［ブロックチェーン技術の最新動向］

② 分散台帳上での匿名送金とその監査について
ゼロ知識証明を利用したセキュアプロトコル

長沼　健　日立製作所

技術背景：透明性 vs 匿名性

近年，Bitcoin，Ethereum をはじめとする暗号通貨，およびそのコア技術であるブロックチェーンが新しい決済システムの基盤として大きな注目を集めている．Bitcoin などのブロックチェーンベースの暗号通貨の特徴として，銀行などの中央機関を必要とせず（非中央集権），低い手数料で決済処理を行える点が挙げられる．たとえば，Bitcoin ではオープンな P2P ネットワーク上に送信された取引情報（以下，トランザクションと呼ぶ）を不特定多数のマイナーと呼ばれるノードが正当性を確認した後，ブロックにまとめられ，Proof-of-Work（PoW）と呼ばれる特定の閾値以下のハッシュ値を求める作業を行うことで承認処理を行っている．承認されたトランザクションは，ブロックチェーンと呼ばれる P2P ネットワーク上の分散台帳に記録される．また，Bitcoin のトランザクション，および分散台帳は，P2P ネットワーク上の全ノードが参照可能であるため，どのユーザからどのユーザにコインの送金が行われたかといった取引に関する情報は，ネットワーク参加者全員が確認できる．結果，非常に透明性の高い送金システムといえるが，一方でこの透明性は，セキュリティの文脈において匿名性と背反する☆1．当然，トランザクションを生成するユーザが，送／受金者，送金額などの機微情報を暗号化することで匿名性を確保可能だが，マイナーがトランザクションの正当性を確認できないといった問題が発生する（たとえば，UTXO：Unspent transaction output モデルにおいてトランザクションの input の金額の合計値は output の合計値と一致しているか，であったり，送金額は正の値か，といった正当性の確認ができない）．

この相反する透明性と匿名性を両立するためにゼロ知識証明と呼ばれる技術の利用が検討されている．たとえば，Zcash（暗号通貨の名称）では，zk-SNARK[1)] と呼ばれるゼロ知識証明プロトコルを用いて，この問題を解決している．具体的には，Zcash ではユーザがトランザクションの送／受金者，送金額といった機微情報を暗号化しつつ，zk-SNARK を使ってマイナーにその正当性を証明することでトランザクションの承認処理を行っている．

本稿では，Ethreum 等でも利用が検討されているこの zk-SNARK と，それを用いた匿名送金機能（Zcash プロトコルの一部）を暗号技術の非専門家向けに解説する．さらに，このような匿名性の高い暗号通貨が，マネーロンダリングなどの不正な送金手段として利用される恐れが指摘されているが，ゼロ知識証明を用いることで，適切な監査機能を実現できることも紹介する．そして最後に，これら暗号通貨，匿名送金機能，その監査の実社会での利用にあたっての課題を述べたい．

☆1　本稿では，匿名化という言葉を「どのノードからどのノードにいくらの送金が行われたか」といった送／受金者のリンク情報，および金額の秘匿化といった意味で用いる．Bitcoin などの暗号通貨では，ノードの ID（アドレスと呼ばれる）はユーザの本名ではなく，ランダムなバイナリ値に仮名化されているため，通常の意味で，ある程度の匿名性は確保されている．

ゼロ知識証明の分散台帳への応用

本章では，ブロックチェーン上での匿名性確保，およびトランザクションの正当性検証に，ゼロ知識証明がどのように利用されているかを解説する．

ゼロ知識証明とは？

ゼロ知識証明とは，ある命題 X（多くの場合，秘密情報に関係する）に対して，証明者 P と検証者 V の間でデータを送受信し，P が V に命題 X が正しいことを確信させ，かつ命題が正しいこと以外の情報を V に与えないプロトコルである．この定義だけでは，何を意図しているのか分かりづらいので，以下，ゼロ知識証明の具体例を見ていく．

たとえば，証明者 P は，アルゴリズムが公開されているハッシュ関数 H とその出力値 A に対して，$H(x)=A$ となるような秘密の入力値 x を知っていることを検証者 V に証明したいとする．当然，P は V に x を開示することで知識を有していることを証明できるが，秘密の入力値 x が漏れてしまいゼロ知識証明とはならない．ゼロ知識証明プロトコルでは，x の代わりに x から生成されるゼロ知識証明値 π を証明者 P が計算し，(π, H, A) を検証者 V に渡し，V が特別な検証処理を行い，V は「命題:P が $H(x)=A$ となるような秘密の入力値 x を知っている」を確認する．このとき，ゼロ知識証明値 π から x に関する情報がいっさい漏洩しない点がポイントである．

分散台帳への応用

具体的にゼロ知識証明が，ブロックチェーンでどのように利用されているかを解説する前に，直感的な利用のイメージを述べる． UTXO モデルでは，トランザクションの input として，まだ使われていない（二重支払でない）コインが指定され，output として，複数のアドレスとそのアドレスに対する送金額が指定される．もしこのとき，匿名性確保のため，トランザクション内のアドレス，送金額情報が暗号化されていたら，マイナーはトランザクションの正当性（二重支払ではないか，input の合計金額は output の合計金額と一致しているか，金額は正の値かなど）が確認できない．この問題を解決するために，トランザクション生成者は，正当性検証用のゼロ知識証明をトランザクションに対して与える．つまり，ゼロ知識証明の文脈で述べると，証明者 P はトランザクション生成者（ブロックチェーンの一般ユーザ）であり，検証者 V はマイナーであり，証明する命題 X は，「このトランザクション内のアドレス，送金額情報等は暗号化されていますが，二重支払いなどの不正はなく，ルールに従い正しく生成されています」である．

zk-SNARK の解説

本章では，Zcash などで利用されている代表的な zk-SNARK : Pinocchio 方式を解説する．

なぜ zk-SNARK なのか？

zk-SNARK とは，zero-knowledge Succinct Non-interactive ARgument of Knowledge の頭文字を取った言葉である．つまり，簡潔（Succinct）かつ非対話型（Non-interactive）なゼロ知識証明という意味である．非対話型という言葉は，ゼロ知識証明の検証処理の際に，証明者 P と検証者 V の間で複数回のデータのやりとりが発生しない，つまり，証明者 P が一方的にゼロ知識証明 π を検証者 V に送り，検証者 V が検証処理を実行することを意味する．前章で述べた通り，ブロックチェーン上でゼロ知識証明を利用する際，P2P ネットワーク上の不特定多数のマイナーが検証者 V となるため，証明者 P が，全マイナーノードと対話を行うことは非現実的である．よって，ゼロ知識証明をブロックチェーン上で利用する際には，非対話型が必須の条件となる．また，簡潔という言葉は，ゼロ知識証明 π のデータサイズが証明したい命題 X を算術回路（もしくは

ブール回路）[☆2]で表現した際のサイズによらず一定であることを意味する[☆3].

表 -1 に zk-SNARK を含む代表的な非対話型ゼロ知識証明方式（zk-SNARK，zk-STARK，Bullet proof）の比較を示す．ブロックチェーン上でゼロ知識証明を利用する際に，トランザクションデータにゼロ知識証明 π を追加するが，このデータサイズによって，トランザクションの手数料が変わる[☆4]．よって，手数料削減かつ台帳サイズ圧縮によるスケーラビリティ向上のために，ゼロ知識証明値のデータ長が短い（命題 X の回路サイズの増加に対して証明長が一定），簡潔な方式が望ましい．以上，2 つの理由から zk-SNARK がブロックチェーンで利用されている．一方で，現状提案されている zk-SNARK 方式は，セットアップの際に，信頼できる第三者機関が必要であり，かつ量子計算機に対する耐性を持たない点に注意する．

☆2 通常，ゼロ知識証明技術では，証明したい命題を算術回路（もしくはブール回路）で表現し，証明者が規定された出力値を与える入力値を持つことを証明する．

☆3 さらに検証者の計算量が回路の既知入力サイズの定数倍以下である必要もある．

☆4 本稿執筆時（2019 年 9 月）の Bitcoin では，1byte あたり 0.0000005BTC（=50 satoshi）の手数料設定で，30 分程度でブロックチェーンに取り込まれている．手数料はマイニングに対する報酬となるため，手数料の低いトランザクションはブロックに取り込まれづらくなる．

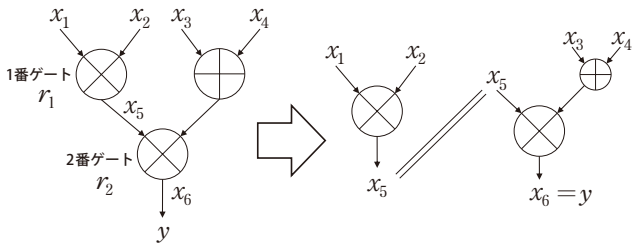

図 -1　算術回路とその R1CS 表現

Pinocchio 方式の概要

本節では，Zcash にも利用されている代表的な zk-SNARK である Pinocchio 方式を解説する（いくつか説明を簡略化している．より正確な方式を知りたい読者は文献 1) を参照されたい）．

数式を用いた説明の前に，直観的な方針を述べると，回路中の全ゲートの左からの入力値，右からの入力値，と出力値の間にある満たすべき関係式を多項式で表現し，証明者の知識をその多項式の係数（を知っていること）に変換する．さらに，証明者はその多項式の係数を暗号化し，検証者に渡し，検証処理を行うことで知識証明を行う．この暗号化状態での検証処理にはペアリング暗号の技術を用いる．

以下，図 -1 の左側に示すような 4 つの入力値 $[x_1, x_2, x_3, x_4]$ と 1 つの出力値 $[y]$ を持つ，深さ 2 の関数（算術回路）$f(X_1, X_2, X_3, X_4)=(X_1\times X_2)\times(X_3+X_4)$ を考える．この関数 f の出力値 y に対して $f(x_1, x_2, x_3, x_4)= y$ となる入力値 $[x_1, x_2, x_3, x_4]$ を知っていることをゼロ知識証明する例を考える（実際にはハッシュ関数のような複雑な算術回路に対して証明を行う）．

Pinocchio 方式では信頼できる第三者によって，全ユーザが利用するシステムパラメータを生成する事前の処理が必要である（Setup 処理と呼ぶ）．

Setup 処理（信頼できる第三者機関が事前に行う）

1. 関数 f 内の各掛算ゲートに番号を振り（図中の 1 番ゲート，2 番ゲート），その出力値を新しい変数 $[x_5, x_6]$ とする．
2. 各掛算ゲートに対して乱数 $[r_1, r_2]$ を生成し，ターゲット多項式 $t(X):=(X-r_1)(X-r_2)$ を計算する．
3. 関数 $\phi_1(X)$ を $\phi_1(r_1)= 1, \phi_1(r_2) =0$ となるよう

表 -1　ゼロ知識証明方式の比較表（N は回路サイズ）

	証明サイズ	P の計算量	V の計算量	第三者機関	耐量子性
zk-SNARK	288bytes	$O(N*\log(N))$	$O(1)$	必要	なし
zk-STARK	45 ～ 200KB	$O(N*\text{poly}\log(N))$	$O(N*\text{poly}\log(N))$	必要なし	あり
Bullet proof	～ 1.5KB	$O(N*\log(N))$	$O(N)$	必要なし	なし

に生成する（1 番ゲート上で値が 1，2 番ゲート上で値が 0 となるラグランジュ型補間多項式）．同様に関数 $\phi_2(X)$ を $\phi_2(r_1)=0$, $\phi_2(r_2)=1$ となるように生成する．

4. $t(X)$ と $\{\phi_1(X), \phi_2(X)\}$ を公開する．

このとき，多項式

$$x_1\phi_1(X) \times x_2\phi_1(X) - x_5\phi_1(X) \qquad (1)$$

は，$X=r_2$ を解に持ち，さらに $X=r_1$ を解に持つ必要十分条件が，$x_1 \times x_2 = x_5$ であることに注意する（1 番ゲートに対応する制約条件）．また，多項式

$$x_5\phi_2(X) \times (x_3\phi_2(X)+x_4\phi_2(X)) - x_6\phi_2(X) \qquad (2)$$

は，$X=r_1$ を解に持ち，$X=r_2$ を解に持つ必要十分条件が，$x_5 \times (x_3+x_4) = x_6$ であることに注意する（2 番ゲートに対応する制約条件）．この算術回路の変形で重要な点は，深さ 2 の算術回路が式（1），（2）のような，いくつかの制約条件の付いた深さ 1 の回路（Rank-1 Constraint System）で表現可能である（図 -1 の右側参照）．より一般の回路に対しても，同様の方法で深さ 1 の回路と制約条件で表現できる．

この多項式と各ゲートに対して $A(X)$:=[左からの入力値]，$B(X)$:=[右からの入力値]，$C(X)$:=[出力値] をまとめると，

$$\overbrace{(x_1\phi_1(X)+x_5\phi_2(X))}^{\text{左からの入力値}} \times \overbrace{(x_2\phi_1(X)+x_3\phi_2(X)+x_4\phi_2(X))}^{\text{右からの入力値}}$$
$$-\overbrace{(x_5\phi_1(X)+x_6\phi_2(X))}^{\text{出力値}} = A(X)B(X)-C(X) =: P(X),$$

$P(X)$ が $X=r_1, r_2$ を解に持つ（ターゲット多項式 $t(X)$ で割り切れる）ための必要十分条件は，$x_1 \times x_2 = x_5$ かつ $x_5 \times (x_3+x_4) = x_6$ が成立することとなる．このように多項式 $P(X)=A(X)B(X)-C(X)$ がターゲット多項式 $t(X)$ で割り切れるかどうかによって算術回路の正しい入出力値のペアか表現することを QAP : Quadratic Arithmetic Program と呼ぶ．

今，正しい知識 $f(x_1, x_2, x_3, x_4)=y$ を満たす，(x_1, x_2, x_3, x_4) を有する証明者 P は，実際に (x_1, x_2, x_3, x_4) を代入することで，$A(X)=a_2X^2+a_1X+a_0$ の係数を計算可能である．$B(X)$，$C(X)$ に関しても同様である．さらに，$P(X)$ は $t(X)$ で割り切れるので，$H(X) := P(X)/t(X) = h_2X^2+h_1X+h_0$ の係数も計算可能である．この計算によって，証明者 P の持っている知識が多項式の係数情報に変換された．よって，証明者 P は，「命題：ターゲット多項式 $t(X)$ に対して

$$A(X)B(X) - C(X) = H(X)t(X) \qquad (3)$$

を満たす多項式 $A(X), B(X), C(X), H(X)$ の係数を知っている」を検証者 V に証明することで，本来示したい「命題 : $f(x_1, x_2, x_3, x_4)= y$ を満たす，(x_1, x_2, x_3, x_4) を知っている」を証明可能である．これにはペアリング暗号の技術を利用する．ペアリング暗号とは，ペアリングと呼ばれる双線形写像を用いた公開鍵暗号の一種である．

$$e : G_1 \times G_2 \longrightarrow G ; (g_1^a, g_2^b) \longmapsto e(g_1, g_2)^{ab},$$

を（アルゴリズムが公開されている）ペアリング写像とする．先ほどの Setup 処理実行者は，事前に乱数 s を生成し，$(g_1, g_1^s, g_1^{s^2}, g_2, g_2^s, g_2^{s^2}, g_2^{t(s)})$ をシステムパラメータとして公開する．このパラメータを利用することで，$A(X)= a_2X^2+a_1X+a_0$ の係数 (a_0, a_1, a_2) を知る証明者 P は，$g_1^{a_0} \cdot (g_1^s)^{a_1} \cdot (g_1^{s^2})^{a_2} = g_1^{A(s)}$ を計算可能である．同様に，$g_2^{B(s)}, g_1^{C(s)}, g_1^{H(s)}$ も計算可能である．つまり，多項式 $A(X), B(X), C(X), H(X)$ の点 s での評価値が，指数部分となる離散群の元を求めることが可能である． 証明者 P は，ゼロ知識証明値 π として，

$$\pi=(g_1^{A(s)}, g_2^{B(s)}, g_1^{C(s)}, g_1^{H(s)})$$

を検証者 V に渡す[☆5]．そして検証者 V は，システムパラメータの $g_2^{t(s)}$ を使い，以下の等号が成立するかを確認する．

$$e(g_1^{A(s)}, g_2^{B(s)})=e(g_1^{H(s)}, g_2^{t(s)})\cdot e(g_1^{C(s)}, g_2)$$

左辺の指数部分は $A(s)\cdot B(s)$ であり，右辺の指数部分は $H(s)\cdot t(s)+C(s)$ となる．もし，証明者 P が正しい知識を持ち，$A(X), B(X), C(X), H(X)$ を計算可能であれば，(3) 式が成立するため，上式の等号も成立する．検証者 V は，この等号の成立を確認することで，証明者 P が正しい知識 $f(x_1, x_2, x_3, x_4)=y$ となる (x_1, x_2, x_3, x_4) を知っていることを確認する．注意として (3) 式は，多項式としての等号であり，それを点 s での値の一致のみで検証を済ますことに違和感を感じるかもしれない（偶然，値が一致する場合があるので），しかし実用上は，ペアリングのパラメータを大きく設定することで，偶然一致する確率を限りなく 0 に近づけることが可能である．また，この構成方法から分かるように Setup 処理時の乱数 s は秘密のパラメータであり，g_1^s から s を求めることが計算量的に困難であるなどの仮定を必要とする．同時に乱数 s が秘密情報であることから，zk-SNARK の運用には信頼できる第三者による Setup 処理が必要な理由も明らかである．

実は上述のプロトコルは正しい知識を持つことを証明しているが，ゼロ知識証明にはなっておらず，π から入力値に関する情報が漏れる．$A(X)$ 等に乱数 δ で $A(X)+\delta t(X)$ とマスクすることで，ゼロ知識性を達成することが可能である．

[☆5] 正確には，f の出力値 y に対応する部分 $x_6=y$ の代入計算は検証者側で実行する．

Zcash プロトコルの解説

本章では，代表的な匿名暗号通貨である Zcash において送／受金者のリンク情報がどう秘匿化されているかを解説する．

図 -2 は，Alice から Bob に送金を行う際のイメージである．H をアルゴリズムが公開されているハッシュ関数とし，システム管理者，もしくは信頼できる第三者によって，以下に示す手順 4 の命題に対する Pinocchio 方式のシステムパラメータが生成，公開されているものとする．送金の処理手順は以下のとおりである．

1. Alice は受金者アドレスを秘匿（図では− − −）したトランザクションをブロックチェーンに

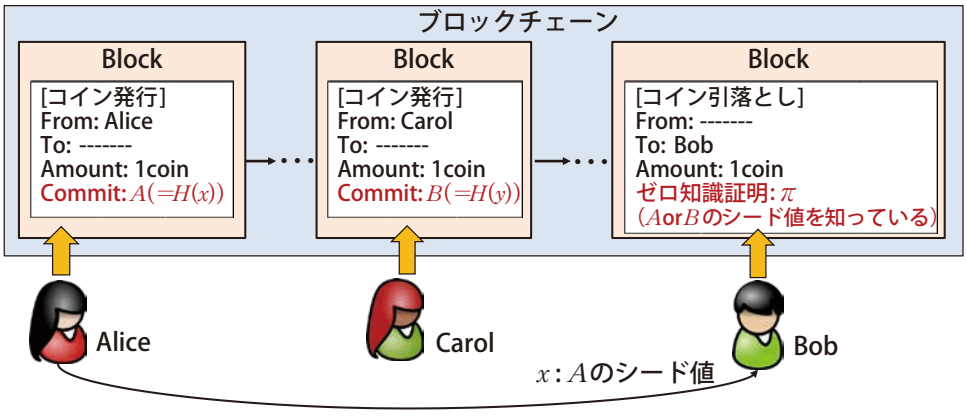

図 -2　送金を行う際のイメージ

送信する．この際に秘密のシード値 x を生成し，そのハッシュ値 $A=H(x)$ をコミットメント値としてトランザクションデータに追加する（匿名コイン発行）．

2. Carol も同様に受金先を秘匿化し，コミットメント値 $B=H(y)$ 付きトランザクションをブロックチェーンに送信する（匿名コイン発行）．
3. Alice は Bob に発行済み匿名コイン A のシード値 x を渡す．渡す方法はブロックチェーン外部の通信手段でもよいし，Bob の鍵で暗号化してブロックチェーンに書いてもよい．
4. Bob は発行済みの全匿名コインのコミットメント値（図では A, B）に対して，「命題：ハッシュ関数 H の出力値が A or B となる入力シード値を知っている」を証明するゼロ知識証明 π を生成し，送金者アドレスを秘匿化したトランザクションデータにこの証明を追加し，ブロックチェーンに送信する．

上述のプロトコルにおいて，ゼロ知識証明を検証するマイナー，およびほかのユーザは，実際に Bob がコミットメント値 A，もしくは B のどちらのシード値を知っていたかは分からない．しかし，少なくともどちらか 1 つのシード値を知っていたことは確認できる（コインを受け取る権利は持っている）．結果として Alice から送金が行われた事実を秘匿することが可能である．このプロトコルでは送／受金者のリンクは秘匿化されているが，送／受金者のアドレス，送金額は秘匿化されていない（たとえば，ユーザごとに送金額が異なる場合，送金額から受金者が特定可能である）．また，Alice による二重支払いや Bob による二重引落しを防止する機能もない． 実際の Zcash プロトコルでは，より複雑なトランザクション（Sprout/Sapling）と，それに対応したゼロ知識証明を利用して，これらの問題に対応している．

匿名送金に関する監査

前章までに述べたように，匿名暗号通貨はユーザのプライバシを守る一方で，マネーロンダリングなどの不正な送金手段として利用される恐れが指摘されている．

匿名暗号通貨が実社会に受け入れられるためには，従来の金融機関，もしくはほかの仮想通貨と同様，当局による適切な監査に対応する必要がある．以下に述べるように，ゼロ知識証明を再び利用することで，一般のブロックチェーンユーザには，送金の流れが秘匿化されているが，監査当局には秘匿化が解除可能な仕組みが構築できる．

匿名暗号通貨において，通常の暗号通貨のように監査を実現する簡単な方法は，トランザクション内で秘匿化されている情報を監査者の公開鍵で暗号化し，監査用の情報としてトランザクションデータに追加することである．こうすることで一般のブロックチェーンユーザに情報を秘匿しつつ，秘密鍵を持つ監査者にのみ情報へのアクセス権限を与えることが可能である．しかし，この際に問題となるのはマイナーの存在である．マイナーは監査者ではないので，暗号化されている監査用の情報が，正確な情報なのか確認できない．この問題に対して，トランザクション生成者は，ゼロ知識証明を利用することで，暗号化されている監査用の情報が正しく生成されていることを証明できる．つまり，「命題：暗号化されている監査用の情報は，監査者の公開鍵で暗号化されており，さらにその平文はトランザクション内の秘密情報と一致する」を zk-SNARK などを使って証明する．たとえば前章の例では，Bob が実際には $A=H(x)$ のシード値を知っていることを示す情報を監査者の公開鍵で暗号化し，さらにそのことを示すゼロ知識証明をトランザクションデータに追加することで，監査者は Alice から Bob への送金の流れを把握することが可能である（具体的な方式は，文献 2）参照）．

暗号通貨の価値から見える課題

暗号通貨が持つ本来の価値として，非中央集権性とは別に手数料の安い決済手段といった側面がある．たとえばBitcoinでは，手数料自体をBTCで払っているため，1BTCの価格が高騰するとそれに合わせて手数料も高騰する．その結果として，安い決済手段としての価値が低落し，価格に対して負のフィードバックが働く．

よって最終的にある均衡点となる金額に落ち着くが，その金額は銀行，クレジットカードなどの通常の決済手段に比べて安価であることが望ましい（そうでなければ既存の決済手段が利用される）．2019年現在，暗号通貨に関しては投機的な取引が活発なため，まだこの均衡点には到達していない．

本稿では代表的な匿名暗号通貨Zcashとその基盤技術であるゼロ知識証明方式zk-SNARK，それを使った監査機能の実現方法について解説した．

すでに見てきたとおり，匿名暗号通貨では，誰が誰にいくら送金したか，といった情報を第三者に秘匿することができる．結果として投機的な売り買い情報が見えにくくなり，通貨価格の安定化が期待される．

現状，ゼロ知識証明を使った匿名暗号通貨は，トランザクションのデータサイズが増加するため，手数料の観点からまだ一般の利用割合は高くない．

一方で秘匿性の高さからマネーロンダリングなどの不正な送金手段として利用されている．この問題に対しては，監査機能を付与し適切な監視を行うことが望ましい．当然，監査機能の追加はさらなる手数料の増加を招く．良貨が悪貨に駆逐されないためにもインセンティブを与えるなどの利用促進に向けた施策，運用が望まれる．

参考文献

1) Parno, B. et al. : Pinocchio: Nearly Practical Verifiable Computation, In IEEE Symposium on Security and Privacy, S&P (2013).
2) Naganuma, K. et al. : Auditable Zerocoin, In European Symposium on Security and Privacy Workshops, EuroS&PW (2017).

(2019年10月8日受付)

長沼　健　ken.naganuma.dn@hitachi.com
日立製作所研究開発グループ所属，東京大学大学院新領域創成科学研究科博士課程在学中，入社以来モバイルアプリケーション，GPS利活用，車載無線セキュリティ，医療データ分析の研究開発に取り組む．2014年より暗号通貨，ブロックチェーンの研究開発に携わる．