プライバシーを考慮した ブロックチェーンの取引者間事前合意プロトコル

安坂 祐紀[1,a)]　渡辺 知恵美[2,b)]　天笠 俊之[3,c)]　北川 博之[3,d)]

概要：ブロックチェーンは，複数の参加者間で中央サーバを持たずに取引データを一貫して共有する技術である．資産売買や貿易などのアプリケーション実装のため，許可された参加者のみで構成されるプライベート型の開発が進んでいる．一方，取引額や所持金を秘匿したいといったプライバシーの保護要求がある．プライバシーを考慮すると，取引者以外の参加者からは取引の正当性のみの確認しかできないため，意図しない取引が承認されてしまう場合がある．本稿では，プライベートブロックチェーンにおいて，取引者間で事前に合意を取った上で，取引額及び所持金を秘匿するプロトコルを提案する．秘匿したい金額を加法準同型暗号で暗号化したまま計算し，取引額及び支払い可能な所持金を持つことをゼロ知識証明することで，信頼できる第三者を必要とせずに取引の正当性を検証可能であることを示した．

キーワード：ブロックチェーン，プライバシー保護，電子商取引

A Privacy-preserving Pre-Consensus Protocol for Blockchain

Yuki Yasusaka[1,a)]　Chiemi Watanabe[2,b)]　Toshiyuki Amagasa[3,c)]　Hiroyuki Kitagawa[3,d)]

Abstract: Blockchain is a technology that allows us to process transactions and share committed data among the participants without having a central server. In order to implement applications such as asset sales and trade, a special type of blockchain systems called private blockchain, where permitted parties are allowed to participate, has been developed. Meanwhile, there are growing needs of privacy-preservation that sensitive information, such as amount of trade and balances, is not disclosed. However, this prevents the parties that are not involved in the transaction from checking the correctness, leading to commit unintended transactions. To this problem, in this paper, we propose a protocol that enables checking the consistency regarding the transaction amount and the balance without disclosing their values. Specifically, we exploit a homomorphic encryption to encrypt transaction amount and balances. Thus, we show that the correctness of a transaction is publicly verifiable by the parties using zero knowledge proof without having a trusted third party.

Keywords: Blockchain, Privacy Preservation, Transaction Systems

[1] 筑波大学大学院 システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
[2] 筑波大学 図書館情報メディア系
Faculty of Library, Information and Media Science, University of Tsukuba
[3] 筑波大学 計算科学研究センター
Center for Computational Sciences, University of Tsukuba
[a)] yuki@kde.cs.tsukuba.ac.jp
[b)] chiemi@slis.tsukuba.ac.jp
[c)] amagasa@cs.tsukuba.ac.jp
[d)] kitagawa@cs.tsukuba.ac.jp

1. はじめに

近年，Bitcoin [1] をはじめとする仮想通貨並びに Bitcoin を実現する分散台帳共有技術であるブロックチェーンが注目を集めている．Bitcoin のような不特定多数の Peer が参加することができるパブリックなネットワーク以外にも，許可された Peer のみが参加可能なプライベートなネットワークを構築する，よりエンタープライズ向けのプライベートブロックチェーンも開発が進んでいる．また，仮想

© 2018 Information Processing Society of Japan

通貨以外にも資産管理や金融・証券取引，更には Internet of Things のシステムにおけるユースケースの検討とその実証実験も多く行われている．

ブロックチェーンでは，送金や資産移動などの記録をトランザクションとして台帳に書き込まれる．全参加者が中央サーバを持たない完全な Peer-to-Peer のネットワークに属し，全員が台帳のコピーを持つことで一貫した情報の記録を全ての参加 Peer で共有する．不正な記録がないか第三者（この場合は，取引に参加していない別の Peer）が監視することができる．またブロックチェーンはアドレスが口座番号の役目を果たしており，そのアドレスが誰のものかという個人情報には結びつかないため匿名性を保証する．

しかしながら，全てのトランザクションは全参加 Peer に公開されるため，資産管理や金融などでの使用を検討する際にはトランザクション内容の秘匿化をする必要がある．例えば不動産の売買において，原則取引価格は非公開である．その際に取引額を明かさなければならない一方で，アプリケーションによっては取引額は第三者の秘匿が必要な場合や，取引額の推定を防止するための各参加 Peer の所持金秘匿の必要も考えられる．

Sasson ら [3] や Kosba ら [5] はトランザクション内容を秘匿し，ゼロ知識証明によって取引に関係ない第三者でも検証可能なシステムを提案している．しかしこれらのシステムでは，トランザクション内容が秘匿されているために取引者間で事前に取り決められていた取引内容と異なる内容のトランザクションが発行されるケースがある．そのためシステム内で事前合意をするプロセスを設ける必要がある．このような事前合意がプロセスに組み込まれていないことにより生じる問題の例として CoinCheck の NEM[2] 流出事件が挙げられる．この事件では約 580 億円分の NEM がダークウェブサイトを通じてほぼ全て別の仮想通貨へと換金された．盗難した犯人は不特定多数に対して NEM を送金することで追跡を困難にした．トランザクション発行プロセスに取引者間の事前合意が組み込まれることによりこのような抜け道を防ぐことができる．

本稿では，プライベートブロックチェーンにおいて，取引者間で事前に合意を取った上で，取引額及び所持金を秘匿するプロトコルを提案する．提案プロトコルではプライベートブロックチェーンのトランザクション発行プロセスを事前合意プロセスと通常のトランザクション発行プロセスの 2 フェーズで構成する．事前合意のプロセスでは，お互いの所持金を秘匿しつつ取引者間のみで取引額について合意を行い，第三者も検証可能な署名を付す．その後，通常のトランザクション発行プロセスで，各参加 Peer の所持金及び取引額を秘匿しつつトランザクションの正当性を検証し，トランザクションを発行する．なお，秘匿したい金額を加法準同型暗号で暗号化したまま計算し，取引額が不正な値（負の値）でないこと及び送金者が支払い可能な所持金を持っていることをゼロ知識証明することで，信頼できる第三者を必要とせずに取引の正当性を検証可能にする．

本稿の構成は以下のようになっている．まず，2 節で関連研究について述べ，3 節で本稿で使用する Hyperledger Fabric について述べる．次に，4 節で本稿の問題設定について述べ，5 節で提案手法について述べる．最後に，6 節にてまとめと今後の課題について述べる．

2. 関連研究

代表的なブロックチェーン基盤に Bitcoin[1], Ethereum[10], Hyperledger Fabric[9] がある．[1], [10] は不特定多数が参加可能なパブリックブロックチェーン基盤であるのに対し，[9] は許可された参加者（Peer）のみ参加可能なプライベートブロックチェーン基盤である．[9] にはプライベートデータを扱うための multichannel という仕組みがあり，ネットワークを分けることによりプライバシーの保護を実現している．しかし，ネットワークに属している他の Peer からはトランザクション内容が見えてしまう．

Sasson ら [3] は，仮想通貨 Bitcoin をベースに非対話型ゼロ知識証明の技術である zk-SNARKs を用いて，匿名仮想通貨 zerocash を提案した．現在は Zcash[4] として実用化されている．また仮想通貨以外のアプリケーションにおけるトランザクションデータの秘匿に関する研究は Kosba ら [5] を筆頭に数多く取り組まれている．Kosba らはトランザクション内のプライベートデータを semi-honest[17] な第三者 Peer である Minimally trusted manager に検証させ，非対話型ゼロ知識証明によりその正当性を示すことでトランザクションの秘匿を実現する Hawk を提案した．また Ceccetti ら [6] は，より現実の金融システムに適応するため，銀行間中継台帳として銀行に属しているユーザ間のトランザクションを完全に秘匿するフレームワークである Solidus を提案した．シンガポール中央銀行 (MAS)[7] は匿名性のあるリアルタイムグロス決済システムを開発するプロジェクトを進めている．取引を行う二者は個別にプライベートネットワークを作り，その全てのプライベートネットワークに MAS が入ることで，決済のネッティングを行う．これらの手法は実用的な構成になっているが，それぞれ信頼できる第三者 Peer（Trusted Third Party）として Minimally trusted manager 及び fully trusted な銀行間を前提としている．また，Zyskind ら [8] は off-chain でセキュアマルチパーティ計算を使用することでデータのプライバシー保護を実現する Enigma を開発した．このシステムでは off-chain で秘密計算を行なっているが，プライベートネットワークでの使用を考えると関係者以外に要配慮情報が渡る事は望ましくない．

表 1 に本稿の提案手法と関連研究との比較を示す．zerocash[3] も信頼できる第三者 Peer を必要とせずに送金を

© 2018 Information Processing Society of Japan

表 1　関連研究との比較

	第三者 Peer	事前合意
提案手法	無	有
zerocash[3]	無	無
Hawk[5]	有	無
Solidus[6]	有	無
Ubin[7]	有	無
Enigma[8]	無 (ただし off-chain)	無

可能にしているが，事前合意プロセスがないので悪用されてしまう可能性がある．

3. Hyperledger Fabric

本稿では，プライベートブロックチェーン基盤 Hyperledger Fabric ver 1.1 [9] を用いる．Hyperledger Fabric は数多くのプロジェクトに採用されており，Go，Java，JavaScript の言語にも対応していることから拡張性があると考えられる．本節では，プライベートブロックチェーン基盤 Hyperledger Fabric ver 1.1 の概要について紹介する．

3.1　トランザクションのデータ構造

Hyperledger Fabric では，チューリング完全なプログラミング言語によって記述された，チェーンコード（スマートコントラスト）によりトランザクションが発行される．本稿では送金トランザクションを対象としているため，必ず取引額の入力値と出力値が同じとなるような UTXO(Unspent Transaction Output) ベースのデータ構造のトランザクションを発行することを想定する．

3.2　ブロックチェーンの構成要素

ブロックチェーンの構成要素は以下の通りである．

ユーザ

ユーザ u_i はクライアントアプリケーションを通じて，ネットワークに参加する Peer P_i に接続する．なお，ユーザとアカウントは 1 : 1 対応であると仮定する．

Peer

Hyperledger Fabirc のネットワークは次の二種類の Peer から構成される．

(1) トランザクションを発行可能な Peer: P_i

ユーザ u_i（各個人・機関）の保有するストレージから構成される．

(2) ブロックを生成する特殊な Peer: Orderer

トランザクション

Peer P_i はトランザクションを発行する上で，最初にトランザクションプロポーザルを全体に送り，承認された場合には実際にブロックに書き込まれるトランザクションを発行する．

(1) トランザクションプロポーザル

ユニークな ID を T_j，送金者を $u_{s,j}$，受取者を $u_{r,j}$，取引額を v_j として，tran-proposal:=(T_j, $u_{s,j}$, $u_{r,j}$, v_j) とする．

(2) トランザクション

参加ユーザ u_i による署名を Sig_i，その集合を S とすると，トランザクション transaction:=(T_j, $u_{s,j}$, $u_{r,j}$, v_j, S) とする．

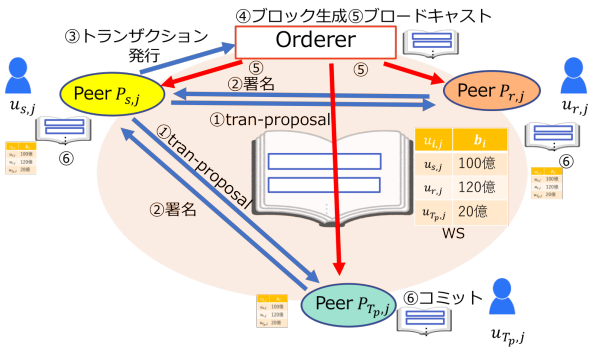

図 1　通常のトランザクションの流れ

ブロック

複数のトランザクションは Orderer によってブロックという単位でまとめられる．ブロックはその直前に発行されたブロックのハッシュ値を持つ，チェーン構造になっている．そのため，仮にブロック内のトランザクションの一部が改竄されても検知することができる．

ワールドステート:WS

全ユーザ u_i の取引後の最新の所持金を保存するステート DB．全 Peer で同じ内容のコピーを持つ．データベースには，$\langle u_i, b_i \rangle$ のように格納される．

3.3　トランザクション処理の流れ

Hyperledger Fabric における送金トランザクションの一例の流れについて図 1 に示す．ここで，トランザクションプロポーザル tran-proposal は (T_j, $u_{s,j}$, $u_{r,j}$, v_j) を表しており，(T_1, A 社, B 社, 10 億円) の様なトランザクションプロポーザルが発行される．なお，この tran-proposal を発行するのは取引者のいずれかであり，通常は送金者が発行するため，今回は $u_{s,j}$ を発行者とする．まず，$u_{s,j}$ はネットワーク内の他 Peer にトランザクションプロポーザル tran-proposal を送る (図 1 (1))．tran-proposal を受け取った各 Peer はそれぞれトランザクションの検証を行う．ここでの検証事項としては，送金者 $u_{s,j}$ は取引額の支払いが可能であるかどうか，つまり取引額以上の所持金を保有しているかどうかであり，WS を確認することにより検証することができる．検証できた場合には署名 Sig_i を付して tran-proposal の送信者に返す (図 1 (2))．署名が事前に定められた一定数 (例:ネットワークの 2/3) 集められ，エンドースメントポリシーに規定された合意を取った後に，トランザクション transaction(T_j, $u_{s,j}$, $u_{r,j}$, v_j, S) を発行する (Orderer に送る)(図 1 (3))．次に，Orderer は規定の

© 2018 Information Processing Society of Japan

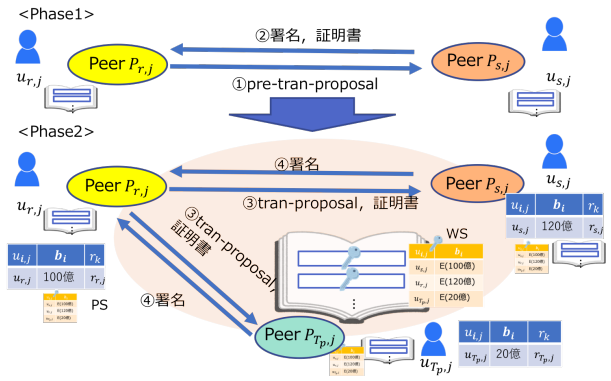

図 2 提案プロトコル

署名 (合意) が集められていることを確認した後，ブロックを生成し (図 1 (4))，ネットワーク全体にブロードキャストする (図 1 (5))．ブロックを受け取った Peer はそれぞれ検証を行い，自身の持つ台帳にブロックを追加，及び取引者 (この場合は，$u_{s,j}$ 及び $u_{r,j}$) の WS を更新する (図 1 (6))．

4. 問題設定

Hyperledger Fabric[9] の従来のトランザクション発行プロトコルや，[3] や [5] のようなトランザクションを秘匿する関連研究のシステムには事前合意が含まれていない．しかし，前述した通り，トランザクション内容が秘匿されているために取引者間で事前に取り決められていた取引内容と異なる内容のトランザクションが発行されたとしても，それを検出することができない．そのためシステム内で事前合意をするプロセスを設ける必要がある．本研究で扱う問題をまとめると以下の通りである．

- 取引者がお互いの所持金を秘匿したまま，任意の送金トランザクションについて，そのトランザクションが成立しうるか（送金額が残高を越えない，あるいは，送金額が負ではないこと）を検証可能である．
- 取引者が事前合意したことを第三者 Peer は検証可能である．
- 事前合意が成立した送金トランザクションについて，取引に関係しない第三者 Peer はトランザクションの内容を知ることなく，それが正当な送金トランザクションであることを検証可能である．

5. 提案手法

図 2 に提案プロトコルの概要図を示す．本プロトコルは，Phase1 で 5.2.2 節に述べる取引者間の事前合意を行い，Phase2 で 5.3 節に述べる取引額 v_j を秘匿したネットワーク全体へのトランザクションプロポーザル及び，5.4 で述べるブロック生成を行う 2 フェーズの設計となっている．

5.1 標準プロトコルからの変更点

標準プロトコルからの変更点を以下に示す．

Peer

全参加 Peer P_i は共通の楕円 ElGamal 暗号の公開鍵 pk を保持する．ここで，鍵生成は第三者機関である公開鍵基盤 (PKI: Public Key Infrastructure) が行うこととし，秘密鍵は参加 Peer は持つことはできない．

プライベートステート

Peer P_i のローカルステート DB をプライベートステート PS_i として，以下を保存する．

(1) 所持金 b_i（平文）

(2) b_i の暗号化に使用した直前の乱数 $r_{k,i}$

データベースには，$\langle u_i, b_i, r_{k,i}\rangle$ のように格納される．r_k に関しては取引を行う度に，更新されていく．例えば u_i が送金する際には，加法準同型性を用いて $Enc_{pk}(b_i - v_j)$ を計算する．ここで，$Enc_{pk}(b_i)$ と $Enc_{pk}(v_j)$ はそれぞれ，乱数 $r_{k,i}, r_j$ を用いて暗号化された値である．そのため，計算後の最新の u_i の所持金は，A.1 で紹介する楕円 ElGamal 暗号の性質より，乱数 $r_{k,i} - r_j$ を用いて暗号化された値となる．同様に u_i が受取する際には，同様に加法準同型性を用いて $Enc_{pk}(b_i + v_j)$ の計算を行い，この暗号文に使用された乱数は $r_{k,i} + r_j$ へと更新される．本稿ではこの楕円 ElGamal 暗号の特性を利用して，乱数を取引の度に更新されていく秘密鍵のように使う．

ワールドステート

全ユーザ u_i の暗号化された最新の所持金を保存する．データベースには，$\langle u_i, Enc_{pk}(b_i)\rangle$ のように格納される．

トランザクション

取引額 v_j を暗号化に使用する乱数を r_j，取引額 v_j が 0 以上であることを示すゼロ知識証明の署名を σ_v，取引後の $u_{s,j}$ の所持金 $b_s - v \geq 0$ のゼロ知識証明の署名を σ_{b_s} として，

$(T_j, u_{s,j}, u_{r,j}, Enc_{pk}(v_j), Sig_s(Enc_{pk}(v_j)), \sigma_v, \sigma_{b_s}, S)$ と表される．

5.2 取引者 ($u_{s,j}, u_{r,j}$) 間の事前合意

事前合意における検証事項と流れについて述べる．

5.2.1 検証事項

$u_{r,j}$ は下記（1），（2）を検証し，次のフェーズであるネットワーク全体へのトランザクションプロポーザルの発行をもって $u_{r,j}$ の合意とする．

(1) $u_{s,j}$ の取引後の所持金 $b_s - v \geq 0$

(2) （1）の証明が本当に $b_s - v$ の検証であること

一方，$u_{s,j}$ は $u_{r,j}$ から送られてきた事前トランザクションプロポーザルの取引額 v_j を確認し，暗号化した取引額

© 2018 Information Processing Society of Japan

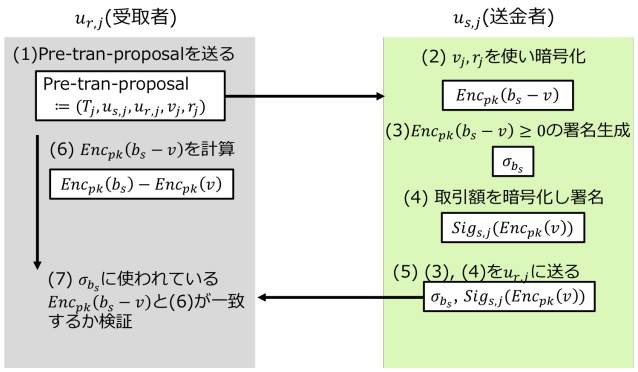

図 3 事前合意のプロセス

$Enc_{pk}(v_j)$ に署名を付すことで合意とする．

5.2.2 事前合意の流れ

図 3 に取引者（$u_{s,j}, u_{r,j}$）間の事前合意のプロセスを示す．受取者の $u_{r,j}$ が事前トランザクションプロポーザル Pre-tran-proposal（T_j, $u_{s,j}$, $u_{r,j}$, v_j, r_j）を送金者 $u_{s,j}$ に送る（図 3 (1)）．$u_{s,j}$ は Pre-tran-proposal を受け取ったら，$PS_{s,j}$ から直前に自身の所持金を暗号化するのに利用した乱数 $r_{k,i}$ を参照し，乱数 $r_{k,i} - r_j$ で取引後の所持金の平文である $b_s - v_j$ を暗号化する（図 3 (2)）．次に，取引後の $u_{s,j}$ の所持金 $b_s - v$ が 0 以上であることを示す Range proof σ_{b_s} を生成する．ここで，$\sigma_{b_s} := (Enc_{pk}(b_s - v_j), Ds(s = 0, 1, ..., 9), PK_{st}(s = 0, ..., 9, t = 0, 1, ..., 9), \sigma_s)$ 及び，$\sigma_s := \{e_0, s_0, s_1, ..., s_n\}$（図 3 (3)）．更に，取引額を暗号化し署名を付す（$Sig_{s,j}(Enc_{pk}(v_j))$）．これは後に $u_{r,j}$ が全体にトランザクションプロポーザルを送る際に，$u_{s,j}$ が取引額 v_j に合意したことを示すために行う（図 3 (4)）．$u_{s,j}$ は (3) 及び (4) で生成した（σ_{b_s}, $Sig_{s,j}(Enc_{pk}(v_j))$）を $u_{r,j}$ に送る（図 3 (5)）．$u_{r,j}$ は事前トランザクションプロポーザルを送った後に WS から $Enc_{pk}(b_s)$ を参照し，$Enc_{pk}(b_s) - Enc_{pk}(v_j) = Enc_{pk}(b_s - v_j)$ を計算する（図 3 (6)）．(5) で受け取った σ_{b_s} 内の $Enc_{pk}(b_s - v_j)$ と (6) で計算した $Enc_{pk}(b_s - v_j)$ が一致することを検証する．ここで，一致しなかった場合 σ_{b_s} は正しい署名でないため abort される（図 3 (7)）．

5.3 取引額 v_j を秘匿した，ネットワーク全体へのトランザクションプロポーザル

上記取引者 $(u_{s,j}, u_{r,j})$ 間の事前合意が取れたトランザクションプロポーザルを $u_{r,j}$ が全体に対して送る．σ_v を取引額 v_j の Range proof として，トランザクションプロポーザルは，

tran-proposal:=

$(T_j, u_{s,j}, u_{r,j}, Enc_{pk}(v_j), Sig_{s,j}(Enc_{pk}(v_j)), \sigma_v, \sigma_{b_s})$ となる．この tran-proposal を受け取った全ユーザ u_i は，それぞれ tran-proposal の内容を検証し，署名 Sig_i を付して $u_{r,j}$ に送り返す．なお，σ_v, σ_{b_s} の検証方法については 5.3.1 節に記す．$u_{r,j}$ は受け取った署名の集合 S がエンドースメントポリシーを満たしていることを確認し，transaction として (tran-proposal, S) を Orderer へ送る．

5.3.1 σ_{b_s} 及び σ_{v_j} の検証

取引に関わっていない第三者 $u_{T_p,j}$ は暗号化された $u_{s,j}$ の取引後の所持金が 0 以上であることを示す $\sigma_{b_s} := (Enc_{pk}(b_s - v_j), Ds(s = 0, 1, ..., 9), \sigma_s)$ の検証を行う．まず，検証者 $u_{T_p,j}$ は

$$\sum_{s=0}^{9} Ds = Enc_{pk}(b_s - v_j)$$

となることを検証する．一致しなかった際には，トランザクションプロポーザルは abort される．次に，$Ds(s = 0, 1, ..., 9)$ から n 個のリング署名の公開鍵ペアを生成する．公開鍵ペアは証明者が生成したものを検証者に送るのではなく，検証者が自ら計算することで証明者の不正を防ぐことができる．同様に σ_{v_j} も計算できる．

5.4 ブロック生成

Orderer は $u_{r,j}$ から transaction を受け取り，エンドースメントポリシーを満たしていることを検証する．検証できた際には，トランザクションの順序付けを行い，ブロックを生成する．ブロックは全体へとブロードキャストされ，各 Peer P_i は受け取ったブロックを検証し，自分の持つブロックチェーンのコピーに追加する．また，併せてトランザクションを反映させた WS も更新する．更に取引の当事者は，PS_i の平文の所持金と更新後の乱数 $r_{k,i}$ を更新する．

6. おわりに

本稿では，プライベートブロックチェーン基盤 Hyperledger Fabric において，取引者同士が互いに所持金の情報を秘匿したまま事前合意を取った上で，信頼できる第三者を置かずに所持金及び取引額に関する正当性を第三者 Peer が検証可能なプロトコルを提案した．所持金及び取引額は加法準同型性を持つ楕円 ElGamal 暗号により暗号化し，各取引ごとに異なる乱数で取引額を暗号化することで，取引額及び所持金の推測を防ぐことができる．また，取引額及び取引後の所持金に関しては，その値が 0 以上であることを Range proof により証明することで，ネットワークに参加する誰もが検証可能となり，信頼できる第三者を必要としない設計にすることができた．

今後の課題としては，Range proof の署名生成及び検証，加法準同型暗号での所持金の計算による計算コストが挙げられる．実験により実際にトランザクション発行にかかる時間を測定し，それが実用的であることを検証する．ここ

© 2018 Information Processing Society of Japan

で，仮想通貨のような数 ms のようなリアルタイムトランザクションへの対応は難しいが，不動産の資産売買など数秒単位のリアルタイムなトランザクションであれば十分に実現可能であると考える．また，各プロセスにおいての最適化についての検討を行う．

参考文献

[1] Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008.
[2] NEM - Distributed Ledger Technology, https://nem.io/.
[3] Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E. and Virza, M., Zerocash: Decentralized Anonymous Payments from Bitcoin, In *Security and Privacy (SP), 2014 IEEE Symposium on*, pp. 459–474, 2014.
[4] Zcash, https://z.cash/.
[5] Kosba, A., Miller, A., Shi, E., Wen, Z. and Papamanthou, C., Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, In *Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 839–858, 2016.
[6] Cecchetti, E., Zhang, F., Ji, Y., Kosba, A., Juels, A. and Shi, E., Solidus: Confidential distributed ledger transactions via PVORM, In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 701–717, 2017.
[7] Monetary Authority of Singapore(MAS), Project Ubin, http://www.mas.gov.sg/Singapore-Financial-Centre/Smart-Financial-Centre/Project-Ubin.aspx, 2017.
[8] Zyskind, G., Nathan, O. and Pentland, A., Enigma: Decentralized computation platform with guaranteed privacy, arXiv preprint arXiv:1506.03471, 2015.
[9] HyperledgerFabric, https://github.com/hyperledger/fabric.
[10] Ethereum, https://www.ethereum.org,
[11] Camenisch, J., Kiayias, A. and Yung, M., On the Portability of Generalized Schnorr Proofs, In *2009, EUROCRYPT*, 2009.
[12] Krawczyk, H. and Rabin, T., Chameleon hashing and signatures, Cryptology ePrint Archive, 1997.
[13] Abe M., Ohkubo, M. and Suzuki, K. 1-out-of-n signatures from a variety of keys, In *Advances in Cryptology - ASIACRYPT 2002, LNCS, Springer-Verlag*, pp. 415–432, 2002.
[14] Chain, https://blog.chain.com/.
[15] Koblitz, N., Elliptic Curves Cryptosystems, In *Mathematics of Computation, 48(1987)*, pp. 203-209, 1987.
[16] El Gamal, T., A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Information Theory*, Vol. 31, No. 4, pp. 469–472, 1985.
[17] Goldreich, O., *Foundations of Cryptography: Volume 2, Basic Applications*, Vol. 2. Cambridge University Press, 2004.

付　　録

A.1　楕円 ElGamal 暗号

本稿では，所持金及び取引額の暗号化に楕円 ElGamal 暗号 [15] を利用する．楕円 ElGamal 暗号は，有限体上の ElGamal 暗号 [16] を楕円曲線上で表した暗号である．平文を m，有限体 $\mathbb{F}_p$ 上の楕円曲線 C のベースポイントを G，秘密鍵を x として公開鍵 $P := xG$ で暗号化する．m の暗号文を $E(m)$ とすると，暗号化は以下の式で表すことができる．

$$E(m) = (m + rP, rG)$$

ここで，秘密鍵 x を求めるためには楕円離散対数問題 (ECDLP) を解く必要があり，それは非常に難しいという仮定の元で安全である．

また平文を m_1, m_2，暗号文をそれぞれ $E(m_1)$ および $E(m_2)$ とすると，以下の式 (A.1) のように乱数 $r_1 + r_2$ を用いて $m_1 + m_2$ を暗号化した値とみることができる．このように楕円 ElGamal 暗号は $m_1 + m_2$ を暗号化したまま計算することができる，加法準同型性を持つ．

$$\begin{aligned} &E(m_1) + E(m_2) \\ &= (m_1 + r_1P, r_1G) + (m_2 + r_2P, r_2G) \\ &= (m_1 + m_2 + (r_1 + r_2)P, (r_1 + r_2)G) \\ &= E(m_1 + m_2) \end{aligned} \tag{A.1}$$

A.2　Schnorr 署名

Schnorr 署名 [11] は電子署名の技術であり，その性質を利用してゼロ知識証明にも応用されている．本稿では，A.3 節で紹介するリング署名に使用されているカメレオンハッシュ [12] を適用した Schnorr 署名を用いる．

A.2.1　鍵生成

G を有限体 $\mathbb{F}_p$ 上の楕円曲線 C のベースポイント，署名対象のメッセージを m，秘密鍵を x として公開鍵 $P = xG$，ハッシュ関数を $\mathrm{H}(\cdot)$，乱数を s とする．
証明者は，$e = H(m||sG - eP)$ となる e を求め，(e, s) を電子署名として出力する．

A.2.2　検証

検証者は，平文 m，ベースポイント G，公開鍵 P，及び (e, s) を用いてハッシュ値 e' を計算し，e と一致することを確認する．

$$\begin{aligned} e' &= H(m||sG - eP) \\ &= e \end{aligned}$$

A.3　リング署名

リング署名とは，あるグループの中の”誰か”からの取引やメッセージであることが，あるグループの中の誰からなのかを特定できない特徴を持つデジタル署名の技術の 1 つ

© 2018 Information Processing Society of Japan

である．本稿では，リング署名として AOS リング署名 [13] を用いる．Schnorr 署名のように $kG = sG - eP$ を e によりコミットするのではなく，e のハッシュ集合 $\{e_i\}_{i=0}^{n-1}$ により $k_iG = s_iG - e_iP_i$ をコミットする．検証鍵数を n とした際の署名生成及び検証のプロセスを以下に示す．

A.3.1 鍵生成

(1) ランダムな整数 k_0 を選び，

$$e_0 = H(m||k_0G)$$

となる e_0 を計算．

(2) ランダムな整数 s_1 を選び，

$$\begin{aligned} e_0 &= H(m||k_1G) \\ &= H(m||s_1G - e_1P_1) \end{aligned}$$

となる e_1 を計算．

(3) 同様に s_2 を選び，$e_1 = H(m||k_2G)$ となるような e_2 を計算．

(4) これを $i = n-1$ まで繰り返す．

(5) $i = n-1$ の時，

$$\begin{aligned} e_{n-1} &= H(m||k_0G) \\ &= H(m||s_0G - e_0P_0) \end{aligned}$$

となる s_0 を計算．

(6) $\sigma = \{e_0, s_0, s_1, ..., s_n\}$ を署名として出力する．

A.3.2 検証

検証者は平文 m，ベースポイント G，公開鍵 $P = \{P_0, ..., P_i, ..., P_n\}$ と署名 σ から検証する．

(1) 署名 σ の e_0, s_1, P_1 より

$$\begin{aligned} e_0 &= H(m||k_1G) \\ &= H(m||s_1G - e_1P_1) \end{aligned}$$

となる e_1 を計算．

(2) 同様に e_1 と署名 σ の s_2, P_2 より，$e_2 = H(m||k_2G)$ となるような e_3 を計算．

(3) これを $i = n-1$ まで繰り返す．

(4) $i = n-1$ の時，e_{n-1} を用いて

$$\begin{aligned} e_{n-1} &= H(m||k_0G) \\ &= H(m||s_0G - e_0P_0) \end{aligned}$$

となる s_0 を計算し，署名 σ の s_0 と一致することを確認する．

A.4 Range proof

本稿ではある暗号化された値が 0 以上であることを示すために，Chain[14] のプロトコルに実装されている Range proof を利用する．[14] では値が 0 以上であることを，その値が $[0, 2^n - 1)$ の範囲にあることにより示している．

A.4.1 Range proof の概要

示したい平文の値を M とし，楕円 ElGamal 暗号で暗号化した値を $E(M) = (M + rP, rG)$ とする．

(1) $E(M) = (M + rP, rG)$ の桁分解を行う．

$(i+1)$ 桁目の M の値を $M_{(i+1)}$ として，

$$\begin{aligned} E(M) &= \sum_{i=0}^{9} Di \\ &= \sum_{i=0}^{9} (M_{(i+1)} \cdot 10^i + r_iP, r_iG) \end{aligned} \tag{A.2}$$

となるような Di を定義する．

(2) 各 $(i+1)$ 桁において，

$$PK_{ij} = Di - j \cdot 10^j (j = 0, 1, ..., 9)$$

となる 10 個の公開鍵を生成する．この時，ある 1 つの PK_{ij} では $j \cdot 10^j$ により $M_{(i+1)} \cdot 10^j$ が相殺されるので，$PK_{ij} = r_iP$ となり，r_j を秘密鍵とするリング署名を生成することができる．

例えば，$M_{(i+1)}$ の値が 5 で，$i = 2$ の時，$PK_{25} = D2 - 500 = r_2P$ となり，r_2P が秘密鍵となる．リング署名のどの鍵によって署名されたのかを特定できない性質により，$M_{(i+1)}$ は 0 OR 1 OR $\cdots$ OR 9 のいずれかであることが証明できる．このリング署名を全ての桁において生成することで，証明したい M が 0 以上の値であることを示すことができる．なお，Chain プロトコルでは 64 ビットの値を想定しているため，扱える桁数は 20 となる．そのため，20 桁それぞれにおいてその桁の値が 0〜9 のいずれかである証明を生成することで，示した M が $[0, 2^n - 1)$ の範囲にあることを示すことができる．

A.5 提案手法における所持金の初期値設定

プライベートネットワークの構築時及び，新たなユーザがネットワークに参加する際には所持金の初期値設定を行う必要がある．これは，$(u_i, u_i, Enc_{pk}(v))$ のようなトランザクションを発行することで所持金の初期化を行うことが可能である．その際には併せて，PS_i には平文 v と，暗号化に使用した乱数 r_k を記録する．

© 2018 Information Processing Society of Japan