帳技術が持つべき特徴は満たされていない．

取引の健全性とは，正規ユーザが生成したトランザクションのみが承認され，第三者がなりすましを行って生成した不正なトランザクションは承認されないことを示す．既存の仮想通貨やスマートコントラクトでは，トランザクションへの電子署名により取引の健全性を担保している．トランザクションへ電子署名を付与するためには，ユーザの秘密鍵を使う必要があるため，その秘密鍵が正規ユーザの管理下にある限りは，正規ユーザはトランザクションを生成することができ，第三者がなりすましを行って不正なトランザクションを承認させることが困難である．秘密鍵を管理する手段としては，ネットワークに接続された端末やサーバで管理する手段や，ネットワークに直接接続されていないデバイス内で管理する手段，QR コードなどの印刷物で管理する手段などがあるが，マルウェア感染やデバイス/印刷物の物理的盗難などが発生すると，第三者が秘密鍵を使ってなりすましが行える状態になる．このため，電子署名に基づくトランザクションは，秘密鍵の盗難によるなりすましを原理的に防げない．また，秘密鍵を紛失/破棄した場合は，正規ユーザでもトランザクションを生成できなくなる．たとえばビットコインの場合，発行量全体の 17%～23%がすでに失われており，移動できない資産になっているという報告がある [4]．銀行口座では，被害を補償したりキャッシュカードを再発行することができるが，分散台帳技術では取引を無効化したりアカウントを再発行したりする権限を持つ組織が存在しないため，基本的にこのような対応をとることはできない．例外的に，不正なトランザクションを無効化することに大半の承認者が同意すれば，ハードフォーク [5] と呼ばれる仕組みで不正トランザクションを後から無効化することができるが，非中央集権型の原則に反しているとして，ハードフォークは行うべきでないという意見も多い．これらの課題は，トランザクションを生成する権利を秘密鍵に与えていることに起因する．秘密鍵はデジタルデータであり，端末やサーバにおいてどのように管理したとしても一定の漏えい/紛失のリスクが残る．この課題は，秘密鍵に代わりユーザと不可分な秘密情報をトラストアンカとしてトランザクションを生成することで，解決すると考えられる．ユーザと不可分な秘密情報の一例としては，指紋，顔，虹彩，静脈などに代表される生体情報があげられる．このようなユーザの身体的特徴を秘密情報として用いれば，盗難や紛失のリスクが低減できる．

クレデンシャルとは，本人であることを示すための認証情報であり，秘密鍵や ID・パスワード，生体情報などを指す．クレデンシャルの非中央集権的管理とは，クレデンシャルを信頼できる第三者機関（Trusted Third Party; TTP）が管理するのではなく，ユーザ 1 人 1 人が管理し，任意のユーザ同士が TTP の仲介なしで取引を行うことができることを意味する．分散台帳技術は元々非中央集権型の管理を志向して開発されたため，本来は TTP なしで取引を行うことができる．しかしながら実際には，ユーザが取引所を介して仮想通貨で取引を行う場合，トランザクションを生成するために用いる秘密鍵は，取引所のサーバが管理する場合が多い．この仕組みでは，取引所が実質的に TTP となるため，完全な非中央集権型とはいえない．このため，取引所の内部不正やサイバー攻撃によりユーザの仮想通貨が盗難に遭うリスクがあり，実際に巨額の仮想通貨が盗まれる事件が複数発生している [6], [7]．この課題

© 2019 Information Processing Society of Japan

は，サーバにおいて秘密情報をいっさい管理せず，ユーザが直接トランザクションを生成することで，解決できると考えられる．

承認権限の独占困難性とは，トランザクションを承認する権限が分散しており，少数のユーザもしくは組織が権限を独占することが困難であることを意味する．たとえば，ビットコインは，承認者の計算能力に応じて承認権限が得られる仕組みを採用しており，少数のユーザもしくは組織が権限を独占できないように設計された．しかしながら，実際にはビットコインのマイニングにおけるハッシュレート（計算能力）は上位 3 位までのマイニングプールで 52.3%を占めており，すべて中国のマイニングプールである [8]．ビットコインが採用しているアルゴリズムである PoW（Proof of Work）[1]（2.1.2 項で詳しく説明）は，理論的に 51%以上のハッシュレートを独占した承認者は過去の支払いを取り消して 2 重支払いを行うなどの不正取引を行うことが可能であり，上位のマイニングプールが結託すると実際に不正取引が可能になる．この状況は，権限を独占した組織の意向次第で 2 重支払いによる資産の盗難が可能となるため，リスクが高い状況といえる．このような権限の独占を防止するためには，計算能力のようにコストを掛ければ容易に独占できるリソースに代わり，独占が困難なリソースをトランザクション承認の根拠とすることが必要となる．独占困難なリソースの一例としては，個人がトランザクション承認に使った時間（承認に必要な時間が計算能力に依存しない）や，承認者の数（選挙のように承認が 1 人 1 票限定）があげられる．このようなリソースをトランザクションの承認に用いると，計算能力のように計算機に投資することで容易にリソースを独占することが困難となり，不正なトランザクションが承認されるリスクを低減することができる．

上記の 3 つの要件を満たすためには，TTP を介さずにユーザと不可分な情報を用いて直接トランザクションを生成し，それを承認する権限を分散させて少数の承認者が独占できないようにすることが必要となる．

本論文では，上記の要件を満たす方式として，ユーザと不可分な情報を使ってトランザクションの生成と承認を行うことで，安全かつ信頼できる分散台帳を実現する Human Trusted Distributed Ledger（HTDL）を提案する．

本論文の主な貢献は以下のとおりである．

- 既存の分散台帳技術の課題を解決するため，ユーザと不可分な情報を秘密情報として用いることで，トランザクションとユーザを直接紐づける分散台帳のコンセプトとして，Human Trusted Distributed Ledger（HTDL）を提案し，その定義，処理手順，要件を整理した（3 章）．
- HTDL の具体的な構成方法として，ユーザの生体情報を秘密鍵として用いて電子署名を生成できる Fuzzy signature と，有向非循環グラフ（DAG）を使った分散台帳技術を提案し，HTDL の要件を満たすことを確認した（4 章）．

2. 分散台帳技術

本章では，分散台帳技術の一般的な構成要素について述べ，安全かつ信頼できる分散台帳技術が持つべき特徴を定義する．さらに，実際に運用されている分散台帳の課題をあげる．分散台帳技術の構成要素を**図 1** に示す．

2.1 分散台帳技術の構成要素

分散台帳技術は，仮想通貨 [1] やスマートコントラクト [2] に関する取引を記述したトランザクションを生成し，そのトランザクションを分散台帳に記録し，合意形成を行う．図 1 に示すように，分散台帳技術はトランザクション生成と合意形成（コンセンサス）より構成される [3]．以降では，トランザクション生成とコンセンサスについてそれぞれ説明する．

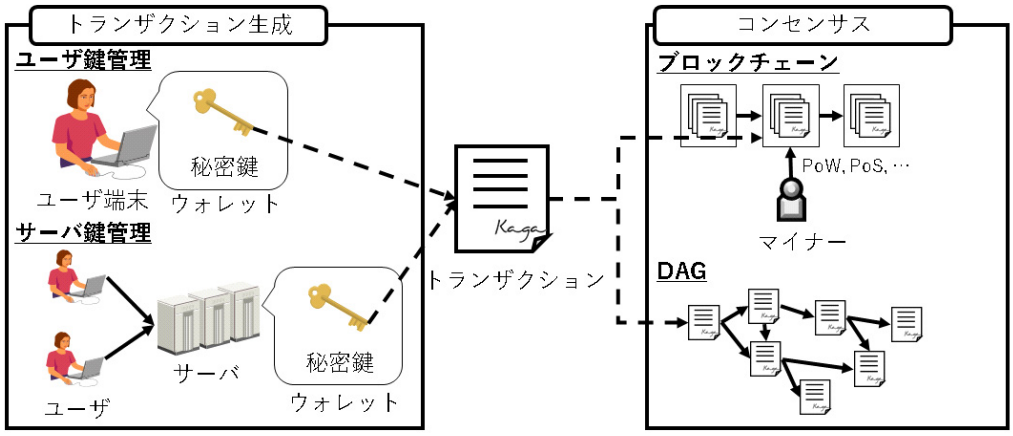


図 1 分散台帳技術の構成要素
Fig. 1 The elements of DLT.

© 2019 Information Processing Society of Japan

2.1.1 トランザクション生成

既存の分散台帳技術では，図 1 左側に示すように，ユーザ端末またはサーバで秘密鍵を管理し，その秘密鍵を使ってトランザクションへ電子署名を付与することで，トランザクション生成が行われる．したがって，トランザクション生成時の安全性・信頼性は秘密鍵の管理方法に依存する．秘密鍵の管理方法は，ユーザ鍵管理とサーバ鍵管理の 2 つに大別され，それぞれ以下のような管理方法を含む [9]．

ユーザ鍵管理

(1-1) Keys in Local Storage
秘密鍵ファイルをそのままローカルのストレージへ格納する方式．ユーザはパスワードの記憶，デバイスの携帯，情報の入力などをいっさい行う必要がないため，利便性が高いものの，端末がマルウェアに感染して秘密鍵が漏えいするリスクが高い．

(1-2) Password-protected Wallets
端末内の秘密鍵をパスワードで暗号化しておく方式．マルウェアで秘密鍵が漏洩した場合でもただちになりすましが成功するわけではないが，辞書アタックやソーシャルエンジニアリング [10] によりパスワードが解読され，秘密鍵が復元されるリスクがある．

(1-3) Offline Storage
USB デバイスのようなネットワークに接続されていない端末内や紙に印刷した 2 次元コードなどで秘密鍵を管理する方式．デバイスや紙に秘密鍵が格納されるため，それらが盗難に遭ったり，2 次元コードが盗撮に遭った場合，秘密鍵が漏洩するリスクがある．

(1-4) Air-gapped Storage
Hardware Security Module（HSM）のようなデバイスの中に秘密鍵を格納し，署名生成をデバイス内で行う方式．ハードウェアウォレットと呼ばれるデバイスがこれに該当する．デバイスから秘密鍵が漏えいするリスクは低いが，デバイスそのものが盗難に遭った場合，他人が不正なトランザクション生成を行うリスクがある．

(1-5) Password-deriverd Keys
PBKDF2 [11] などの方式に基づき，パスワードから秘密鍵を動的に生成する方式．パスワードを忘れた場合，秘密鍵を紛失したときと同様にトランザクションの生成ができなくなる．また，ユーザのパスワードがブルートフォースアタックなどで破られた場合，秘密鍵が復元され攻撃者が不正トランザクションを生成可能となる．

サーバ鍵管理

(2-1) Hosted Wallet (Hot)
ネットワークに接続されたサーバの中に秘密鍵を格納する方式．サーバのマルウェア感染などにより秘密鍵が漏洩するリスクがある．また，サーバのログインにパスワードを使う場合，攻撃者がパスワードを盗みサーバへログインして不正なトランザクションを生成するリスクがある．サーバの管理者は容易に不正トランザクションを生成できるため，鍵を管理するサーバは Trusted Third Party（TTP）であることを前提とする必要がある．

(2-2) Hosted Wallet (Cold)
ネットワークに接続されたサーバから分離されたオフラインのデバイスの中に秘密鍵を格納する方式．Hosted Wallet (Hot) よりも情報漏えいが起きにくいものの，ログインパスワードが漏洩した場合には，同様に不正なトランザクションが生成されるリスクがある．サーバの管理者は容易に不正トランザクションを生成できるため，上記と同様に鍵を管理するサーバは Trusted Third Party（TTP）であることを前提とする必要がある．

(2-3) Hosted Wallet (Hybrid)
暗号化された秘密鍵をサーバで管理し，クライアント側で復号してトランザクションを生成する方式である．サーバのマルウェア感染やサーバ管理者の内部不正に対して安全性が向上するが，秘密鍵の復号に用いるパスワードが漏洩した場合，不正トランザクションが生成されるリスクがある．

以上の方法は，トランザクション生成のために必要となる秘密鍵を安全に管理するために，パスワードによる暗号化やオフライン端末などを駆使して安全性を向上させている．しかしながら，端末やサーバで秘密鍵のデータを管理する以上，端末そのものやパスワードの盗難，サーバ管理者の不正などを想定すると，十分な安全性を実現する方法は知られていない．たとえば，仮想通貨の取引所ではサーバ鍵管理を行っているが，近年複数の取引所から多額の仮想通貨が盗まれる事件が発生している [6], [7]．これは，取引所の “Hosted Wallet (Hot)”（通称ホットウォレット）で運用された秘密鍵が漏洩したことが原因の 1 つといわれており，安全性を高めるため，“Hosted Wallet (Cold)”（通称コールドウォレット）で運用することが推奨されている．ただし，上記で示したとおり，コールドウォレットでもユーザパスワードの盗難やサーバ管理者自身の不正が発生するとなりすましが容易に行えるため，十分に安全とはいい難い．この秘密鍵管理の安全性は社会問題となっており，日本は 2017 年 4 月に改正資金決済法を施行し，仮想通貨交換業者を登録制にし，安全性に課題がある取引所の営業を認めない方針とした [12]．しかし，安全性がサーバ運用に依存する仕組みでは，完全に不正を防止するのは困難であり，安全性がサーバ運用に依存しない，すなわち TTP を前提としないシステムが望ましい．

2.1.2 コンセンサス

コンセンサスは，トランザクションが有効なものであるか

© 2019 Information Processing Society of Japan

否かに関して合意形成を行うアルゴリズムであり，図 1 右側に示すように有効なトランザクションをブロックチェーンまたはそれ以外の方式に基づき構造化して管理する．代表的なコンセンサス方式は以下のとおりである．

ブロックチェーン

(1) Proof of Work（PoW）[1]
ビットコインで採用されているコンセンサスアルゴリズムである．承認者（マイナ）はトランザクションをまとめたブロックを生成し，ブロックに関する計算問題を解く．この計算問題を解く作業をマイニングと呼ぶ．この計算問題は，ビットコイン [1] の場合，ブロックの内容とノンスと呼ばれる数字から算出されるハッシュ値が一定未満となるノンスを見つける問題である．ノンスを見つけたら，ブロックにそのノンスを付与してブロックチェーンネットワークへ送信する．他のマイナは，そのブロックを検証し，計算結果が正しくかつ他のブロックと矛盾がなければ，そのブロックをブロックチェーンに追加して次のブロックの生成を行う．この手法により，二重支払いなどの不正トランザクションは排除され，台帳管理を行う TTP を前提とせずに安全な取引を行うことができる．ただし，PoW ではノンスを見つける計算量に応じて承認の権限が与えられるため，効率的な計算が可能なマイニング専用のハードウェアである Application Specific Integrated Circuit（ASIC）[13] を用いたマイニングが行われており，一部の組織が大きな権限を持っている．

(2) Proof of Stake（PoS）[14]
PoW の欠点の一部を解消するために導入されたコンセンサスアルゴリズムであり，仮想通貨 Ethereum [2] に採用されている．PoW が計算量のみを根拠に承認を行うのに対し，PoS ではマイナが保持する資産量を加味する．具体的には，ノンスを求める計算問題の難易度を，資産量に応じて下げることで，計算量を削減している．PoW よりも必要な計算量が抑えられるものの，多くの資産を持つマイナが有利になるため，依然として一部のマイナに権限が集中するという課題がある．PoS の派生として，Proof of Activity [15]，Proof of Importance [16] などがあるが，本論文では PoS の一種と見なす．

(3) Proof of Authority（PoA）[17]
PoW，PoS のように一般のマイナが承認を行うのではなく，あらかじめ権限を与えたサーバがブロックの承認を行うコンセンサスアルゴリズム．権限を与えられたサーバが実質的に TTP となるため，コンセンサスの安全性がサーバ運用に依存する．

(4) Proof of Human work（PoH）[18]
PoW, PoS のようにノンスを求める計算問題を解く代わりに，人間がコンピュータには識別できない文字の解読などを行う問題 Completely Automated Public Turing Test to tell Computers and humans Apart（CAPTCHA）を解くことでブロックを承認するコンセンサスアルゴリズム．ブロックを承認する権限が計算能力に依存せず，人間の作業に依存するため，権限の独占が困難となるが，まだ理論検討の段階であり，安全に実装するためには課題が残されている．

ブロックチェーン以外

(5) Directed Acyclic Graphs（DAG）[19]
ブロックチェーンと異なり，トランザクションをまとめたブロックを作成せず，トランザクションをノードとする有向非循環グラフ（Directed Acyclic Graphs; DAG）を構成するコンセンサスアルゴリズムである．トランザクション生成時に，トランザクション生成者はランダムに選択された複数のトランザクションを検証し，矛盾がなければ自分のトランザクションを検証したトランザクションにつなげる．この手順をすべてのトランザクション生成者が繰り返すことで，矛盾がないトランザクションには承認が集まり，二重支払いなどの矛盾をかかえたトランザクションは承認されないため，矛盾を排除した合意形成を行うことができる．DAG を使ったコンセンサスアルゴリズムとして，Tangle [19] がある．この Tangle では，承認作業をトランザクション生成者自身が行うため，マイナが存在せず，トランザクションに対する報酬が発生しない．ただし，攻撃者が計算能力を独占した場合，PoW と同様に権限が集中し，2 重支払いなどの不正トランザクションが承認されるリスクがある．

2.2 分散台帳技術が持つべき特徴と課題

次に，分散台帳を安全かつ信頼できる取引基盤として普及させるために，分散台帳技術が持つべき特徴と課題をあげる．仮想通貨を例にすると，取引はユーザが生成した任意の秘密鍵をトラストアンカとし，秘密鍵で電子署名を生成することでトランザクションを生成する．さらに，電子署名を公開鍵で検証することでトランザクションの検証を実施し，PoW などのコンセンサスアルゴリズムに従いトランザクションの承認を行う．このとき，トランザクションは秘密鍵と結びついており，正規ユーザの秘密鍵を不正に入手した他人は容易に正規ユーザになりすますことができる．このため，秘密鍵の盗難や紛失が発生すると，そのまま資産の盗難や紛失につながる．この特徴は，誰がどの資産を所持しているかは記録されず，容易に盗難や紛失が発生する点で，現金に近い．実際に，仮想通貨の交換業者から巨額の仮想通貨が盗まれる事件が複数発生している [6], [7]．

一方で，より信頼性が高い資産管理方法として，銀行口座のように最初に個人を特定したうえで，個人に資産を紐

© 2019 Information Processing Society of Japan

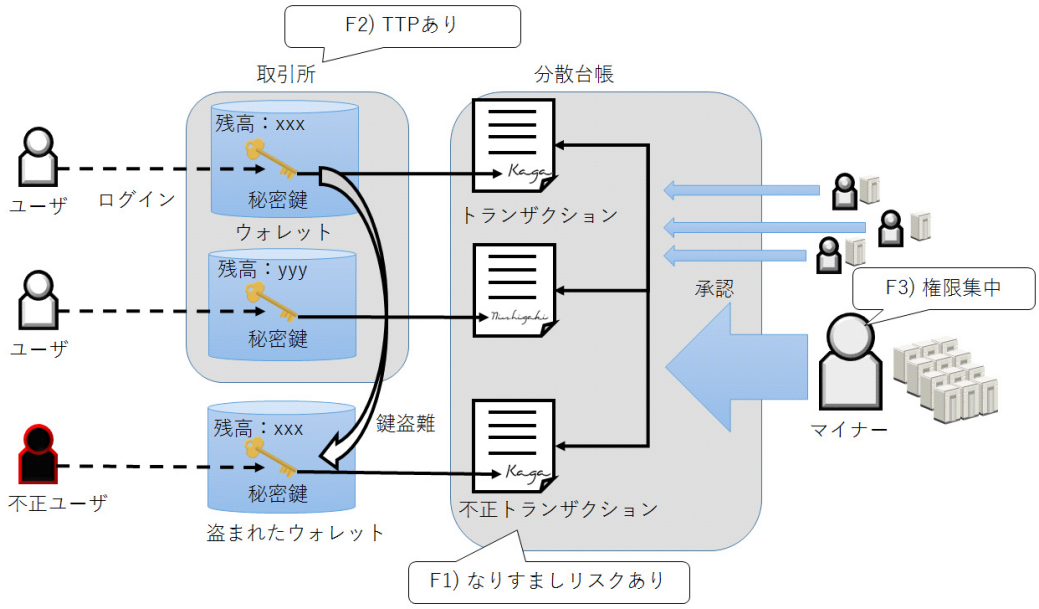


図 2 既存の分散台帳技術の課題
Fig. 2 The challenges of conventional DLT.

づけるものがある．このような資産は，銀行や国がその保有を保証することで，盗難や紛失が発生するリスクが低くなっている．分散台帳技術が広範な社会システムを支える基盤として普及するためには，銀行口座のように資産が個人と確実に紐づき，資産の盗難や紛失が発生するリスクが低い基盤を非中央集権的に実現することが望ましい．

このような分散台帳技術を実現するためには，以下の特徴を備えることが必要となる．

(F1) 取引の健全性 分散台帳を安全に運用するためには，正規ユーザが生成したトランザクションが承認され，正規ユーザ以外が生成したトランザクションは承認されないことが必要となる．トランザクション生成に必要となる秘密鍵などのクレデンシャルが漏えいした場合，不正トランザクションが生成されるリスクがある．また，クレデンシャルを紛失した場合，その後正規ユーザがトランザクションを生成することができなくなる．このため，クレデンシャルは漏えい/紛失するリスクが十分低い方法で管理する必要がある．

(F2) クレデンシャルの非中央集権的管理 安全かつ信頼できる分散台帳を運用するためには，TTP を置かず，各々のユーザが自分の秘密鍵を管理する形が必要となる．しかしながら，2.1.1 項で述べたように，現状運用されている分散台帳基盤は取引所のような仲介組織によって運用されることが多く，実質的に中央集権的な管理になっている．この運用では，仲介組織がサイバー攻撃を受けた場合や，内部不正が発生した場合，不正なトランザクションが生成されてしまう．

(F3) 承認権限の独占困難性 2 重支払いなどの不正取引を防止するためには，トランザクションを承認する権限を分散させ，少数のユーザや組織に権力が集中するのを防ぐ必要がある．しかしながら，2.1.2 項で述べたように，PoW に代表されるコンセンサスアルゴリズムは計算能力に投資することで権力を集中させることができる．少数のユーザや組織による独占が困難なリソースを使ったコンセンサスが必要となる．

上記で述べた分散台帳の課題を，**図 2** にまとめた．

3. Human Trusted Distributed Ledger の提案

本章では，2 章で述べた分散台帳技術が持つべき特徴を満たす基盤として，トランザクションとユーザを不可分な形で紐づける Human Trusted Distributed Ledger（HTDL）を提案する．HTDL では，ユーザの身体的行動的特徴である生体情報をクレデンシャルとして用いることで，安全性の高い分散台帳技術を実現する．

3.1 Human Trusted Distributed Ledger のコンセプト

まず，HTDL のコンセプトを説明する．既存の分散台帳技術では，本人であることを示すトラストアンカとして秘密鍵を生成し，それを端末やサーバで管理していた．このため，秘密鍵の漏洩や紛失が発生した場合に，資産の盗難や凍結が発生していた．これに対して，HTDL ではユーザ本人とは不可分の情報を秘密鍵として用いてトランザクション生成を行うことで，ユーザとトランザクションを直接紐づける．ユーザ本人と不可分な情報の例としては，指紋，顔，虹彩，静脈などの生体情報があげられる．これらの情報は，人間の身体的または行動的特徴を表す情報であり，ユー

© 2019 Information Processing Society of Japan

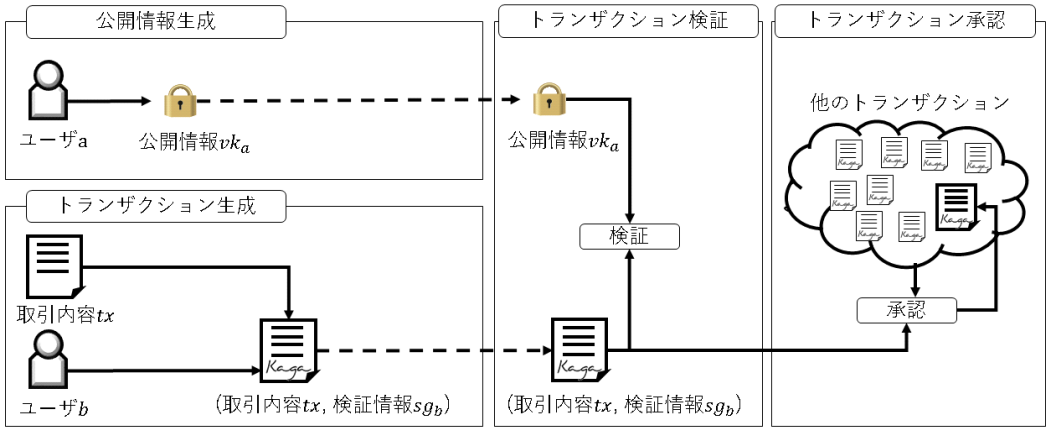


図 3 HTDL の定義
Fig. 3 The definition of the HTDL.

ザ本人と不可分な情報である．また，人間 1 人 1 人が異なる生体情報を持っているため，ユーザ本人の生体情報が他人の生体情報と一致する可能性はきわめて低く，分散台帳におけるトラストアンカとして用いることができる．これにより，端末やサーバにおける秘密鍵の管理が不要となり，資産の盗難や凍結のリスクを低減させることができる．

さらに，計算能力や資産のように少数の組織が投資を行うことで独占が可能となるリソースに代わり，計算能力や資産の増強では独占困難なリソースを使ってトランザクションを承認することで，信頼性の高い分散台帳技術を実現する．独占困難なリソースの例としては，PoH のように人間の物理的な作業により時間あたりの承認数を制限する方法や，選挙のようにトランザクション承認に対する寄与を 1 人 1 票までに制限する方法が考えられる．

3.2 HTDL を実現するシステムの処理概要

本節では，HTDL を実現するシステムの処理概要について説明する．HTDL では，既存の分散台帳のように秘密鍵と公開鍵を生成して秘密鍵を安全に管理するのに代わり，ユーザ本人と不可分な秘密情報とそれに対応する公開情報を生成し，ユーザ本人がその秘密情報を使ってトランザクション生成を行う．つまり，秘密鍵などのクレデンシャルを端末やサーバで管理するのではなく．ユーザ自身と不可分な情報がクレデンシャルの役割を果たす．具体的な処理手順を図 **3** に示す．各処理の内容は以下のとおりである．

公開情報生成

Step1 ユーザ a と不可分な秘密情報を秘密鍵としたときの公開鍵に相当する情報として，公開情報 vk_a を生成する．

Step2 公開情報 vk_a をユーザ a のアドレスとして公開する．

トランザクション生成

Step1 トランザクションの取引内容 tx（仮想通貨であれば，送金先，金額など）を生成する．

Step2 ユーザ b が取引内容 tx に対する検証情報 sg_b を生成する．

Step3 (tx, sg_b) をトランザクションとして生成する．

トランザクション検証

Step1 トランザクション (tx, sg_b) を入手し，公開情報 vk_a を使って検証情報 sg_b の検証を実施する．

Step2 ユーザ a とユーザ b が同一人物であれば検証に成功し，異なる人物であれば検証に失敗する．

トランザクション承認

Step1 検証情報 sg_b の検証に成功した場合，他のトランザクションとの整合性を確認する（たとえば，二重支払いが存在しないか）．

Step2 他のトランザクションとの整合性が確認できたら，トランザクション (tx, sg_b) を正しいトランザクションとして承認する．

HTDL では，ユーザに対応する公開情報を生成する．この公開情報により，ユーザが生成した検証情報が本人のものであるかどうかを検証することができる．以上の手順により，秘密鍵やパスワードなどの盗難/紛失のリスクがあるクレデンシャルに安全性を依存せず，ユーザ自身が直接トラストアンカとなる分散台帳を実現することができる．

3.3 Human Trusted Distributed Ledger の要件

HTDL が，2.2 節で示した分散台帳が持つべき特徴を満たすために，必要となる要件を定義する．HTDL の概要と定義した要件を図 **4** に示す．具体的な要件の内容は以下のとおりである．

R1) トランザクションの健全性

正規ユーザが生成した検証情報は承認され，正規ユーザ以外が生成した検証情報は承認されない．この要件を満たすことができれば，**F1) 取引の健全性**を満たすことができる．この要件は，さらに詳細化して R1-1)

© 2019 Information Processing Society of Japan

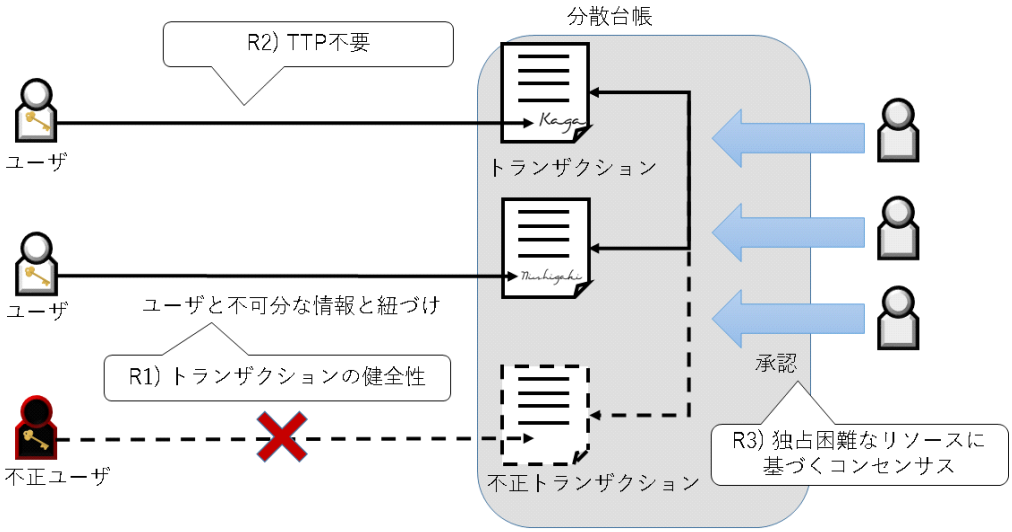


図 4 HTDL の要件
Fig. 4 The requirements for our HTDL.

と R1-2) に分割することができる．

R1-1) クレデンシャルの秘匿性
トランザクション生成に必要なクレデンシャルが，正規ユーザと不可分な形で管理される．したがって，端末やサーバで秘密鍵を管理する必要がなく，端末やサーバからの情報漏洩や管理者の内部不正を想定したとしても，他人がなりすましを行うことが困難である．

R1-2) 検証情報の健全性
正規ユーザが生成した検証情報は，十分高い確率で検証に成功し，正規ユーザ以外のユーザが生成した検証情報は，十分高い確率で検証に失敗する．

R2) TTP 不要
公開情報生成，トランザクション生成，トランザクション検証，トランザクション承認をすべて任意の端末で実行することができ，かつ端末やサーバに保存されているいっさいの秘密情報を必要とせず，公開情報のみを参照する．これにより，HTDL は TTP を必要とせず，**F2) クレデンシャルの非中央集権的管理**を満たすことができる．

R3) 独占困難なリソースに基づくコンセンサス
計算量や資産のように投資すれば容易に独占できるリソースに代わり，単純な投資では独占困難なリソースをコンセンサスに適用する．これにより，**F3) 承認権限の独占困難性**を満たすことができる．

上記の 3 つの要件を満たすことができれば，F1〜F3 に示した分散台帳技術が持つべき特徴を満たすことができ，安全かつ信頼できる分散台帳技術を実現することができる．次章では，HTDL を構成する具体的な方法について提案する．

4. Human Trusted Distributed Ledger の具体的な構成法

3 章で提案した HTDL に用いる，ユーザ本人と不可分な秘密情報の一例として，指紋，顔，虹彩，静脈などの生体情報があげられる．本章では，3 章で提案した HTDL の構成方法として，ユーザの生体情報をクレデンシャルとして用いて分散台帳を実現する方法を提案する．本方式では，生体情報をクレデンシャルとして用いる．この生体情報は，ユーザから取得するたびに誤りを含むデータであり，一般的な電子署名における秘密鍵として用いることができない．したがって，秘密鍵に一定の誤りを許容するバイオメトリック署名と呼ばれる方式を用いる必要がある．このバイオメトリック署名をアルゴリズムレベルで実現する方式としては，Fuzzy extractor [20] や Fuzzy signature [21], [22] がある．さらに，生体情報に基づいた署名生成をシステムレベルで実現する方法としては，安全な個人認証プロトコルである FIDO（Fast IDentity Online）[23] がある．4.1 節では，署名生成を行う際に補助情報と呼ばれる情報が不要であり，秘密鍵を端末/サーバで管理することが不要である，Fuzzy signature を使ったトランザクション生成方法を説明する．

また，独占困難なリソースに基づくコンセンサスの構成方法の一例としては，単位時間あたりの承認数を制限する方法と個人あたりの承認数を制限する方法が考えられる．PoW における計算量や PoS における資産量は，投資により独占を行うことができるが，時間はすべてのユーザに平等に与えられたリソースであるため，単位時間あたりの承認数を制限した場合，承認権限の独占は困難である．さらに，承認回数を選挙のように 1 人 1 票に限定することができれば，きわめて民主的に承認を行うことができるように

© 2019 Information Processing Society of Japan

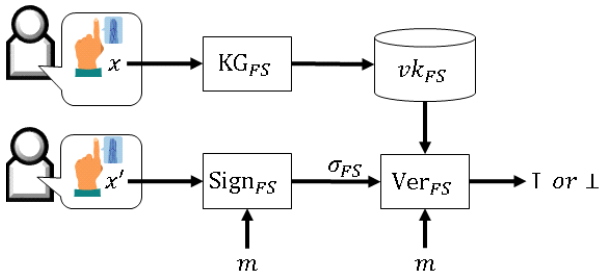


図 5 Fuzzy signature の処理
Fig. 5 The procedures of fuzzy signature.

なり，承認権限を独占するためには，実際に大半の承認者の同意を得る必要がある．4.2 節では，Fuzzy signature を用いてこのようなコンセンサスを実現する方法について説明する．

4.1 Fuzzy signature [21], [22] を用いたトランザクション生成

まず，HTDL のトランザクション生成に用いる Fuzzy signature について説明する．Fuzzy signature は，Public Key Infrastructure（PKI）における秘密鍵として，誤差が含まれるデータを適用し，公開鍵生成時と署名生成時のデータの誤差が一定未満であれば署名検証に成功するように設計された署名アルゴリズムである．Fuzzy signature の処理を**図 5** に示す．Fuzzy signature の処理手順は以下のとおりである．

(1) ノイズを含むデータ x を取得する．
(2) データ x を鍵生成関数 KG_{FS} へ入力し，検証鍵 vk_{FS} を生成する．
(3) ノイズを含むデータ x' を取得する．
(4) データ x'，平文 m を署名生成関数 $Sign_{FS}$ へ入力し，署名値 σ_{FS} を生成する．
(5) 検証鍵 vk_{FS}，署名値 σ_{FS}，平文 m を検証関数 Ver_{FS} へ入力し，x と x' との誤差が一定未満であれば，$\top$（検証成功），それ以外の場合は $\bot$（検証失敗）を出力する．

文献 [21], [22] では，この Fuzzy signature の安全性について証明を行っており，選択文書攻撃に対する存在的偽造不可（Existential UnForgeability against Chosen Message Attacks; EUF-CMA）であることが示されている．このため，データ x に十分近い x' を知っているユーザ以外は，署名値を偽造して検証に成功することができない．また，文献 [21] では，Fuzzy signature の入力データとして，指紋などの生体情報を用いることで，ユーザの生体情報を秘密鍵として用いる PKI である Public Biometrics Infrastructure（PBI）を実現することができるとしている．この PBI を使って分散台帳技術におけるトランザクション生成を行えば，秘密鍵を使わずに直接人間とトランザクションを紐づけることが可能となり，資産の盗難や紛失のリスクを抑えることが可能となる．

4.2 Fuzzy signature を使ったコンセンサスの検討

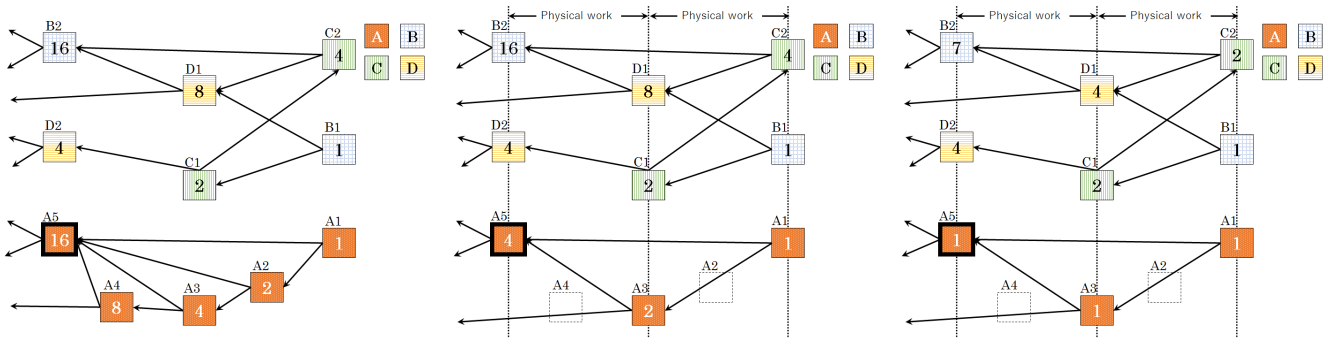
本節では，Fuzzy signature を活用して分散台帳のコンセンサスを行う方法について述べる．HTDL における独占困難なリソースに基づくコンセンサスの実現方法としては，個人の単位時間あたりの承認数を制限する方法と個人あたりの承認数を制限する方法が考えられる．

(1) 個人の単位時間あたりの承認数を制限する方法
PoH [18] のように，計算機ではなく人間が行う作業をトランザクション承認に適用すると，計算能力に依存せず個人の単位時間あたりの承認数を制限することができる．生体情報を秘密情報としてトランザクション生成を行う場合も人間に生体情報を提供する作業を求めるため，PoH と同様に計算能力に依存せず個人の単位時間あたりの承認数を制限することができる．たとえば，生体情報の取得に 1 秒かかる場合は，1 人のユーザは 1 時間に最大 3,600 回しかこの作業を実施できない．このため，この物理的作業の総量を増加させるためには，多数のユーザを雇い作業に従事させる必要がある．人の雇用は，単純に計算機の購入で増強できる計算能力と比較して，コストが高く独占が困難である．このように人間の物理的作業を用いてトランザクションを承認する方式を Proof of Physical work（PoP）と名付ける．

(2) 個人あたりの承認数を制限する方法
従来の秘密鍵を使う承認方法では，同一ユーザが任意の個数の秘密鍵を持つことができるため，個人あたりの承認数を制限するのは困難であった．これに対して，生体情報を秘密情報として用いる場合は，ユーザは決められた個数の生体情報しか持てない．たとえば，指紋では 10 個，顔では 1 個のように，生体情報の数は人間の身体的な制約に依存する．よって，同一の生体情報によるトランザクション承認を 1 度に制限することができれば，この個人あたりの承認数の制限が実現する．このように事後に同一人物の署名を特定するやり方は，Lazy signature [24] として定式化されている．この Lazy signature をトランザクションの承認へ適用した場合，1 人のユーザの特定のトランザクションに対する承認数を制約することが可能になるため，物理的作業よりもさらに強い制約となり，承認権限の独占が困難となる[*1]．このように生体情報を用いて個人の

[*1] ただし，生体情報は取り替えられない情報であるため，秘密情報の更新回数に制約が発生する．たとえば，指を負傷して指紋を更新する場合，最大 10 回までしか更新ができない．秘密情報の数を増加させる方式として，複数の生体情報（指紋と顔など）を選択的に利用できる方式があげられるが，個人あたりの承認数の制約を緩めることにつながるため，個人が所有できる秘密情報の数はユースケースの要件に応じて適切に設定する必要がある．

© 2019 Information Processing Society of Japan



(a) Tangle (b) PoP (c) PoB

図 6 各方式における累積荷重の例
Fig. 6 Examples of cumulative weights in each method.

承認数を制約する方式を，Proof of Biometrics（PoB）と名付ける．

ここで，上記の PoP および PoB を実際に運用する際の制約について考える．ブロックチェーンはブロックの承認にインセンティブを設けることで，多数の承認者を集め，システムの信頼性を担保している．このため，ブロックチェーンの承認に上記のような制約を設けることを考えると，マイニングをするインセンティブが小さくなるため，承認者が少なくなり，結果として信頼性が低下する可能性がある．そこで我々は，トランザクション生成者が同時にトランザクション承認を行う，DAG ベースのコンセンサスへ Fuzzy signature を適用する．トランザクション生成者が承認を行う場合は，上記のように承認者が少なくなることを考える必要がなく，信頼性の高いシステムを実現することができる．まず，DAG を使ったコンセンサスアルゴリズムの代表例である，Tangle [19] に基づく方式の手順を説明する．なお，Tangle は，IoT デバイスにおけるマイクロペイメントを目指した仮想通貨 IOTA に用いられているが，適用範囲は IoT デバイスに限ったものではない．ここでは，PC やスマートフォンなどから行う取引を承認するためのアルゴリズムとして Tangle を用いる．この Tangle はトランザクション生成者自身がトランザクションの承認を行うモデルであるため，Tangle の処理手順は，トランザクション生成，トランザクション検証，トランザクション承認を含む．

Tangle の処理手順

Step1 トランザクションの取引内容（仮想通貨であれば，送金先，金額など）を生成する．

Step2 トランザクション選択アルゴリズムに従い，承認する 2 つの他のトランザクションを選択する．

Step3 選択した 2 つのトランザクションを検証し，矛盾（たとえば，二重支払いなど）が含まれていないか確認する．

Step4 PoW と同様に，条件を満たすノンスを見つける計算問題を解く．

Step5 取引内容，2 つの選択したトランザクション，ノンスを含むトランザクションに対して，ユーザの秘密鍵を用いて署名を付与する．

Step6 署名付きトランザクションを送信する．

Step7 送信したトランザクションが直接的/間接的に承認したトランザクションの累積荷重（トランザクションの信頼度）を更新する．

Tangle では，トランザクションを生成するときに他の 2 つのトランザクションを選択して，それらをトランザクション生成者自ら検証して承認する．このため，PoW などの他のコンセンサスアルゴリズムと違い，トランザクション生成者と異なる承認者が存在しない．このため，承認者が担う承認作業の対価としてトランザクション生成時に手数料を支払う必要がなく，無料でトランザクションを生成することができる．また，各トランザクションに対して累積荷重を使った信頼度を定義し，不正に承認されたトランザクションは合意から排除される仕組みが導入されている．この累積荷重は，対象のトランザクションを直接的もしくは他のトランザクションを介して間接的に承認しているトランザクションの荷重の総和であり，累積荷重の値が大きくなると合意形成が行われ取引が成立したと見なされる．

Tangle のトランザクションと累積荷重の例を**図 6** (a) に示す．図中の四角はトランザクションを表し，矢印は根元のトランザクションから先のトランザクションへの承認を表し，四角の中の数字はトランザクションの累積荷重を示す．また，トランザクションの色はトランザクションの生成者を表しており，この例ではユーザ A～D の 4 名がトランザクションを生成している．ただし，簡単のため各トランザクションの荷重は 1 としている．図より，多くの承認を集めたトランザクションに対する累積荷重が大きくなっていることが分かる．この累積荷重の計算過程を**表 1** (a) に示す．この式中の A1～D2 は，各トランザクションの累積荷重を表す．たとえば，トランザクション A5 は，A1，A2，A3，A4 から承認されており，自らの荷重が 1 であることから，A1 + A2 + A3 + A4 + 1 で累積荷重が求めら

© 2019 Information Processing Society of Japan

表 1 各方式における累積荷重の計算過程
Table 1 Calculation procedures of cumulative weights in each method.

(a) Tangle	(b) PoP	(c) PoB
A1 = 1	A1 = 1	A1 = 1
A2 = A1 + 1		
A3 = A1 + A2 + 1	**A3 = A1 + 1**	**A3 = Max(A1, 1)**
A4 = A1 + A2 + A3 + 1		
A5 = A1 + A2 + A3 + A4 + 1	**A5 = A1 + A3 + 1**	**A5 = Max(A1, A3, 1)**
B1 = 1	B1 = 1	B1 = 1
B2 = B1 + C1 + C2 + D1 + 1	B2 = B1 + C1 + C2 + D1 + 1	**B2 = Max(B1, 1) + Max(C1, C2) + D1**
C1 = B1 + 1	C1 = B1 + 1	C1 = B1 + 1
C2 = B1 + C1 + 1	C2 = B1 + C1 + 1	**C2 = B1 + Max(C1, 1)**
D1 = B1 + C1 + C2 + 1	D1 = B1 + C1 + C2 + 1	**D1 = B1 + Max(C1, C2) + 1**
D2 = B1 + C1 + 1	D2 = B1 + C1 + 1	D2 = B1 + C1 + 1

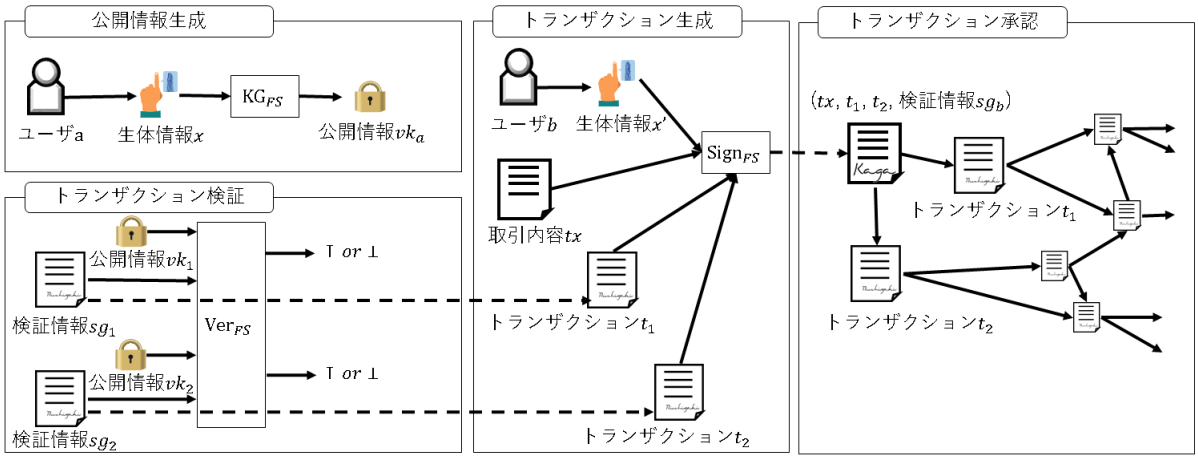


図 7 HTDL の構成
Fig. 7 HTDL construction.

れる．ただし，トランザクション A5 は，ユーザ A が自らのトランザクションに対して多数の承認を行うことによって，その累積荷重を増加させている．トランザクション A5 は他のユーザから承認されておらず，トランザクション生成者本人のみが承認を繰り返しているため，2 重支払いなどの不正トランザクションである可能性がある．このような攻撃は，Step4 のノンスを見つける計算能力を独占することで成功するリスクがある．このため，他のユーザの承認が集まるまでは累積荷重が大きくならないことが望ましい．

本論文ではこのような攻撃に対処するため，DAG と Fuzzy signature を融合させ，生体認証の制約を合意形成に取り入れる．こうすることで，攻撃者は計算能力を独占したとしても，多数のトランザクションを生成することが困難となり，上記のような攻撃が成功するリスクを低減させることができる．

4.3 Fuzzy signature と DAG に基づく分散台帳の構成

本節では，Fuzzy signature と DAG に基づく分散台帳の構成を提案する．提案する構成を図 7 に示す．提案する分散台帳の処理手順は以下のとおりである．

公開情報生成

Step1 ユーザ a から生体情報 x を取得する．

Step2 生体情報 x を鍵生成関数 KG_{FS} へ入力し，検証鍵 vk_{FS} を生成し，HTDL の公開情報 vk_a とする．

Step3 公開情報 vk_a をユーザ a のアドレスとして公開する．

トランザクション検証

Step1 トランザクション選択アルゴリズムに従い，承認する 2 つの他のトランザクション t_1，t_2 を選択する．

Step2 t_1，t_2 に対応する公開情報 vk_1，vk_2 を使って，t_1，t_2 に含まれる検証情報 sg_1，sg_2 の検証を実施する．

Step3 t_1，t_2 に矛盾（たとえば，二重支払いなど）が含まれていないか確認する．

トランザクション生成

Step1 t_1，t_2 に矛盾がなければ，トランザクションの取引内容 tx（仮想通貨であれば，送金先，金額など）を生成する．

© 2019 Information Processing Society of Japan

Step2 ユーザ b から生体情報 x' を取得する．

Step3 生体情報 x'，(tx, t_1, t_2) を署名生成関数 Sign_{FS} へ入力し，署名値 σ_{FS} を生成し，HTDL の検証情報 sg_b とする．

Step4 (tx, t_1, t_2, sg_b) をトランザクションとして生成する．

トランザクション承認

Step1 トランザクション (tx, t_1, t_2, sg_b) をネットワークへ送信し，DAG のノードとして登録する．

Step2 トランザクションが直接的/間接的に承認したトランザクションに対する累積荷重を更新する．

上記の処理手順で HTDL が構成されるが，Fuzzy signature の特性を活用して信頼できるシステムを実現するためには，端末側のソフトウェアおよび生体センサに制約が必要となる．たとえば，攻撃者が他人の指紋画像や顔画像などの生体情報を入手して，端末側のソフトウェアを改ざんして，生体センサ経由でなく端末に保存された生体情報（指紋画像，顔画像など）を入力する攻撃が考えられる．このような場合，生体情報がユーザと不可分な形で紐づいておらず，従来の秘密鍵をトラストアンカとする分散台帳技術のようになりすましのリスクがある状態になるため，HTDL の要件を満たすことはできない．したがって，Fuzzy signature を用いた HTDL の構成では，端末側のソフトウェアは改ざんされておらず，仮に改ざんされた場合はサーバ側で検知してトランザクション生成を防止できることを前提とする．

さらに，漏洩した他人の指紋画像や顔画像などの生体情報に基づき，人工物による偽造生体を作成して，生体センサへ入力する攻撃も考えられる．このような攻撃が発生する場合も同様に HTDL の要件を満たすことができないため，生体センサには生体検知機能が搭載されており，人工物による偽造生体が入力された場合，それを検知してトランザクション生成を防止できることを前提とする．

以上の前提の下で処理に対する制約を詳細化すると，HTDL におけるコンセンサスアルゴリズムは，トランザクション生成/承認時の制約によって Proof of Physical work（PoP），Proof of Biometrics（PoB）の 2 つに分類することができる．

Proof of Physical work（PoP） 4.3 節で提案した HTDL の処理手順に対して，トランザクション生成の Step2，およびトランザクション承認の Step2 を以下のステップで置き換えたアルゴリズムである．

トランザクション生成

Step2′ ユーザ b から生体情報 x' を取得する．ただし，ユーザが生体センサへ生体情報を入力するために一定の時間が必要となる．たとえば，指紋センサで指紋情報を取得するために 1 秒必要であれば，ユーザは最大毎秒 1 回しかトランザクションを生成できない．

トランザクション承認

Step2′ トランザクションが直接的/間接的に承認したトランザクションに対する累積荷重を更新する．ただし，累積荷重の算出方法は，Tangle と同様のアルゴリズムで行う．

上記のような制約を加えることで，生体情報入力には一定の時間が必要になるため，特定のトランザクションに対する累積荷重は計算能力ではなく，承認者数と承認に要した時間に比例する．PoP における累積荷重の例を，図 6 (b) に示す．図より，トランザクション A5 の累積荷重を (a) と比較すると，16 から 4 に減少している．これは，Tangle ではユーザ A が高い頻度で 5 つのトランザクションを生成していたのに対し，PoP では生体情報入力（Physical work）より高い頻度でトランザクションを生成できないため，トランザクション A2，A4 が生成できなくなったことに起因する．また，PoP における累積荷重の計算過程を，表 1 (b) に示す．表より，トランザクション A5 の累積荷重はトランザクション A1，A3 にのみ依存しているため，A2，A4 の分累積荷重が減少している．このような制約を導入することで，高い計算能力を持つユーザ A が独占的に多数のトランザクションを生成できなくなり，他のユーザと同じスループットでトランザクションを生成するようになる．これにより，Tangle と比較して不正トランザクションに関する合計形成が困難となり，信頼性の高い分散台帳が実現されたといえる．

Proof of Biometrics（PoB） 4.3 節で提案した HTDL の処理手順に対して，トランザクション生成の Step2，およびトランザクション承認の Step2 を以下のステップで置き換えたアルゴリズムである．

トランザクション生成

Step2′ ユーザ b から生体情報 x' を取得する．ただし PoP と同様に，単位時間あたりのトランザクション生成数に制約がある．

トランザクション承認

Step2′ トランザクションが直接的/間接的に承認したトランザクションに対する累積荷重を更新する．累積荷重の更新は，Tangle における更新方法に代わり，下記の処理に基づき行う．

Step2-1′ 累積荷重の計算対象のトランザクションを直接的もしくは間接的に承認するトランザクションを収集する（図 6 (c) の B2 を対象とする場合，B1，B2，C1，C2，D1 が収集される）．

Step2-2′ 収集したトランザクションに含まれる公開情報と検証情報をすべての組合せで検証し，トランザクションを同一の承認者からなるクラスタに分割する．この検証では，事後に同一人

© 2019 Information Processing Society of Japan

物の署名を特定する Lazy signature [24] に基づき，公開情報と検証情報の生成者が同一人物か否かを判定する（図 6 (c) の B2 を対象とする場合，{B1,B2}，{C1,C2}{D1} にクラスタリングされる）．

Step2-3′ クラスタごとに累積荷重の最大値を計算し，それをクラスタの累積荷重とする．このクラスタの累積荷重の総和を計算し，対象トランザクションの累積荷重とする（図 6 (c) の B2 を対象とする場合，累積荷重は B2 = Max{B1, 1} + Max{C1, C2} + Max{D1} = B1 + C1 + D1 で求められる）．

上記のような制約を加えることで，1 人のユーザが同一のトランザクションを複数回承認して累積荷重を上昇させる攻撃が防止でき，累積荷重は異なる承認者により生成された承認トランザクションの荷重の総和となる．特定のトランザクションの累積荷重は，計算能力や要した時間に依存せず，承認者の数に比例する．ただし，検証情報を使った同一人物の承認の特定は，DAG が大規模になるのにともなって処理時間が増加するため，承認トランザクションの検証を実用上問題ない範囲に限定するなどの実装上の工夫が必要となる可能性がある．PoB における累積荷重の例を，図 6 (c) に示す．なお，簡単のためユーザが持つ生体情報は 1 人 1 つのみで，毎回必ずその生体情報が入力されると仮定する．図より，トランザクション A5 の累積荷重を (a)，(b) と比較すると，16，4 から 1 に減少している．また，PoB における累積荷重の計算過程を，表 1 (c) に示す．PoB では，A1，A3，A5 が同一人物による承認であるため，この中で累積荷重が最大のトランザクションのみが累積荷重の算出に用いられる．たとえば A5 の累積荷重は，A5 = Max(A1, A3, 1) となる．これにより，同一人物が累積荷重を上昇させることができなくなり，Tangle，PoP よりも承認権限の独占が困難で信頼性の高い分散台帳が実現されたといえる．

Tangle では，無制限に不正トランザクションが発行されることを防止することを目的にノンスを見つける計算を導入していたため，特定のトランザクションの累積荷重は，承認に要した計算量に比例する．PoP，PoB では，Fuzzy signature を導入することにより，生体情報の入力作業や承認者数を独占困難なリソースとして用いる．これにより，既存のコンセンサスアルゴリズムよりも権限の独占が困難となり，信頼性の高いシステムが実現できる．

5. 評価

本章では，既存手法および 4 章で提案した方式が，HTDL の要件を満たすか否かを評価するとともに，HTDL の導入により新たに発生する課題について考察する．

5.1 HTDL の要件に対する評価

分散台帳技術は，トランザクション生成（秘密鍵管理方法）とコンセンサスの 2 つより構成されるため，それぞれの組合せに対して HTDL の要件を評価した．評価結果を**表 2** に示す．

ユーザ鍵管理は，秘密鍵をユーザが管理する端末や紙などで管理するモデルであり，2.1.1 項で示した管理方式のうち，(1-1)〜(1-5) を示す．サーバ鍵管理は，秘密鍵をサーバ上で管理するモデルであり，2.1.1 項で示した管理方式のうち，(2-1)〜(2-3) を示す．トランザクション生成に Fuzzy signature を用いる場合は，秘密鍵を端末やサーバのどこにも保存せず，ユーザの生体情報を動的に取得して秘密鍵として用いる．

また，コンセンサスとしては，代表的なアルゴリズムとして，PoW，PoS，PoA，PoH，Tangle に加え，4.3 節で提案した PoP，PoB を比較対象とする．

R1) トランザクション健全性については，秘密鍵が端末，

表 2 トランザクション生成とコンセンサスの組合せに対する HTDL 要件評価結果
Table 2 HTDL requirement satisfaction.

トランザクション生成	コンセンサス	R1) トランザクション健全性	R2) TTP 不要	R3) 分散権限
ユーザ鍵管理	PoW	×	○	×
	PoS	×	○	×
	PoA	×	○	×
	PoH	×	○	○
	Tangle	×	○	×
サーバ鍵管理	PoW	×	×	×
	PoS	×	×	×
	PoA	×	×	×
	PoH	×	×	○
	Tangle	×	×	×
Fuzzy signature	PoP	○	○	○
	PoB	○	○	◎

© 2019 Information Processing Society of Japan

紙，サーバなどに管理されており，漏洩によるなりすましリスクがある場合は，×とした．秘密鍵がどこにも保存されておらず，端末やサーバ上のすべての情報の漏洩や管理者の内部不正を考慮しても，なりすましが困難である場合は，○とした．R2) TTP 不要については，秘密鍵をユーザ管理する場合や秘密鍵の管理が不要である場合が○であり，秘密鍵をサーバ管理する場合は実質的にサーバが TTP となるため，×と評価した．R3) 独占困難なリソースに基づくコンセンサスについては，計算能力や資産のように投資により容易に権限を独占できるリソースに基づいている場合は×とした．コンセンサスが承認者数と時間に依存している場合は，計算能力や資産よりも承認権限の独占が困難であるとし，○とした．コンセンサスが承認者数のみに依存している場合は，さらに承認権限の独占が困難であるとし，◎とした．

表 2 より，提案手法以外の組合せでは，HTDL の要件をすべて満たすものは存在しない．これに対して，提案した Fuzzy signature と PoP，PoB に基づく方式の要件評価結果を以下に述べる．

R1-1) クレデンシャルの秘匿性 Fuzzy signature に基づくトランザクション生成では，ユーザの生体情報をクレデンシャルとして用いる．クレデンシャルとして虹彩や静脈などの生体情報を用いる場合，取得に特殊な生体センサが必要になるため，他人の生体情報を不正に入手するのは困難であると考えられる．また，指紋や顔のように，物体に遺留したり，通常の写真から入手できたりするものを用いた場合は，トランザクション生成の際に生体情報を直接人間から取得していることを生体検知技術 [25] などを用いて検証することで，なりすましを行うことは困難となる．

R1-2) 検証情報の健全性 Fuzzy signature は，文献 [21], [22] で，EUF-CMA であることが証明されている．このため，攻撃者が検証鍵や署名値から生体情報を推定することや，正規ユーザの生体情報を持たない攻撃者が正規ユーザの署名値を偽造することは困難である．

R2) TTP 不要 Fuzzy signature に基づくトランザクション生成では，トランザクション生成に秘密鍵を用いないため，秘密鍵管理を行う必要がない．また，PoP または PoB に基づきコンセンサスを行うときも TTP を前提とせずにトランザクション生成者が相互に承認しあうため，TTP が不要である．

R3) 独占困難なリソースに基づくコンセンサス Fuzzy signature+PoP では，増加させることができるトランザクションの累積荷重が，承認者数と時間に比例する．従来の Tangle では，各ユーザが同一の計算能力を持っている場合に累積荷重が PoP と等しくなり，少数のユーザや組織が計算能力を独占した場合に不正トランザクションが承認されるリスクが増加する．したがって，PoP において不正トランザクションが承認されるリスクは，従来の Tangle 以下である．

また，Fuzzy signature+PoB では，増加させることができるトランザクションの累積荷重が，承認者数に比例する．従来の Tangle および PoP と比較すると，各ユーザが同一トランザクションを 1 度しか承認できない場合の累積荷重が PoB と等しくなる．したがって，PoB において不正トランザクションが承認されるリスクは，Tangle および PoP 以下である．この議論は，ユーザが 1 つの生体情報しか持てず，同一人物が生成した公開情報と検証情報は確実に検証に成功し，紐づけができるという仮定をおいていた．しかしながら，実際には 1 人のユーザが複数の生体情報を持つこともある．たとえば，指紋であれば，各指から取得できるので，1 人 10 パターンを保持することができる．この場合でも，1 人のユーザは自分のトランザクションを 9 回までしか承認できないため，累積荷重の上限は承認者数に固定の係数が掛け合わされるだけで，独占困難なリソースであることに変わりはない．

以上より，提案する Fuzzy signature+PoP，Fuzzy signature+PoB の組合せだけが HTDL の要件をすべて満たしており，トランザクションから本人性が確認できる信頼性の高い分散台帳を実現しているといえる．

5.2 新たに発生する課題

本節では，前節で議論した観点以外で，HTDL と従来の分散台帳技術を比較し，新たに発生する課題について論じる．

5.2.1 トランザクション検証における本人拒否/他人受入

HTDL では生体認証を活用するため，一定の確率で本人が生成した検証情報が検証に失敗する事象（本人拒否）や，他人が生成した検証情報が第三者の公開情報による検証に成功する事象（他人受入）が発生する．本人拒否は，HTDL では本人が生成したトランザクションが検証に失敗し承認されないことに対応する．これを防止するためには，トランザクション生成時に生成した検証情報が正しく検証できることを確認し，検証できないときには再度生体情報を取得して検証情報を生成し直す必要がある．

生体認証では他人受入率が生体情報の情報量に依存するため，他人の生体情報から生成した検証情報と本人の公開情報との検証に成功してしまう確率を十分小さくできない可能性がある．これに対しては，複数の生体情報を併用して認証精度を高めるマルチモーダル認証 [26] を活用することで，不正なトランザクションが誤って承認されるリスクを低減させることができる．

5.2.2 生体の欠損や死去への対応

HTDL では秘密鍵を管理せず，ユーザの生体情報を秘密情報として検証情報を生成することで，秘密鍵の紛失にと

© 2019 Information Processing Society of Japan

もなって資産が失われるリスクを低減させている．しかしながら，トランザクションを生成するためには生体情報を使って検証情報を生成する必要があるため，事故や死去などにより生体が失われると自身の資産が失われるリスクが発生する．事故などによる生体の欠損に対する対応としては，複数生体の登録が考えられる．複数生体の登録は，生体の欠損に備えて，複数の生体（たとえば，指紋に加えて顔，虹彩など）を登録することで可用性の低下を防ぐ対策である．複数の生体から検証鍵を生成してそれらを HTDL の公開情報とすることで，複数の生体のうちどれか 1 つを入力することでトランザクション生成ができるシステムが実現できる．

また，死去への対応については，本人が自らの生体情報を入力することで取引の意思表示を行うことが困難となるため，HTDL としては通常の取引は行えなくなるのが正しい対処だと考える．ただし，遺族が遺産を受け取ることは必要となるため，あらかじめ死去したときに相続人へ資産を委譲するスマートコントラクトを生成しておくなど，HTDL のような分散台帳技術よりも上位のレイヤで救済策を用意する必要がある．

5.2.3 トランザクションのデータ量

Fuzzy signature の構成要素として用いる電子署名の公開鍵サイズを $s(pk)$，電子署名の署名サイズを $s(\sigma)$，生体情報のサイズを $s(c)$ とすると，Fuzzy signature に基づき生成される公開情報と検証情報のサイズはそれぞれ，$s(pk)+s(\sigma)+s(c)$ および $s(\sigma)+s(c)$ である [21]．既存の分散台帳における公開情報と検証情報のサイズは，それぞれ $s(pk)$，$s(\sigma)$ であるため，トランザクションのサイズは，$s(\sigma)+2s(c)$ だけ増加する．

たとえば典型的な例として，電子署名サイズを 1,024 bit，生体情報のサイズを 512 byte [27] に設定すると，トランザクションサイズは 1,152 byte 増加する．Tangle を採用している仮想通貨 IOTA のトランザクションサイズは 1,650 byte [28] であり，HTDL の導入で 1.7 倍程度に増加する計算である．各ノードが保持するデータ量が増加するものの，十分実用に耐える範囲であると考える．

5.2.4 各処理の計算時間

公開情報生成，トランザクション検証，トランザクション生成では，それぞれ Fuzzy signature の検証鍵生成，署名検証，署名生成を行う．これらの処理は，従来の電子署名に加えて生体情報に対する誤り訂正処理を行う必要があるため，計算時間が増加する．ただし，この計算時間は鍵長に対して線形であり，短時間で処理できるため実用上問題ないといえる．

トランザクション承認に関しては，PoP では従来の Tangle と変わらない計算時間であるものの，PoB では計算時間が大きく増加する可能性がある．PoP では，単に対象のトランザクションを直接的もしくは間接的に承認しているトランザクションの荷重の総和を算出することで累積荷重を算出する．しかしながら，PoB では対象のトランザクションを直接的もしくは間接的に承認しているトランザクションの数を N としたときに，${}_NC_2$ 回の検証情報の検証が必要となる．この処理は，N^2 に比例する計算時間を要するため，HTDL のネットワークが肥大化すると実用的な時間で終わらない可能性がある．このため，計算対象の承認トランザクション数に上限を設けて，上限を超える分に関しては累積荷重の計算対象から除外するなどの対策を講じる必要がある．ただし，上限を小さく設定すると十分に累積荷重が大きくならずトランザクションが承認されないという問題が発生する可能性がある．この計算時間とシステムの安定性はトレードオフの関係にあり，適切なパラメータ設定は今後の課題である．

5.2.5 ユースケースの制約

提案した HTDL の構成法は，IoT デバイスをターゲットにした仮想通貨 IOTA に用いられている Tangle をベースにしているが，IoT デバイスを対象としていない．HTDL のユースケースとしては，ユーザのスマートフォンや PC から，銀行口座のように高い安全性を持つ仮想通貨の取引，スマートコントラクトを使った仲介者なしの契約処理が考えられる．このようなユースケースでは，資産の盗難や紛失のリスクを抑え，安全性の高い基盤を実現する必要があるため，HTDL の要件を満たすことが必要となる．

5.2.6 端末の制約

既存の分散台帳技術になかった制約として，取引時に生体情報を取得する生体センサ（指紋センサなど）が必要になる．この生体センサは，近年スマートフォンへの搭載が進んでおり 2019 年には 100%のスマートフォンに生体センサが搭載されると予測されている [29]．よって，生体センサの利用は無理な制約ではなく，提案手法は十分広範に利用することができるソリューションであると考えられる．

一方で生体認証を使って台帳管理をするという特徴は，ユーザの利便性を向上させる．従来の秘密鍵を中央集権的に管理する分散台帳において新たなトランザクションを生成するためには，秘密鍵を格納した端末，ハードウェアウォレット，紙に印刷したペーパーウォレットなどを用いる必要があった．このため，ウォレットが格納された端末のみでトランザクションが生成できるという制約や，ハードウェアウォレットやペーパーウォレットを持参しないとトランザクションを生成できないという可用性の課題があった．これに対して提案手法は，秘密鍵を管理する必要がないため，特定の端末や所有物に依存せず，ユーザが生体情報を入力するだけでいつでもトランザクションを生成できる．たとえば仮想通貨に提案手法を応用した場合，仮想通貨の支払い時にユーザが生体情報を入力するだけで，自分の仮想通貨から支払いができる．この特徴は，既存の分散台帳技術にはないものであり，提案手法は安全性と同

© 2019 Information Processing Society of Japan

時に高い利便性も達成することができる．

6. まとめ

本論文では，既存の分散台帳の課題を分析した後，その課題を解決する Human Trusted Distributed Ledger（HTDL）のコンセプトを提案し，その要件を整理した．さらに，Fuzzy signature と有向非循環グラフ（DAG）を用いて，生体情報を秘密鍵として用いてトランザクション生成から承認までを一括して行う方式を提案し，HTDL の要件を満たすことを確認した．

今後の課題としては，HTDL の具体的な構成に対する理論的分析や，実際に実装した HTDL の性能や攻撃に対する耐性の分析があげられる．

参考文献

[1] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008), available from ⟨https://bitcoin.org/bitcoin.pdf⟩ (accessed 2018-10-12).

[2] Wood, G.: Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper 151 (2014), available from ⟨http://gavwood.com/paper.pdf⟩ (accessed 2018-10-12).

[3] Zheng, Z. et al.: Blockchain challenges and opportunities: A survey, Work Pap. 2016 (2016), available from ⟨https://www.henrylab.net/pubs/ijwgs_blockchain_survey.pdf⟩ (accessed 2018-10-12).

[4] Roberts, J.J. and Rapp, N.: Exclusive: Nearly 4 Million Bitcoins Lost Forever, New Study Says (2017), available from ⟨http://fortune.com/2017/11/25/lost-bitcoins/⟩ (accessed 2018-10-12).

[5] Atzei, N., Massimo, B. and Tiziana, C.: A survey of attacks on Ethereum smart contracts (SoK), *International Conference on Principles of Security and Trust*, Springer, Berlin, Heidelberg (2017).

[6] Decker, C. and Roger, W.: Bitcoin transaction malleability and MtGox, *European Symposium on Research in Computer Security*, Springer, Cham (2014).

[7] 一橋ビジネスレビュー：次世代産業としての航空機産業，2018 年 SPR，Vol.65, No.4 (2018).

[8] Conti, M. et al.: A survey on security and privacy issues of bitcoin, *IEEE Communications Surveys & Tutorials* (2018).

[9] Eskandari, S. et al.: A first look at the usability of bitcoin key management, arXiv preprint arXiv:1802.04351 (2018).

[10] Yan, J. et al.: Password memorability and security: Empirical results, *IEEE Security & Privacy*, Vol.2, No.5, pp.5–31 (2004).

[11] Josefsson, S.: PKCS# 5: Password-based key derivation function 2 (PBKDF2) test vectors, No.RFC 6070 (2011).

[12] 志波和幸：仮想通貨の拡大に悩む規制当局，東洋経済，2017 年 11/04 号 (2017).

[13] Magaki, I. et al.: ASIC clouds: Specializing the datacenter, *ACM SIGARCH Computer Architecture News*, Vol.44, No.3, IEEE Press (2016).

[14] King, S. and Scott, N.: Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, self-published paper (2012), available from ⟨https://peercoin.net/assets/paper/peercoin-paper.pdf⟩ (accessed 2018-10-12).

[15] Bentov, I. et al.: Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake, *ACM SIGMETRICS Performance Evaluation Review*, Vol.42, No.3, pp.34–37 (2014).

[16] Technical Reference NEM Version 1.0, Technical Report (2015), available from ⟨https://nem.io/NEM_techRef.pdf⟩ (accessed 2018-10-12).

[17] Li, X. et al.: *A survey on the security of blockchain systems*, Future Generation Computer Systems (2017).

[18] Jeremiah, B. and Zhou, H.-S.: *Designing proof of human-work puzzles for cryptocurrency and beyond*, Theory of Cryptography Conference, Springer, Berlin, Heidelberg (2016).

[19] Popov, S.: The Tangle (2016), available from ⟨https://iota.org/IOTA Whitepaper.pdf⟩ (accessed 2018-10-12).

[20] Dodis, Y., Leonid, R. and Adam, S.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg (2004).

[21] Takahashi, K. et al.: A signature scheme with a fuzzy private key, *International Conference on Applied Cryptography and Network Security*, Springer, Cham (2015).

[22] Matsuda, T. et al.: Fuzzy signatures: Relaxing requirements and a new construction, *International Conference on Applied Cryptography and Network Security*, Springer, Cham (2016).

[23] Fast IDentity Online (FIDO) (2015), available from ⟨https://fidoalliance.org/⟩ (accessed 2018-10-12).

[24] 米山裕太，高橋健太，本部栄成，西垣正勝：生体情報を用いた認印型デジタル署名：Lazy Signature，情報処理学会研究報告，2012-CSEC-58-43, pp.1–7 (2012).

[25] Sousedik, C. and Christoph, B.: Presentation attack detection methods for fingerprint recognition systems: A survey, *Iet Biometrics*, Vol.3, No.4, pp.219–233 (2014).

[26] Murakami, T., Ohki, T. and Takahashi, K.: Optimal sequential fusion for multibiometric cryptosystems, *Information Fusion*, Vol.32, pp.93–108 (2016).

[27] Schroff, F., Dmitry, K. and James, P.: Facenet: A unified embedding for face recognition and clustering, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2015).

[28] IOTA Foundation, available from ⟨https://github.com/iotaledger⟩ (accessed 2018-10-12).

[29] Acuity, The Global Biometrics and Mobility Report: The Convergence of Commerce and Privacy, Market Analysis and Forecasts 2016 to 2022 (2017).

加賀 陽介

2006 年北海道大学工学部情報工学科卒業．2008 年同大学大学院修士課程修了．同年（株）日立製作所入社．現在，同社研究開発グループ研究員．生体認証および情報セキュリティの研究開発に従事．電子情報通信学会会員．

© 2019 Information Processing Society of Japan

藤尾 正和 （正会員）

日立製作所研究開発グループシステムイノベーションセンタセキュリティ研究部所属．現在，生体認証に関する研究開発に従事．博士（理学），人工知能学会，電子情報通信学会各会員．

長沼 健

2007 年株式会社日立製作所研究開発グループ入社，モバイルアプリケーション，GPS 利活用，車載無線，医療データ分析，ブロックチェーン，およびこれらに関係するセキュリティ技術，暗号技術の開発に従事，2017 年東京大学大学院新領域創成科学研究科博士課程入学，現在に至る．

高橋 健太 （正会員）

1998 年東京大学理学部情報科学科卒業．2000 年同大学大学院理学系研究科情報科学専攻修士課程修了．同年（株）日立製作所入社．2012 年東京大学大学院情報理工学系研究科博士後期課程修了，博士（情報理工学）．2015 年より東京大学大学院客員准教授．現在，（株）日立製作所研究開発グループユニットリーダ主任研究員．生体認証，暗号技術および情報セキュリティの研究開発に従事．2008 年情報処理学会論文賞，2011 年 SISAP Best paper，2012 年 IEEE BTAS Best reviewed paper，2015 年情報処理学会長尾真記念特別賞，2016 年ドコモ．モバイル・サイエンス賞先端技術部門優秀賞，関東地方発明表彰発明奨励賞等受賞．電子情報通信学会会員．

村上 隆夫 （正会員）

2004 年東京大学工学部電子情報工学科卒業．2006 年同大学大学院情報理工学系研究科修士課程修了．2014 年同大学院情報理工学系研究科博士課程修了．博士（情報理工学）．2006 年（株）日立製作所入社．2014 年産業技術総合研究所入所．生体認証，プライバシ保護の研究開発に従事．2011 年平成 22 年度山下記念研究賞受賞．2015 年 IEEE TrustCom Best Paper Award 受賞．電子情報処理学会，IEEE 各会員．

大木 哲史 （正会員）

2002 年早稲田大学理工学部電子情報通信学科卒業．2004 年同大学大学院理工学研究科電子・情報通信学専攻修士課程修了．2010 年早稲田大学理工学術院情報・ネットワーク専攻博士（工学）取得．2010 年早稲田大学理工学総合研究所次席研究員，2013 年産業技術総合研究所特別研究員を経て，2017 年より静岡大学大学院総合科学技術研究科講師．情報セキュリティ全般，特に個人認証を中心としたネットワークセキュリティに関する研究に従事．電子情報通信学会会員．

西垣 正勝 （正会員）

1990 年静岡大学工学部光電機械工学科卒業．1995 年同大学大学院博士課程修了．日本学術振興会特別研究員（PD）を経て，1996 年静岡大学・情報助手．同講師，助教授の後，2010 年より同大学創造科学技術大学院教授．博士（工学）．情報セキュリティ全般，特にヒューマニクスセキュリティ，メディアセキュリティ，ネットワークセキュリティ等に関する研究に従事．2013～2014 年情報処理学会コンピュータセキュリティ研究会主査．2015～2016 年電子情報通信学会バイオメトリクス研究専門委員会委員長．2016 年より日本セキュリティマネジメント学会常任理事．本会フェロー．

© 2019 Information Processing Society of Japan