IMES DISCUSSION PAPER SERIES

分散型デジタルアイデンティティとは？
～概念、仕組み、実現に資する技術と課題～

佐古和恵（さこかずえ）

Discussion Paper No. 2023-J-8

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

日本銀行金融研究所

〒103-8660 東京都中央区日本橋本石町 2-1-1

IMES Discussion Paper Series 2023-J-8
2023 年 7 月

分散型デジタルアイデンティティとは？
～概念、仕組み、実現に資する技術と課題～

佐古 和恵 *

要　　旨

本稿では、インターネット上において個人の属性管理や認証を行うための新しい手段として標準化が進められている分散型デジタルアイデンティティの概念、および、それを実現する技術を紹介する。従来、各個人の属性を含むユーザの情報は、コンピュータ管理者やサービス提供者によって管理されてきた。これに対して、分散型デジタルアイデンティティは、ユーザ情報のコントロールはユーザ自身で行うという考え方に基づく。信頼できる機関がユーザの属性を保証する検証可能クレデンシャル（Verifiable Credential）を発行し、各ユーザがそれを提示することによって自分の属性を他者に示す。また、効率証明付き署名とゼロ知識証明技術を活用することで、検証可能クレデンシャルに記述された属性のうち必要なものだけを開示することもできる（選択的開示）。個人の識別子の管理方法として、ブロックチェーンなどを用いた分散型識別子（Decentralized Identifier）による方法も検討されており、今後の研究・開発の動向が注目される。

キーワード：検証可能クレデンシャル、効率証明付き署名、ゼロ知識証明、ブロックチェーン、分散型識別子、分散型デジタルアイデンティティ
JEL classification: O31、O35、Z00

*　早稲田大学理工学術院教授（E-mail: kazuesako@aoni.waseda.jp）

本稿は、筆者が日本銀行金融研究所客員研究員の期間に行った研究をまとめたものである。本稿の作成に当たっては、崎村夏彦氏（OpenID Foundation 理事長）から有益なコメントを頂戴した。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者個人に属し、日本銀行の公式見解を示すものではない。また、ありうべき誤りはすべて筆者個人に属する。

目　次

1. はじめに

個人（ユーザ）がインターネット上のサービスを利用する際、多くのケースでは自分の属性に関する情報（属性情報）をコンピュータ管理者やサービス提供者（以後、両者を総称する場合、サービス提供者等と呼ぶ）に提供するよう求められる。情報を受け取ったサービス提供者等は、ユーザごとにユニークな識別子を割り当て、属性情報を認証情報（パスワードや暗証番号）とともに管理することで、ユーザのアクセス制御を行っている。多くの場合、ユーザにはサービスごとに別々の認証情報を設定することが求められるため、利用するサービスが多くなれば、それに伴い属性情報と認証情報の管理負担も大きくなる。また、サービス提供者等においては、大量のユーザに関する情報を管理する必要があり、サイバー攻撃や内部の不正行為によってそれらが外部に流出するリスクを抱えることになる。

近年、さまざまなサービスに共通して利用可能な属性情報を提供する主体（IDプロバイダ）が登場し、ユーザやサービス提供者等における情報管理の負担軽減が図られている。もっとも、IDプロバイダにおいて情報流出等のインシデントが発生するリスクは引き続き存在していることから、従来からある課題がすべて解決されたわけではない。さらに、IDプロバイダが仲介するスキームになったことにより、ユーザのどの情報がサービス提供者に実際に提供されたかについて、ユーザが確認することができないという新たな課題を内在することになった。また、IDプロバイダの機能が停止すれば、IDプロバイダが仲介していたサービスすべてを使用できなくなるというリスクもある。そのほか、個々のユーザによる各サービスへのアクセス状況がIDプロバイダによって知られる可能性がある（プライバシ上の問題）といった課題が残されている。

こうした諸課題に対して、近年、インターネット上における個人の属性管理や認証の新しい手段として、分散型デジタルアイデンティティと呼ばれる概念が注目を浴びている。分散型デジタルアイデンティティとは、ユーザが自身の属性情報の管理に関与するという考えを具現化するものである。具体的には、信頼できる機関が属性情報に対する保証書として検証可能クレデンシャル（Verifiable Credential）を各個人に発行し、各個人がそれをサービス提供者等に提示することによって自分の属性を他者に示すことが考えられている。検証可能クレデンシャルには信頼できる機関によるデジタル署名が付与されており、サービス提供者等はその署名を検証することによってユーザの属性が信頼できる機関によって保証されたものであることを確認できるようになる。

上記で説明した検証可能クレデンシャルを発行する方法については、World

Wide Web Consortium（W3C）[1]において標準化されつつある。ユーザは、自分の検証可能クレデンシャルを用いることによって、ID プロバイダ等に仲介してもらうことなく、自分の属性を第三者に示すことができる。このような方法で属性を第三者に示すことを検証可能提示（Verifiable Presentation）と呼ぶ。また、効率証明付き署名とゼロ知識証明技術を活用することによって、検証可能クレデンシャルに含まれる属性情報のなかからユーザが選択したもののみを開示することもできる（選択的開示）。この選択的開示は、ユーザのプライバシに配慮した検証可能提示である。

今後、金融機関が新規顧客の属性情報を入手・管理する際、あるいは、既存顧客の情報を他機関へ連携する際に、分散型デジタルアイデンティティを取り入れれば、顧客の意思を反映した形で属性情報のやり取りを実施することができると期待される。また、検証可能クレデンシャルを活用できるようになれば、金融機関が顧客の様々な属性情報を容易かつ確実に確認できるようになるなど、管理コストの削減にも繋がる可能性がある。これは、金融業務におけるデジタル・トランスフォーメーションの 1 つといえよう。また、こうした仕組みを金融サービスに取り入れることによって、より高いプライバシの保護や属性情報等の自己コントロールを志向する顧客から当該サービスへの支持を得ることも期待できる。

本稿では、2 節において、分散型デジタルアイデンティティが求められる背景を述べ、3 節において、検証可能クレデンシャルの仕組みを述べる。4 節では、選択的開示を実現する方式を紹介する。5 節では、分散型識別子とブロックチェーンの活用可能性について述べる。分散型識別子は、個人が中心となって識別子を管理できるように設計された識別子体系の総称である。これらは必ずしも分散型デジタルアイデンティティにとって必須の技術ではないものの、検証可能クレデンシャルの概念と一緒に議論されることが多いことから本稿でも取り扱うこととする。

2. 分散型デジタルアイデンティティが求められる背景

本節ではデジタルアイデンティティを一元管理するサービスが誕生した経緯と、同サービスの概要を説明する。そのうえで、こうしたサービスがもつ課題を整理し、同課題を解決する方法として分散型デジタルアイデンティティの検討が進められた背景を紹介する。

---

[1] W3C は、World Wide Web で使用される各種技術の標準化を推進するために設立された標準化団体である。

2.1 デジタルアイデンティティの管理

アイデンティティとは、属性の集合と定義され[1]、デジタルアイデンティティは、コンピュータで取り扱うことのできるアイデンティティのことを指す。歴史的にみると、デジタルアイデンティティの管理は、コンピュータを複数のユーザが共同で使用する際に、各ユーザからのアクセスを適切に制御することが主な目的であった。その後、コンピュータの活用が広がったことで、アクセス制御のみならず、どのユーザにどのようなサービスを提供するかというマーケティング目的にも活用されるようになり、機微な個人情報も取り扱われるようになった。本節では、当初どのように属性が管理されてきたかを振り返る。

複数のユーザが 1 台のコンピュータを共同利用する場合、コンピュータ管理者はそのコンピュータ上に各ユーザのアカウントを作成し、管理に必要なユーザに関する情報（ID 情報、狭義のデジタルアイデンティティ [2]）をコンピュータに登録する。こうした ID 情報は、どのユーザにどの資源へのアクセスやどのサービスの利用を許可するかといったアクセス制御に利用され、主に次の 3 種類のデータが含まれる。

① 登録したユーザ [3]を一意に識別できる識別子
② 登録したユーザがログインしたことを確認できる認証情報（例えば、パスワード）
③ ログインしたユーザがどのような資源にアクセスできるかを判断するのに必要なユーザの属性情報（例えば、所属グループ名や権限情報）

ユーザの認証情報や属性情報が誤って登録されたり、登録後にそれらの情報が改ざんされたりすれば、アクセス制御が正しく実行されない。そのため、コンピュータ管理者には、正しいと認めた認証情報と属性情報のみを登録し、コンピュータ管理者以外は変更 [4]できない場所にそれらの情報を保管するといった対応が求められる。

---

[2] デジタルアイデンティティとは、広義には、ユーザの属性を表す情報全般を指し、コンピュータ管理に用いる ID 情報（識別子、認証情報、資源へのアクセスの可否を判断するための属性情報）はその一部である。

[3] 実際には、ユーザがアクセスするコンピュータ内の資源の属性を管理するための ID 管理も必要になるが、ここではユーザだけを対象として説明する。

[4] ユーザ自身が認証情報を更新することができる場合もあるが、それはコンピュータ管理者が正しいと認めた更新手続きに従ったものに限られる。

2.2　ネットワーク化と ID 情報のドメイン内一元管理

コンピュータが単体で動いていた時代であれば、コンピュータ管理者は当該コンピュータを利用するユーザの ID 情報さえ管理すればよかった。しかし、コンピュータがネットワーク化されると、自身の管理するコンピュータ内の資源に別のコンピュータのユーザがアクセスしてくるようになった。コンピュータ管理者が、ユーザによるアクセスの可否を ID 情報に基づいて判断する場合には、そのユーザに関する信頼できる ID 情報の入手が必要となる。同じセキュリティポリシーの下で管理されているコンピュータ同士であれば、それぞれがもつユーザの ID 情報を信頼して共有することができるが、異なるセキュリティポリシー配下のコンピュータの場合には、信頼できる ID 情報を入手する手段を確立することは難しい。そこで、同一のセキュリティポリシーで稼働するコンピュータの集合（ドメイン）内であれば、共有された ID 情報に基づくアクセス制御を実施するが、異なるドメインの場合には、どこのドメインからアクセスされたかという情報に基づいたアクセス制御が採用されるようになった。

自身の管理下にないユーザの ID 情報の正しさを確認する手続きは、すべてのコンピュータ管理者に対して同じように求められる。そのため、各コンピュータ管理者がドメイン内のすべてのユーザの ID 情報を管理する場合には、ドメイン全体でみたときの ID 情報の管理にかかるコストが大きくなってしまう[5]。

そこで、ドメインでの ID 情報の管理コストを下げるために、ドメイン内の ID 情報を一か所で管理する形態（ドメイン単位ですべての ID 情報を管理）への移行が進められた。例えば、マイクロソフト社の windows[6]によるネットワーク管理では、ドメイン内のコンピュータをドメインコントローラと呼ばれるサーバが一元的に管理する仕組みを導入している[17]。こうした管理形態では、すべてのユーザの ID 情報を一元的に管理する主体を運営するコストが必要となるものの、ドメイン内で ID を管理するコンピュータ管理者の数とドメイン全体で管

---

[5] N 台のコンピュータ（とそれぞれのコンピュータにコンピュータ管理者）が存在し、それぞれのコンピュータに M 人のユーザが存在するとする。このとき、コンピュータがネットワーク化されていないときは、1 人のコンピュータ管理者が管理すべき ID 情報の個数は M 個である。一方、ネットワーク化され、各ユーザが N 台のどのコンピュータにもアクセスできるようする場合には、各コンピュータ管理者はそれぞれ他の N 台のコンピュータからのアクセスに備えて M×N 個の ID 情報を管理する必要がある。また、自身のコンピュータでアカウントが追加・削除されたり、ユーザの情報が更新されれば、それ以外の N−1 台に速やかに共有する必要も生じる。すなわち、ドメイン内の N 人のコンピュータ管理者はそれぞれに大きな管理コストを負担することになる。

[6] マイクロソフト、および、windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標である。

理するID情報の量を削減することができる[7]。また、ID情報が更新された際の対応も容易になる。一方、ユーザにおいても、ドメイン内のコンピュータごとにアカウントを作成したり、認証情報を複数管理したりする必要がなくなることから、利便性が向上する。

2.3 アプリケーションサービスとデジタルアイデンティティの一元管理

（1）デジタルアイデンティティ提供サービス

ID情報を一括して管理するメリットはアプリケーションレベルのユーザ管理にも同様に当てはまる。コンピュータレベルのユーザ管理であれば、上述のように、識別子や認証情報といったアクセス制御に必要なID情報の管理で十分であった。しかし、アプリケーションレベルでは、ユーザへの細かいサービスの提供や、新しいサービスの検討に活用するため、ユーザの住所や電話番号、サービス利用履歴といったさまざまな属性情報（広義のデジタルアイデンティティ）が収集されている。このように、サービス提供者にはユーザに関するさまざまな属性情報の入手・管理が必要となっているが、こうした属性情報の管理には相応のコストがかかるため、新規事業者による同サービスへの参入はますます難しくなっている。これに対して、属性情報の活用によって事業の拡大を見込む先は、積極的にユーザの属性情報の収集を進めている。

このような状況に対して、デジタルアイデンティティを一括して管理・提供してくれるIDプロバイダが登場した。IDプロバイダと契約すれば、サービス提供者側ではユーザの認証情報や属性情報を管理する必要がなくなる。IDプロバイダにアカウントをもつユーザは、自分の属性情報を提供することなく、IDプロバイダと契約するサービス提供者のサービスを利用できるようになるほか、管理すべき認証情報も1つに限定できる。多くの便利なサービスが特定のIDプロバイダと連携していれば、そこに多くのユーザが集まり、さらに新規ユーザ獲得を狙った新たなサービスが連携するというネットワーク効果が生み出される。

上記スキームにおいて、ユーザ、サービス提供者、IDプロバイダの3者間におけるID情報の受け渡しは以下の手順で行われる（図1参照）。なお、ユーザは事前にIDプロバイダにアカウントを作成し、ID情報（識別子、認証情報、属性情報）を登録しているものとする。

① サービスを利用するため、ユーザは、サービス提供者にアクセスする。
② サービス提供者は、当該ユーザの情報をIDプロバイダに問い合わせる。

---

[7] ドメイン内のN台のコンピュータにM人のユーザが存在したとしても、必ずしもM×N人のユーザが存在するわけではなく、ユーザの重複がある場合がある。したがって、ドメイン全体で管理するIDの個数はM×N個ではなく、重複を省いた個数に削減できる。

③ IDプロバイダは、ユーザにログイン画面を提示し、ユーザはログイン画面に識別子と認証情報を入力する。
④ IDプロバイダは、ユーザの認証後、ユーザの許可に基づいて当該ユーザの属性情報をサービス提供者に提供する。
⑤ サービス提供者は、提示された属性情報が正しくIDプロバイダから発行されたものであることを確認し、この情報をもとに、ユーザにサービスを提供する。

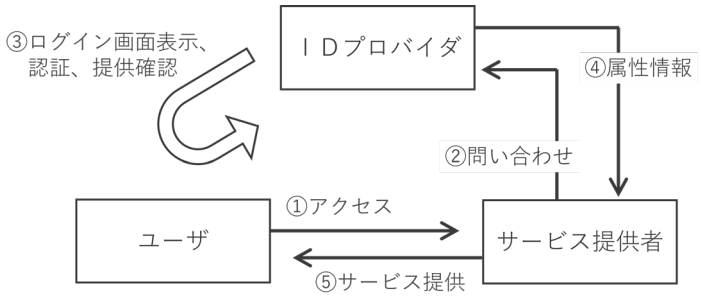

図 1：ID プロバイダによる認証と属性情報の提供

（2）ID プロバイダの情報優位性とユーザ視点での問題点

ID プロバイダを用いて認証情報の管理コストやユーザの属性情報の収集コストを削減したことにより、ユーザとサービス提供者の利便性は向上した。一方、ID プロバイダにユーザの ID 情報が集約されたことにより、ID プロバイダが情報優位性[8]を持つ結果、以下の問題が発生する可能性がある。

- ユーザがアプリケーションを使用する都度、ID プロバイダに認証情報や属性情報の問い合わせが発生する。この問い合わせにより、ID プロバイダに、ユーザが利用するサービスや、その利用頻度が知られる[9]。
- ユーザとの契約に反して、ID プロバイダの一存で当該ユーザの属性情報が開示される恐れがある。
- ユーザとの契約に反して、ID プロバイダからサービス提供者に属性情報

[8] インターネット上のサービスを提供するための基盤を提供するプラットフォーマーが ID プロバイダとしての役割を担うケースが多いため、その点をあわせても情報優位性が顕著である。

[9] 具体的には、問い合わせ結果に基づいてアクセストークンやクッキーをサービス提供者側が発行し、以後それに基づいてアクセスを許可することが多い。したがって、前回発行されたトークン等が有効であれば再度の問い合わせが発生しないが、問い合わせ頻度から使用の頻度がある程度類推される。

が提供されていても、ユーザがそれを認知することが容易でない恐れがある。

- ID プロバイダの一存で、特定のユーザの属性情報がサービス提供者に開示されない恐れがある[10]。
- ID プロバイダの一存で、特定のユーザが（当該 ID プロバイダと契約しているサービス提供者の）サービスを利用できなくなる恐れがある。

これらは、ID プロバイダが信頼できる機関でなかった場合に想定されるリスクである[11]。こうした状況は、ID プロバイダが意図的にユーザの属性情報を不当に取り扱う場合と、ID プロバイダによる過失や不十分なセキュリティ対策の結果として引き起こされる場合がある。

（3）その他の問題点

本節では、サービス提供者と ID プロバイダからみた問題点を述べる。まず、サービス提供者の視点では、ユーザ視点と同様、以下の 3 点が問題点としてあげられる。

- ID プロバイダに認証情報や属性情報を問い合わせるため、ID プロバイダに自身のユーザのことを確実に知られる。
- ID プロバイダの一存で、ID 情報が提供されない恐れがある。その場合、サービスを提供できなくなる。
- ID プロバイダがサービス中断状態になると、自社のサービスを中断せざるを得なくなる恐れがある。

ID プロバイダの観点では、ユーザ視点での問題点として整理したとおり、ID プロバイダからユーザの属性情報が漏洩すれば、ID プロバイダに対する顧客（サービス提供者）からの信頼は失墜する。こうしたビジネスリスクを軽減させるためには確実なビジネス運営とセキュリティ対策が必要である。しかし、一定水準のコストをかけてビジネスを運営できる事業者は限られている。さらに、多数のユーザを抱える ID プロバイダの方が、サービス提供者にとって連携するメリットが大きいことから、新規参入の障壁が高く、ID プロバイダ間での競争原理が働きづらくなっているのが実情である。その結果、少数の ID プロバイダによる寡占状態が続き、ユーザやサービス提供者には ID プロバイダの選択肢がなくなってしまっている。そのため、仮に、ID プロバイダに多少の不満があった

---

10 一方で、ID プロバイダが不適切なサービス提供者への情報提供を事然にくいとめることができるという効用がある。

11 プロバイダが仲介するスキームの 1 つに OpenIDconnect があるが、OpenIDconnect はユーザが設定した本人のポリシーに基づいて ID プロバイダが情報を提供する仕組みとなっていることから、ユーザが自身の情報をコントロール可能であるといえる。

り、全面的に信頼できない場合であっても、ユーザやサービス提供者は現行のIDプロバイダを使い続けるしかないという問題がある。

2.4 事前発行の証明書による解決

2.3 節で述べた懸念は、ID プロバイダがユーザ認証を仲介し、ユーザの属性情報をアプリケーションサービスに提供する形態であること、また、その適切な執行を契約によってのみ保証していることに起因する。このような第三者を通じた情報提供ではなく、ユーザ自身が属性を示すことができる方式へのニーズが高まっており、そうした状況を受けて、W3C において提示方法のデータモデルの検討が進められている[2]。

この方式は、ユーザの属性情報の正当性を保証する第三者機関（CP: Claims Provider）を想定し、次の処理を実行するものである（図 2 参照）。

① CP は、ユーザの属性情報の証明書として、ユーザに検証可能クレデンシャルを発行する。
② ユーザは、サービス利用時など、必要に応じてサービス提供者に検証可能クレデンシャルを提示する。
③ サービス提供者は、検証可能クレデンシャルを検証し、ユーザの属性情報を確認するとともに、正しく CP から発行されたものであることを確認する。そのうえで、提示された情報をもとにユーザにサービスを提供する。

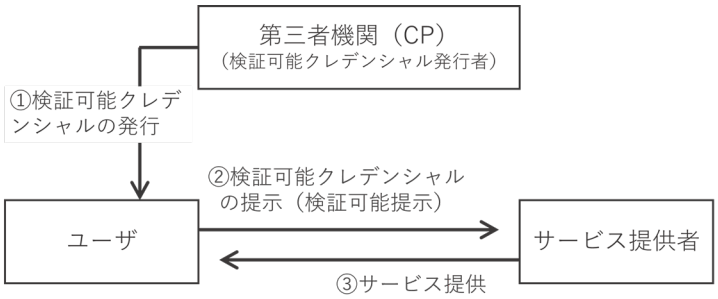

図 2：検証可能クレデンシャルを活用した属性情報提供の概要

こうした方式が実現すれば、ユーザの属性情報とユーザ認証手段が ID プロバイダに集中していた状態[12]（図 3）が図 4 のように解消される。図 3 における単独の ID プロバイダは、サービス提供者に必要なすべての属性情報を保持し、さ

[12] ID プロバイダのモデル自身においては、ID プロバイダがさらに別の保管場所にユーザの属性情報を問い合わせることもできるため、必ずしもユーザの属性情報が集中しているとは限らないが、SNS のソーシャルログインなどの実装では集中している場合が顕著である。

らにその都度ユーザの認証処理も提供する。一方、図 4 のように複数の CP が存在する方式では、ユーザの属性情報が複数の CP に分散され、それぞれのユーザにそれぞれの属性情報が検証可能クレデンシャルの形で提供されることとなる。これにより、ユーザはそれらの中から必要なものだけを自身で選んでサービス提供者に開示することができる [13]。サービス提供者は提示された属性情報が CP によって保証されていることを確認し、ユーザを認証する。このように複数の CP によって実現されるアイデンティティを、分散型デジタルアイデンティティと呼ぶ [14]。

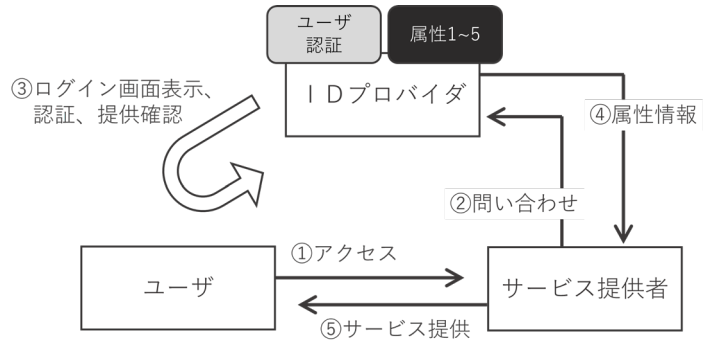

図 3：単独 ID プロバイダ方式

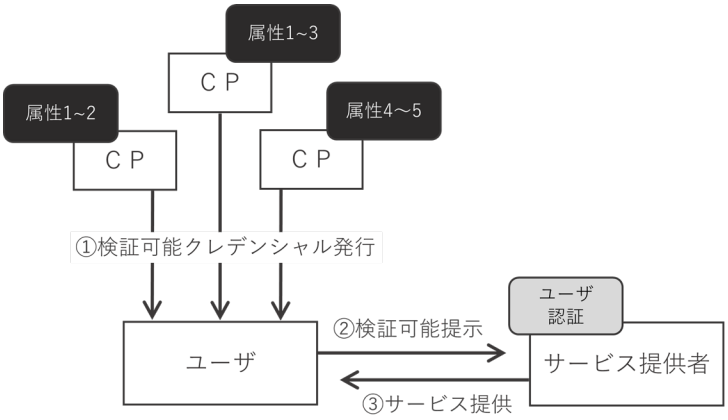

図 4：分散型デジタルアイデンティティ

---

[13] 認証処理はサービス提供者がそれぞれ行うことになるため、ユーザは利用するサービスの数だけ認証情報（例えば、パスワード）を管理する必要がある。ただし、後述の検証可能クレデンシャルを利用した方式であれば、公開鍵を利用したユーザ認証を実行可能であることから、パスワードを記憶しておくといった管理負担はなくなる。

[14] なにをもって分散型というかは諸説あるが、ここではユーザの属性（アイデンティティ）が複数の CP に分散されていることと整理した。なお、分散型識別子を使うことは必須ではない。詳しくは 5 節を参照。

3. 検証可能クレデンシャルを用いた実現方式

本節では、検証可能クレデンシャルを用いてユーザが自分の属性情報をサービス提供者に示す方式の概要を解説する。検証可能クレデンシャルを用いた提示の方法（検証可能提示）、検証可能クレデンシャルの検証に必要な CP の公開鍵の取扱い、検証可能提示において検証可能クレデンシャルに記載の一部の属性情報のみを選択して開示する方法（選択的開示）を説明する。

3.1 検証可能クレデンシャル

W3C では、検証可能クレデンシャルのデータモデルの標準を制定するとともに、そのユースケース集や実装ガイドラインを策定している。ここでは、2022 年 3 月に制定されたデータモデル v1.1[2]をベースに、検証可能クレデンシャルに含まれるデータ構成と、それを提示する手法である検証可能提示を説明する。

（1）ユーザの属性を示すクレームとそれを含む検証可能クレデンシャル

検証可能クレデンシャルは、ある主体（Subject）に関する 1 つまたは複数のクレーム（Claim）を CP が宣言することを目的とした証明書である。クレームとは、その主体がある属性項目（Property）に関して特定の属性値（Value）を持っていることを CP が表現するものである（図 5 を参照）[15]。検証可能クレデンシャルには、クレームに対する CP のデジタル署名が含まれており、当該署名を検証することによって、クレームの内容が CP によって宣言されたものであることを確認することができる。

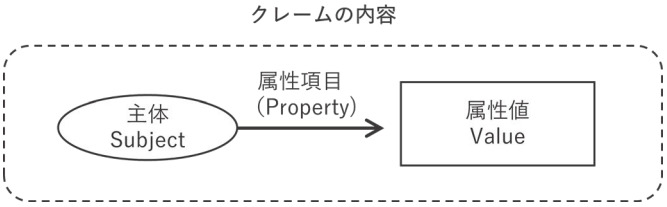

図 5：クレームの内容

例えば、図 5 のグラフによって「A 子さん（主体）」は「女性（属性値）」という性別（属性項目）であることを表現できる。そのほか、属性項目の例として、主体の所属に関する情報（部署名、役職など）や個人に関する情報（氏名、性別、住所）があるほか、行動履歴なども表現することができる。また、A 子さんの性別（属性項目）が女性（属性値）で、住所（属性項目）が C 県 D 市（属性値）であるという複数のクレームの内容はグラフを使って図 6 のように表される。

[15] ここまで単に「属性」と表現していた概念は、（属性項目、属性値）という対のデータで表現される。

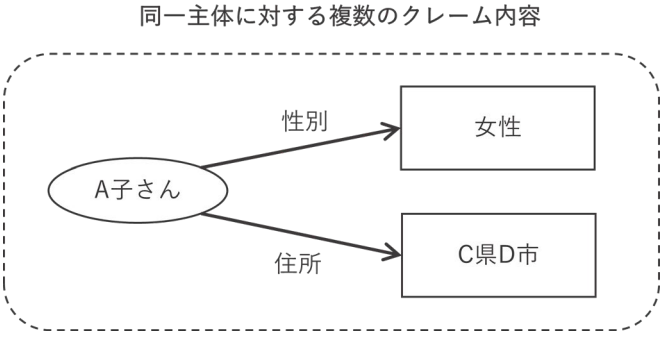

図 6：複数クレーム内容のグラフ表現

このような主体に対するクレームの内容に対して発行日などのメタデータを付与し、これらのデータ全体に対して CP のデジタル署名を付与したものが検証可能クレデンシャルである（図 7 参照）。これによって、「CP は『A 子さんは女性であり、C 県 D 市に住んでいる』ことを主張している」というクレームを表現している[16]。

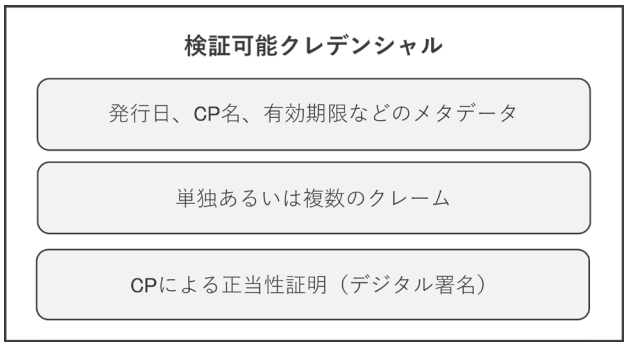

図 7：検証可能クレデンシャルの構成

（2）検証可能提示

ユーザが検証可能クレデンシャルを第三者に示す際には検証可能提示という形態で提示される。検証可能提示によって、検証可能クレデンシャルに記載されているクレームを単に第三者に提示するとともに、検証可能クレデンシャルに記載されているクレームが自分（提示者）のことであると第三者に示すことができる[17]。

---

16 ここでは、CP による主張を表現している。すなわち、本当に A 子さんの住所が C 県 D 市であるかどうかは保証していない。あくまで、CP がそのように主張している、ということである。CP の主張がどの程度信頼できるものであるかは、検証可能クレデンシャルを検証する者の CP に対する信頼（そもそも CP が正しい情報を収集しているか、正しい情報を提供しているか等）に依存する。

17 主体が自分のことであることを示すためには、あらかじめ、検証可能クレデンシャルに

例えば、A 子さんが B 商事営業部員であるというクレームをもつ検証可能クレデンシャルを考える。検証可能クレデンシャルの発行体である CP は B 商事が担っているとする。検証可能クレデンシャルには、A 子さんの属性情報（B 商事営業部員であること）と A 子さんの公開鍵 PKey が記載される（図 8 参照）。通信相手が A 子さん本人かどうかを確認するユーザ認証機能は以下の手続きによって付加することができる。

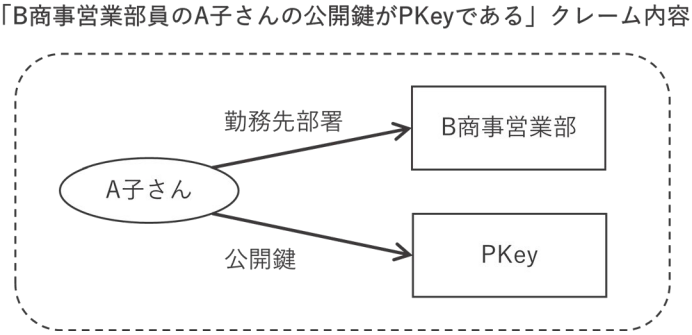

図 8：クレーム内容の例

- Step 1：サービス提供者は、アクセスしてきているユーザに対し、乱数を送る[18]。
- Step 2：ユーザは、送られてきた乱数と検証可能クレデンシャルに対して署名をし、そのデジタル署名を検証可能クレデンシャルとともに、検証可能提示のデータ形式で返信する。
- Step 3：サービス提供者は、検証可能提示を検証する。具体的には以下の 2 つを検証し、検証に成功すれば、アクセスしてきているユーザが B 商事営業部員の A 子さんであると判断する。
- Step 3-1: サービス提供者は、検証可能クレデンシャルに CP として記載されている B 商事の公開鍵を取得し、その公開鍵で検証可能クレデンシャルを検証する。これにより、検証可能クレデンシャルが B 商事によって発行されたものであることを確認することができる。
- Step 3-2：上記の検証が成功した場合、サービス提供者は、検証可能クレデンシャルに記載されている A 子さんの公開鍵 PKey を取り出し、その公開鍵で検証可能提示（のデータ）を検証する。これにより、Step 1 で送った乱数と検証可能クレデンシャルに対するデジタル署名が A 子さんによって生成されたものであることを確認することができる。

主体の公開鍵情報などの認証情報が掲載されている必要がある。このように検証可能クレデンシャルの持ち主の認証情報が付加されたものを紐づけ検証可能クレデンシャルと呼ぶ。

[18] サービス提供者から送信される乱数を利用することで、リプレイ攻撃（過去の検証可能提示を複製してなりすましを行う攻撃）を防止することができる。

検証可能提示に用いられるデータの構成を図 9 に示す。このデータは、①検証可能提示のデータであることを示すメタデータ、②提示したいクレームを含む検証可能クレデンシャル（全部または一部）、③ユーザによるデジタル署名用のデータ（Step 3-2 で使用）からなる。なお、デジタル署名の代わりにゼロ知識証明を用いることも可能であり、その場合、同証明用のデータとなる（3.4 節で後述）。

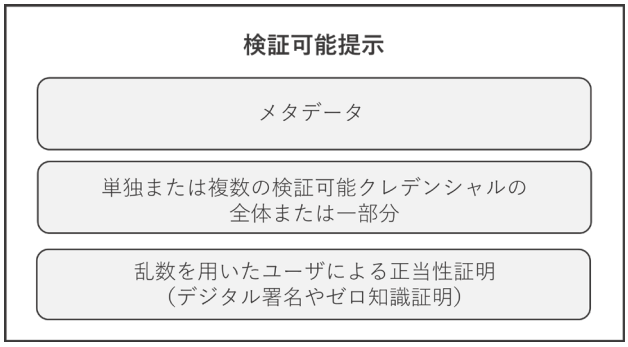

図 9：検証可能提示に用いられるデータの構成

3.2 検証可能データレジストリによる発行者の公開鍵の提供

3.1 節で説明したように、サービス提供者がユーザの検証可能クレデンシャルを検証するためには、CP の公開鍵を取得する必要がある（Step3-1 参照）。CP の公開鍵を検証可能提示に含めることも可能であるが、その公開鍵が真正な CP の公開鍵であることを確認する必要がある。W3C のモデルでは、検証可能データレジストリ（Verifiable Data Registry）の存在を想定したうえで、同レジストリ内に CP の公開鍵が格納され、サービス提供者がそこから CP の公開鍵やその失効に関する情報を取得する仕組みとしている。検証可能データレジストリは、内部のデータが当初の記録時から改変されていないかを検証できる保管庫の総称であり、実装の一例としてブロックチェーンの活用が想定されているほか、それ以外にも実装例はさまざま考えられる[19]。

3.3 さまざまなシーンで活用できる検証可能提示

検証可能提示は、単に検証可能クレデンシャルを提示するだけではなく、様々な活用シーンを想定した概念である。例えば、3.1 節で紹介したように、検証可能クレデンシャルを使ったユーザ認証を実現可能である。具体的には、ユーザが

[19] 例えば、既存の公開鍵インフラ（PKI：Public Key Infrastructure）と同様に、CP の公開鍵の正当性を保証する検証可能クレデンシャルが別の CP によって発行され、それが検証可能データレジストリに格納されるように実装することもできる。

検証可能クレデンシャルの提示に加えて、それに含まれる公開鍵に対応する秘密鍵を知っていることの証（デジタル署名）を提示することができる。また、複数の CP に発行してもらった複数の検証可能クレデンシャルをまとめて 1 つの検証可能提示とすることもできる。例えば、CP である B 商事に「B 商事営業部員」であることを示す属性値と公開鍵 PKey に対する検証可能クレデンシャルを発行してもらい、加えて、C 県 D 市役所に「D 市在住」であることを示す属性値と同一公開鍵 PKey に対する検証可能クレデンシャルを発行してもらえば、PKey という共通項を使って自分（PKey に対応する秘密鍵の保有者）が B 商事営業部員であり、かつ、D 市在住であることを示すことができる。

W3C のモデルでは、プライバシも考慮されており、検証可能クレデンシャルに記載されているクレームのうち、一部のみを提示したいという要望にも応えられるようになっている。例えば図 6 にあるように、住所と性別のクレームが一枚の検証可能クレデンシャルに記載されている場合、住所を秘匿して性別だけを開示する手法が考えられる。これは選択的開示と呼ばれる。この機能は、検証可能クレデンシャルが大きく注目される理由の 1 つとなっている。

3.4 選択的開示の方法

選択的開示とは、3.3 節で説明したとおり、1 人のユーザがもっている複数の属性のうち、特定の属性項目だけを選んで開示する方法であり、その際、開示された属性値に CP の保証がついていることを確認できる方法を指す。こうした選択的開示を実現する方法としていくつか考えられる。

1 つは、ユーザが開示したい属性項目が掲載されている検証可能クレデンシャルを都度 CP に発行してもらう方式である。この方式は、常に CP がオンラインで検証可能クレデンシャルを発行できることが条件となる。

2 つ目は、CP に事前にユーザの属性項目ごとに検証可能クレデンシャルを発行してもらい、選択した属性が記載されている検証可能クレデンシャルを提示する方式である。この方式の場合、属性の数に比例して検証可能クレデンシャルの発行数が増えることから、ユーザ側の管理負担が増大する。また、検証可能クレデンシャルを組み合わせる際に留意すべき点もある。例えば、A さんに発行した「社長」という属性項目に対する検証可能クレデンシャルと、B さんに発行した「会社名」という属性項目に対する検証可能クレデンシャルを組み合わせることで、あたかも A さんが B さんの会社の社長であるかのような偽装が想定される。こうした不正行為が成立しないような仕組みが別途必要となる[20]。

[20] そうした仕組みを採用する場合には注意が必要である。なぜならば、このような不正を

3 つめの方式として考えられているのは、1 枚の検証可能クレデンシャルに記載された複数の属性の中から、必要な属性だけを選択して開示できる機能である。これを実現するにあたっては、検証可能クレデンシャルの生成時に一定の仕掛けが必要であり、そうした仕掛けとして大きく分けて 2 つの方式が提案されている。その詳細は 4 節で述べる。

さらに、検証可能クレデンシャルに付与する署名として特別なデジタル署名アルゴリズム（効率証明付き署名と呼ばれる〈4.3 節で後述〉）を活用することによって、検証可能クレデンシャルで保証している属性値自体を開示することなく、その属性値の性質についての証明が可能である。例えば、「検証可能クレデンシャルに書かれている自分の誕生日は、20 年前の今日の日付より前の値である」ということを証明することができる。これによって、検証可能クレデンシャルに記載されている誕生日そのものを開示することなく、自身が 20 歳以上であることを示すことができる。これにはゼロ知識証明が活用される。

選択的開示やゼロ知識証明を活用することによって、ユーザは、事前に発行された検証可能クレデンシャルが他のどのような属性とともに記載されているかに関係なく、また、どのような粒度で記載されているかとも独立に、サービス提供者が求める情報のみを開示することができるようになる。また、サービス提供者は、受け取った検証可能クレデンシャル全体を確認しなくても、検証可能提示のデータに付与されたゼロ知識証明やデジタル署名などによって、確かに開示された属性は CP が保証した内容であることを確認できる。もっとも、こうした機能を実現するには、処理の対象となるデータのサイズや計算量が増加する。ゼロ知識証明を活用した選択的開示と述語証明と呼ばれる方式の詳細は 4.2～4.4 節で述べる。

3.5 従来の属性証明書との違い

検証可能クレデンシャルと類似したものに、X.509[21]ベースの公開鍵証明書と属性証明書がある。X.509 ベースの公開鍵証明書は、特定の公開鍵の属性を証明

防止するために、「社長」という属性項目が記載された検証可能クレデンシャルと「会社名」という属性項目が記載された検証可能クレデンシャルが同一の人物に発行されたことを示す必要があるが、それを示すために新たな属性を開示することになり、その結果、選択的開示が十全に行えなくなる可能性があるからである。選択的開示を十全に行うためには、同一人物であることを示す新たな属性があることを後述のゼロ知識証明技術などを使って示す方法がある。

[21] ITU-T（International Telecommunication Union Telecommunication Standardization Sector）が定める公開鍵基盤の規格であり、公開鍵証明書や属性証明書の標準フォーマットが含まれる。RECOMMENDATION ITU-T X.509 と呼ばれているが、同じ標準が ISO/IEC でも ISO/IEC 9594-8 として標準化されている。”Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks” https://www.itu.int/rec/T-REC-X.509-201910-I/en

書発行機関が証明するものであり、TLS/SSL でサーバ証明書として幅広く活用されている。同様に、属性証明書は、X.509 のフォーマットを用いて、公開鍵の所有者に関する属性を証明書発行機関が証明するものである。検証可能クレデンシャルはこれらの概念を発展させ、より柔軟に、さまざまな対象の属性値を表現するデータフォーマットを採用するとともに、柔軟な提示ができるように設計されたものである。

具体的には、検証可能クレデンシャルの発行者である CP が X.509 における証明書発行機関に相当する。検証可能クレデンシャルの対象となる主体の属性は、X.509 における公開鍵や公開鍵を保有する主体の属性に相当するが、検証可能クレデンシャルには必ずしも公開鍵を保有するとは限らない主体に関する属性も扱うことができる点が異なる。

技術面での大きな違いは、正当性証明の方法である。X.509 では、正当性証明の手法として、証明書発行機関がデジタル署名を付与することのみが想定されている。それに対して、検証可能クレデンシャルはゼロ知識証明による正当性証明も視野に入れて、データモデルが設計されている。これにより、選択開示を実現する仕組みの導入しやすさに違いが出てくる。実際、X.509 の属性証明書をサービス提供者に提示するといった方法では、サービス提供者に属性証明書の（証明書発行機関による）署名を検証させるため、ユーザは必ずしも開示が必要でない属性も提示しなければならない。その結果、選択開示を実現するには、X.509 の仕様を大幅に変更するか、発行する証明内容を制限した運用を実施する必要がある。

3.6 課題

これまでに整理したとおり、分散型デジタルアイデンティティは、プライバシを保護した形で、複数の CP が保証した個人に関するさまざまな属性を柔軟に活用できる方式であり、デジタル社会における情報の取扱いに関する信頼性向上に資すると期待される。そうした一方で、リスクや留意点も存在する。

（1）秘密鍵管理に関するリスク

まず、秘密鍵管理に関して注意が必要である。本節で紹介した方式は、個人がデジタル署名の秘密鍵をウォレットなどを用いて安全に管理できることが前提になっている。秘密鍵が安全に管理できなければ、秘密鍵の漏洩によるなりすましの脅威や、秘密鍵の紛失により本人であることを証明できなくなるといった問題が発生する。特に秘密鍵の紛失を防止するために、バックアップやリカバリー対策を検討する必要があるが、同時に、バックアップ先やリカバリーに伴う脅威についても考慮する必要がある。例えば、秘密鍵の漏洩のリカバリー策とし

て、公開鍵の更新という対策が考えられるが、最新の公開鍵をどのように周知するか、古い秘密鍵をどのように失効させるかなど、さまざまな課題が考えられる。

（2）ウォレットの動作をどこまで信頼するか

ウォレットとは、格納された秘密鍵を用いて所有者の指示に基づきデジタル署名を発行するモジュールである。個人が自身でプログラムを書いてウォレットを作製することは稀であり、一般には、第三者がプログラムを書いたウォレットのアプリケーションをダウンロードしたり、第三者である企業が提供しているウォレットサービスを利用したりすることが多い。第三者によるウォレットサービスの場合、本来であれば個人が行うべき管理を第三者であるウォレットサービス会社に委託することとなり、個人がコントローラー、ウォレットサービス会社がプロセッサーという位置づけになる。この場合、サービス提供会社との間で契約 [22]を締結することによって、ウォレットが不正な動きをした場合における法律上の責任の所在等を明らかにしておく必要がある。また、第三者によるウォレットの場合には、ウォレットがユーザの期待通りに機能しない可能性について考慮する必要がある。

（3）ID プロバイダサービスとウォレットサービスの位置づけの類似点

分散型デジタルアイデンティティ（前掲図 4）における「ユーザ」を、指示を出す「人間」とそれを受けて処理を実施する「ユーザエージェント」に分けて記述すると、ウォレットはユーザエージェントに含まれる形になり、これを考慮すると図 10 のようになる。この図 10 における人間とユーザエージェントの関係は図 3 におけるユーザと ID プロバイダの関係と同様である。機能的にも、ID プロバイダとウォレットサービス（ユーザエージェント）に大きな違いはない。したがって、ID プロバイダの利用にあたって発生した問題は、ウォレットの利用においても同様に起こり得ることから、ウォレットサービスに求められる要件を明確にしていく必要がある。

2.3 節では、ID プロバイダの利用時に存在する問題の 1 つとして、セキュリティ対策にかかるコストから、ID プロバイダサービスを提供する企業が少数にとどまることを挙げた。同様にウォレットサービスを提供する企業が少なければ、ユーザの選択肢が狭まり、少数のウォレットサービス会社が競争上優位な地位を得ることになりかねない。

---

[22] プログラムをダウンロードする場合は利用契約、処理を委託する場合は委託契約がこれにあたる。

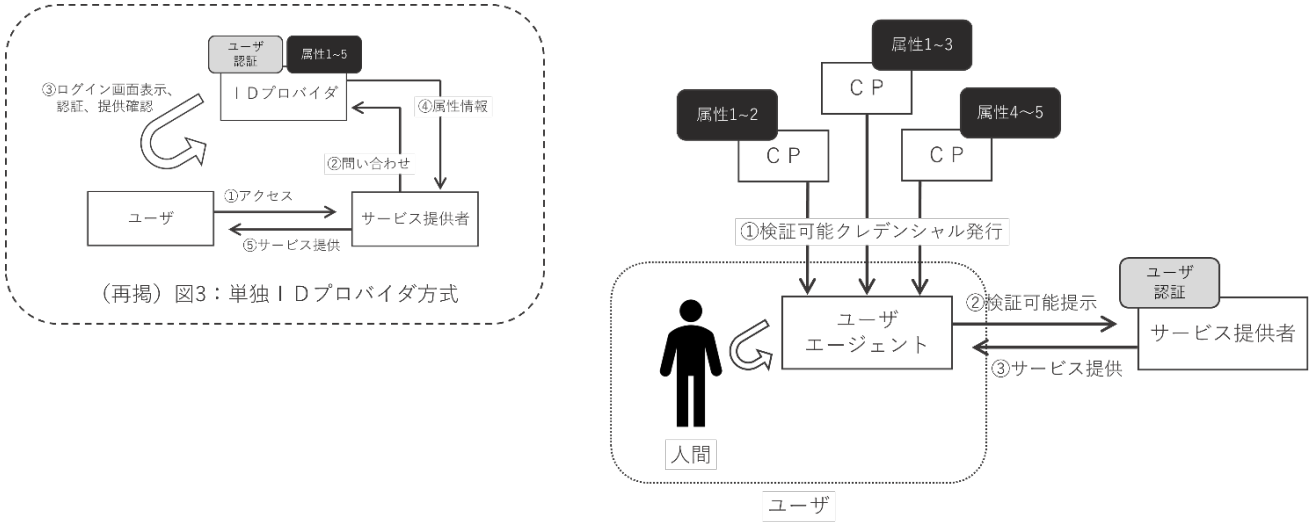

図 10：ユーザエージェントとしてのウォレットの位置づけ

分散型デジタルアイデンティティは、ユーザ情報の流通をユーザ自身がコントロールするという考えに基づくものであり、ユーザ自身が情報管理に責任を持つことから、一定の情報リテラシーが求められる。もしサービス提供者が悪意をもっていることに気づかずにユーザ情報を提供すれば、ユーザにとって意図せざる形で情報が流出してしまうことも考えられる。そのため、分散型デジタルアイデンティティの普及には、ユーザの情報リテラシー向上が重要となる[23]。

4. 選択的開示方法

本節では、3 節で紹介した選択的開示を実現するための具体的な方式を説明する。CP が発行した署名をユーザ側でどのように加工して検証可能提示として表現できるかがポイントになる。

4.1 SD-JWT 方式

ユーザの n 個の属性値が $M=(m_1, m_2,\cdots, m_n)$であるとき[24]、ユーザの検証可能クレデンシャルは、M に対する CP のデジタル署名で構成される。RSA[25]署名や ECDSA などの一般的な署名アルゴリズムを用いる場合、署名の検証には M が必要となることから、検証可能クレデンシャルには M を含む必要がある。その

[23] 信頼できる ID プロバイダやウォレットサービスを利用できる環境であれば、そうしたサービスを利用した方がよいという考え方もある。

[24] $(m_1, m_2,\cdots, m_n)$は、別途スキーマとして与えられている n 個の属性項目に対応するそれぞれの属性値である。

[25] RSA は、米国 RSA Security LLC.の登録商標である。

ため、一般的な証明方式では、M の一部を秘匿しつつ、検証可能クレデンシャルの正当性を示すことは難しい。

暗号研究では、2000 年代から墨塗り署名（Sanitizable Signatures）として、署名された文書の一部を秘匿しても残りの文書の正当性を検証できる手法が研究されてきた[4][5]。この技術を活用すれば、開示したくない部分に墨塗りを施すことにより、選択的開示が可能になる。

IETF[26]では、墨塗り署名と同様の技術を活用して、属性の選択的開示が可能な署名データフォーマット SD-JWT（Selective Disclosure JSON Web Token[27]）が策定されている[6]。この方式は、選択的開示が可能となるように、開示されうる要素それぞれに事前変換処理を施したうえで、変換した M に対して署名を付与する仕組みになっている。具体的には、ハッシュ関数 h とデータ $M=(m_1, m_2,\cdots, m_n)$に対して以下の処理を行う。

- Step 1：CP は、各 m_i に対し、$g_i=h(m_i)$を計算する。
- Step 2：CP は、$G=(g_1, g_2,\cdots, g_n)$に対して、署名 T を計算し[28]、M, G, T を記載した検証可能クレデンシャルを発行する。
- Step 3：ユーザは、m_2 だけを選択的開示する場合には、m_2, G, T を署名の検証者に開示する。
- Step 4：検証者は、$g_2=h(m_2)$であることと、T が G の正しい署名であることを確認する。

上記プロトコルによって、検証者は、m_2 を含むデータに対して CP が署名を付与したことを確認できる。しかし、その他のデータ $m_1, m_3,\cdots, m_n$ はハッシュ化されており、検証者はそれらのデータを知ることができない。

ただし、一般的なハッシュ関数では、同じ入力データからは常に同じ出力のハッシュ値が得られるため、こうした事実から入力値を特定されてしまう可能性がある。例えば、A 子さんの属性が $M_A=$（B 商事、D 市）で、J 郎さんの属性値が $M_J=$（B 商事、E 市）だった場合、「B 商事」のハッシュ値は同一であるため、B 商事という名称は秘匿されるものの、二人が同じ会社に勤めていることがわかってしまう。また、市の数も有限であるので、市の候補を順々にハッシュ化

---

[26] IETF（Internet Engineering Task Force）は、インターネット上の通信プロトコルやデータフォーマットを決めている標準化団体。

[27] JSON Web Token とは、JSON（JavaScript Object Notation）形式で記述されたデジタル署名付きのデータフォーマットであり、IETF の RFC7519 で規格化されている。

[28] 複数のメッセージ（この場合は $g_1', g_2', \cdots, g_n'$ ）に対して署名するには、複数のメッセージを連結して 1 つのメッセージに見立てて署名を計算する方法が考えられる。標準的な方式としては、IETF の RFC7515 で規定されている JWS（JSON web Signatures）などがある。

して属性値のハッシュ値と比較すれば、もとのデータを特定することも可能である。

そこで、SD-JWT の規格では、一般的なハッシュ関数ではなく、ソルト付ハッシュ関数を採用している。ソルト付きハッシュ関数とは、ハッシュ関数の入力にソルトと呼ばれる別のパラメータ x を追加した関数であり、パラメータ x を変更することで同じ入力データからでも異なるハッシュ値を出力することができる関数である。

ソルト付きハッシュ関数 h_2 を用いた SD-JWT の具体例を示す。属性値 $M=(m_1, m_2,\cdots, m_n)$ に対して以下の処理を実行する。

- Step 1：CP は、各 m_i に対し、ランダムにソルト x_i を生成して、$g_i' = h_2(x_i, m_i)$ を計算する。
- Step 2：CP は、$G'=(g_1', g_2',\cdots, g_n')$ に対して、署名 T'を計算し、M, G', T' を記載した検証可能クレデンシャルをユーザに発行する。
- Step 3：CP は、ソルト x_i の系列 $(x_1, x_2,\cdots, x_n)$ をユーザに提供する [29]。
- Step 4：ユーザは、m_2 を開示するときには、m_2, x_2, G', T'を検証者に開示する。
- Step 5：検証者は、$g_2'=h_2(x_2, m_2)$ であることと、T'が G'の正しい署名であることを確認する。

ソルト付きハッシュ関数を活用すれば、A 子さんの「B 商事」と J 郎さんの「B 商事」はそれぞれのソルトによって異なるハッシュ値になるので、同じ会社に勤めていることが推定されにくくなる。

ただし、この方式ではユーザは常に T'という署名を見せることになるため、例えば、自身の名前を秘密にしていたとしても、複数の場所で開示すれば、T'に紐づく形で開示者が同一であるという情報を与えてしまう。こうした問題を解消する方式について次節で述べる。

4.2 リンク不可能性

ISO/IEC 27551[7]では、複数の開示からユーザの同一性を特定困難とする認証方式として、リンク不可能認証（Unlinkable Authentication）を定めている。

[29] メッセージの値域が小さい場合（例えば性別の場合の 2 値など）、ソルトを公開してしまうと、G'からメッセージが類推されてしまうため、が、IETF の SD-JWT の仕様では、CP はソルトの値を本人以外に開示してはならないと記載されている。

ある CP が発行した検証可能クレデンシャルを保有するユーザ X、Y に対し、サービス提供者がそれぞれ独立に認証を行ったとする。このとき、X と Y が同一ユーザであるか、あるいは、異なるユーザであるかをサービス提供者が区別できないとき [30]、この認証方式はリンク不可能性を有しているという。図 11 は、X と Y が同じ属性値を開示しているときのリンク不可能性の状況を示している。

リンク不可能性をもたない認証方式、すなわちリンク可能な認証方式の場合、サービス提供者に対して、ユーザ名はわからないが、以前にアクセスしてきたユーザと同一であるという情報を与えてしまう。そのような情報を集約することによって、ユーザを特定できる可能性が高まることから、プライバシの観点からはリンク不可能性を有することが望ましい。例えば、4.1 節で紹介した SD-JWT 方式は、ユーザが毎回同じ署名（T'）を提示することからリンク可能であり、リンク不可能性をもたない。

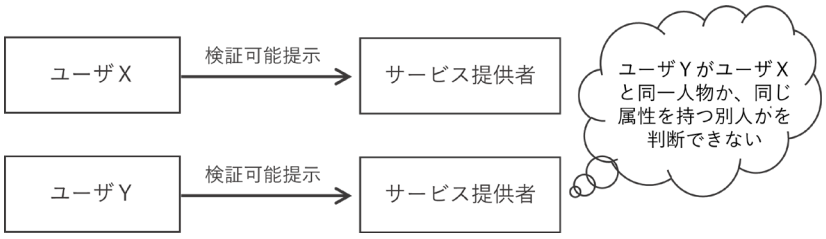

図 11：リンク不可能認証の性質

4.3 効率証明付き署名と選択的開示

リンク不可能性の要件を充足せしめる署名方式として、効率証明付き署名がある。効率証明付き署名とは、署名対象のメッセージを秘匿したまま当該メッセージの性質 [31]を効率的に証明したり、署名値を秘匿したままそれを知っていることを効率的に証明したりすることができる方式の総称である。

選択的開示に使用する効率証明付き署名は、以下の 5 つのアルゴリズムの集合として定義される。

- **鍵生成アルゴリズム**：セキュリティパラメータを入力として、公開鍵と秘密鍵（署名鍵）を出力する。初期設定時に CP が実施して自身の秘密鍵を決定する。

[30] 開示された属性値から明らかに同一人物ではないことがわかる場合を除く。

[31] 例えば、メッセージ（ビット列）の最下位ビットが 0 であるとか、数値と見立てた時にある数より大きい、といった性質を証明することができる。

- **署名生成アルゴリズム**：メッセージ $M=(m_1, m_2,\cdots, m_n)$、公開鍵、秘密鍵を入力として、署名 S を出力する。CP が検証可能クレデンシャルに付与する署名を生成する際に使用する。
- **署名検証アルゴリズム**：メッセージ M、公開鍵、署名 S を入力として、署名の検証結果を出力する。検証可能クレデンシャルの発行を受けたユーザが CP の正しい署名が付与されていることを確認する際に使用する。
- **選択的開示アルゴリズム**：メッセージ M、公開鍵、署名 S と、開示するメッセージの集合 $Mo=(m_{i1}, m_{i2},\cdots, m_{ik})\subset M$ を入力として、導出署名 So を出力する。導出署名 So は、ユーザが自身の属性が Mo であることを選択開示する際に使用するデータであり、ユーザによって生成される。
- **選択的開示検証アルゴリズム**：開示するメッセージの集合 Mo、公開鍵、導出署名 So を入力として、導出署名の検証結果を出力する。サービス提供者は、導出署名 So の検証によってユーザの属性が Mo であることを確認することができる。

このようなアルゴリズムを効率的に構成できる効率証明付き署名として、CL 署名[8]や BBS＋署名[9, 10]がある。BBS＋署名については補論を参考にされたい。

4.4 述語証明

4.1 節や 4.3 節の選択的開示プロトコルは、署名されたメッセージ $M=(m_1, m_2, \cdots, m_n)$のうちの一部である $Mo=(m_{i1}, m_{i2},\cdots, m_{ik})$を開示して、それに CP の署名がついていることを証明するものである。この方式ではそれぞれのメッセージ $m_1, m_2,\cdots, m_n$ に対しては、開示するかしないかの 2 択しかない。

これに対し、述語証明は、ある m_i が（m_i の値を開示することなく）どのような性質を満たすかを証明することができる。例えば、m_i の値が $a<m_i<b$ を満たす（範囲証明）、あるいは、$m_i=m_j$ を満たす（同値証明）といった関係性を示すものである。例えば、これにより、検証可能クレデンシャルに記載の具体的な誕生日を示すことなく、20 歳以上であることを示すことができる。こうした関係性を示す方法として、ゼロ知識証明が多用される [32]。本稿では詳細に触れないが、上述の効率証明付き署名に対して範囲証明や同値証明をはじめとする述語証明を効率よく実現できることが知られている[8][9]。

---

[32] $a<m_i$ は　m_i-a が正であること、$m_i=m_j$ は　m_i-m_j がゼロであることを（個々の値に関してもらす知識をゼロにして）証明する手法である。

5. 分散型識別子とブロックチェーンについて

本稿では、ユーザが検証可能クレデンシャルを用いて本人がもつ属性（デジタルアイデンティティ）をサービス提供者に提示する方法を紹介した。検証可能クレデンシャルを用いる場合、ユーザ自身が選択し管理している識別子である分散型識別子 [33]の利用が必須であると誤解されている向きもある。確かに、[2]におけるモデルは、検証可能クレデンシャルにおいて発行者を特定したりユーザを特定したりする際に分散型識別子を使うことができる仕様となっている。しかし、分散型識別子は、分散型デジタルアイデンティティの実現に必須ではない。もっとも、両者は関係の深い概念であることから、以下で紹介する。

5.1 DID メソッド

プラットフォーマーなどが個人のデジタルアイデンティティを管理する場合、ユーザは、自分に「付与された」属性のすべてを確認することができないほか、サービス提供者を含めて誰にどの属性を提供するかを自由に選択することができない。また、プラットフォーマーは、任意のユーザのアカウントをいつでも停止できてしまう。こうした現状に対して、ユーザが自分のデータをよりコントロールできるようにするにはどうしたらよいか、という問いから発生した概念が分散型識別子（DID：Decentralized Identifier）である。これは、ユーザ個人が自分の識別子を自身で作成し、自分のデータを自身で管理するという考え方である。一方で、ユーザが自由に識別子を選択できるようになると、他人が選んだものと重複する可能性があり、実際に重複した場合には、識別子として機能しないといった問題が発生する。

そこで、重複が生じないようにユーザが識別子を決めることができるように、ブロックチェーン等を使いつつ様々な分散型識別子の管理方式（DID メソッド）が検討されてきた。DID メソッドは、分散型識別子の作成、分散型識別子に関する情報（DID ドキュメント [34]）の参照方法、DID ドキュメントの更新・停止方法といった管理一般のメカニズムである。多数の DID メソッドが提案されているため、W3C では、特定の DID メソッドを標準化するのではなく、DID メソッドにおける分散型識別子を統一的に取り扱うことができる分散型識別子の

---

[33] コンピュータ管理者やサービス提供者が割り当てて管理する識別子は中央集権的に決められた識別子であるのに対し、ユーザ自身が選んで自身で管理する識別子は非中央集権的であるという意味で、分散型識別子と呼ばれる。なお、必ずしもユーザ自身が選んだ文字列が識別子になるわけではなく、ユーザが特定のアルゴリズムを実行した結果として出力されたデータを識別子として利用することを含む。

[34] DID ドキュメントには、分散型識別子が指し示す個人を説明する情報や、個人の公開鍵が記載されている。

表記方法や操作インターフェイスが主に制定された[3][35]。その表記例は上記図12のとおりである。

まず、規定値のdidという文字列があり、コロンの次にDIDメソッドを特定する識別子（図12の「example」）が配置される。さらに、コロンの次に、当該DIDメソッドにおいて割り振られた識別子（図12の「123456789abcdefghi」）が配置される。本稿執筆時点（2023年4月末）では、100を超えるDIDメソッドがW3Cにおいて管理されている[36]。

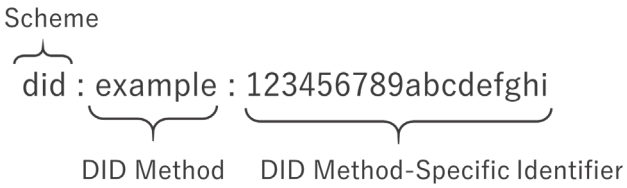

図 12 : DIDの表記例[3]

5.2 検証可能データレジストリ

分散型デジタルアイデンティティの仕組みでは、検証可能クレデンシャルを検証する際に、検証可能データレジストリという信頼できる情報の格納庫が想定されている。DIDメソッドによっては、ユーザのDIDドキュメントをブロックチェーンに搭載するように表現されることがあるが、検証可能データレジストリをブロックチェーンなどの分散台帳とする必要は必ずしもない[2]。

ブロックチェーンを検証可能データレジストリとして使用する場合には、個人情報に該当する可能性があるDIDドキュメントがブロックチェーン上で公開されることになる。そのため、修正不可能な個人情報（Personally Identifiable Information）を公開することを禁じている欧州の一般データ保護規則（GDPR : General Data Protection Regulation）との関係には注意が必要である。ただし、DIDドキュメントに含まれる情報が、個人ではなく組織を特定する公開鍵のみの場合、ブロックチェーンに搭載してもGDPRとの関係で問題は生じないとされている[19,20]。

---

[35] 具体的には、分散型識別子の表記方法、共通データモデル、基本要件、シリアライズ方法、分散型識別子にかかる操作等について規定している。

[36] DIDメソッドは、DID Specification RegistriesというW3CのDecentralized Identifier Working Groupのグループノートにまとめられている。

5.3 分散型識別子の課題

ユーザ個人が自分の識別子を自身で作成できる分散型識別子は、個人の自由度が向上する一方で、課題も残る。個人がだれでも自分の識別子を決められることから、例えば、個人が有名なブランド名を想起させるような識別子を選ぶことも可能になる。あるいは、他人が日常的につかっている識別子を自分のものとして横取りしてしまうことができたり、1 人で複数の識別子を生成することもできる [37]。

したがって、第三者によって分散型識別子が示すクレームが宣言されて初めて、識別子を受け取った人（検証者）は自分とその第三者の関係性からそのユーザの分散型識別子をどのように扱うか決めることができる。こうした第三者による宣言に検証可能クレデンシャルが活用される。

6. まとめ

本稿では、分散型デジタルアイデンティティの概念、標準化が進められている検証可能クレデンシャルの仕組みや関連する技術について紹介した。これまで、ユーザのアイデンティティ（属性）の管理は、コンピュータ管理者やサービス提供者の視点で語られることが多かった。最近では、プラットフォーマーがユーザの属性情報を集中的に管理するようになっているほか、ユーザによるサービスの利用可能性やプライバシ保護の度合いを左右するようにもなっている。分散型デジタルアイデンティティは、ユーザの意思やプライバシをより尊重するアイデンティティ管理を目指し、従来のアイデンティティ管理の構造を見直すものである。検証可能クレデンシャルのほか、選択的開示や、それを実現するための効率証明付き署名やゼロ知識証明といった技術も実装されつつあり、分散型デジタルアイデンティティが今後広く活用される素地が整いつつあるといえよう。

ただし、検証可能クレデンシャルを大量に収集すれば特定のユーザに関する正確な個人情報を大量に集めることが可能となり、プライバシが侵害される恐れもある。プライバシ保護の観点からは、特に選択的開示への対応が必須であると考えられ、適切な運用体制の整備が求められる。

検証可能クレデンシャルを利用した分散型アイデンティティのシステムが普及し、安定的に稼働するためには、エコシステムに関与するステークホルダーが複数存在し、価値が循環され、それぞれにメリットがあるような仕組みとする必

[37] ユーザが複数の識別子を生成できることは、プライバシの観点から奨励されることでもあるが、それが不正に使われないようにすることが重要である。

要がある。また、CP はサービス提供者から信頼される組織でなければならない。組織が社会的信頼を得るには、確実な運営とセキュリティ対策が必要であり、相応のコストが必要という点では、CP も ID プロバイダと同じ課題を抱えているといえよう。

検証可能クレデンシャルに関連する技術は、多くのアプリケーションやサービスで使われてこそ、利便性が高まり、その価値も高まる。今後、ユーザがウォレットで秘密鍵を安全に管理する手法が確立されること、また、それと合わせて検証可能クレデンシャルの技術が幅広く採用されて、個人中心の安全・安心な社会の実現に資する取組みが促進されることを願ってやまない。

参考文献

[1] International Organization for Standardization (ISO), and International Electrotechnical Commission (IEC), *ISO/IEC 24760-1: 2019 IT Security and Privacy -- A Framework for Identity Management -- Part 1: Terminology and Concepts,* ISO and IEC, 2019.

[2] World Wide Web Consortium (W3C), *Verifiable Credentials Data Model v1.1 -- W3C Recommendation 03 March 2022,* W3C, 2022 (available at https://www.w3.org/TR/vc-data-model-1.1/).

[3] World Wide Web Consortium (W3C), *Decentralized Identifiers (DIDs) v1.0 -- Core Architecture, Data Model, and Representations -- W3C Recommendation 19 July 2022*, W3C, 2022 (available at https://www.w3.org /TR/did-core/).

[4] Ateniese, Giuseppe, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik, "Sanitizable Signatures," Proceedings of European Symposium on Research in Computer Security (ESORICS) 2005, Lecture Notes in Computer Science, 3679, Springer, 2005, pp. 159-177.

[5] Miyazaki, Kunihiko, Mitsuru Iwamura, Tsutomu Matsumoto, Ryoichi Sasaki, Hiroshi Yoshiura, Satoru Tezuka, and Hideki Imai, "Digitally Signed Document Sanitizing Scheme with Disclosure Condition Control," The Institute of Electronics, Information and Communication Engineers (IEICE) Trans. Fundamentals, Vol.E88-A, No.1, 2005, pp. 239–246.

[6] Internet Engineering Task Force (IETF), *Selective Disclosure JWT (SD-JWT) draft-ietf-oauth-selective-disclosure-jwt-01*, IETF, 2022 (available at https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/01/).

[7] International Organization for Standardization (ISO), and International Electrotechnical Commission (IEC), *ISO/IEC 27551:2021 — Information Security, Cybersecurity and Privacy Protection — Requirements for Attribute-based Unlinkable Entity Authentication*, ISO and IEC, 2021.

[8] Camenisch, Jan, and Anna Lysyanskaya, "A Signature Scheme with

Efficient Protocols," Proceedings of Security in Communication Networks (SCN) 2002, Lecture Notes in Computer Science, 2576, Springer, 2002, pp. 268-289.

[9] Au, Man Ho, Willy Susilo, and Yi Mu, "Constant-Size Dynamic k-TAA," Proceedings of Security and Cryptography for Networks (SCN) 2006, Lecture Notes in Computer Science, 4116, Springer, 2006, pp. 111-125.

[10] Camenisch, Jan, Manu Drijvers, and Anja Lehmann, "Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited," Proceedings of Trust and Trustworthy Computing (TRUST) 2016, Lecture Notes in Computer Science, 9824, Springer, 2016, pp. 1-20.

[11] Yamamoto, Dan, Yuji Suga, and Kazue Sako, "Formalising Linked-Data based Verifiable Credentials for Selective Disclosure," Proceedings of 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), EuroS&PW, 2022, pp. 52-65.

[12] W3C Credentials Community Group, *Data Integrity 1.0 -- Final Community Group Report 22 July 2022*, W3C, 2022 (available at https://www.w3.org/community/reports/credentials/CG-FINAL-data-integrity-20220722/).

[13] W3C Credentials Community Group, *Verifiable Credentials Implementation Guidelines 1.0 -- Implementation Guidance for Verifiable Credentials -- W3C Working Group Note 24 September 2019*, W3C, 2019 (available at https://www.w3.org/TR/vc-imp-guide/).

[14] W3C Verifiable Claims Working Group, *W3C Verifiable Credentials Use Cases -- W3C Working Group Note 24 September 2019*, W3C, 2019 (available at https://www.w3.org/TR/vc-use-cases/).

[15] OpenID Foundation, *OpenID for Verifiable Credentials -- A Shift in the Trust Model Brought by Verifiable Credentials, OpenID Foundation*, 2022 (available at https://openid.net/wordpress-content/uploads/2022/06/OIDF-Whitepaper_OpenID-for-Verifiable-Credentials-V2_2022-06-23.pdf).

[16] W3C Verifiable Credentials Working Group, *Securing Verifiable Credentials using JSON Web Tokens -- W3C First Public Working Draft 26 October 2022*, W3C, 2022 (available at https://w3c.github.io/vc-jwt/).

[17] Microsoft, "Network Domains and Domain Controllers," Microsoft, 2021 (available at https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-authsod/c4012a57-16a9-42eb-8f64-aa9e04698dca).

[18] Boneh, Dan and Victor Shoup, "A Graduate Course in Applied Cryptography," Dan Boneh and Victor Shoup Part III, Section 20 Proving Properties in Zero Knowledge, 2023 (available at http://toc.cryptobook.us/book.pdf).

[19] Sovrin Foundation, "Innovation Meets Compliance —Data Privacy Regulation and Distributed Ledger Technology" 2020 (available at https://sovrin.org/wp-content/uploads/GDPR-Paper_V1.pdf)

[20] An official website of the European Union "Do the data protection rules apply to data about a company?" (available at https://commission.europa.eu/law/law-topic/data-protection/reform/rules-business-and-organisations/application-regulation/do-data-protection-rules-apply-data-about-company_en)

補論：BBS＋署名

4.3 節で紹介した BBS＋署名は、楕円曲線上のペアリング演算を使用した方式であり、複数のデータを並べたリストをメッセージとして署名生成が可能という特徴をもつ。例えば、BBS＋署名では、リスト化されたメッセージ M=(m_1, m_2, …, m_n)に署名を付与することができる。また、選択開示の際には、「開示するデータの集合以外に、開示しないデータの集合についての知識を自身はもっていることを証明しつつ、メッセージの全体集合（M）に対するデジタル署名の存在についてのゼロ知識証明を行うことができる。

BBS＋署名の 5 つのアルゴリズムの概要は下記である。

- **鍵生成アルゴリズム**：セキュリティパラメータ（p）に対して、ペアリング演算）[38]が可能な楕円曲線 E と生成元（g_1, g_2）、秘密鍵 sk と公開鍵 w と、n+1 個の定数（h_0, h_1,…, h_n）を出力する。ここで、秘密鍵 sk は生成元 g_2 に関する公開鍵 w の離散対数である（すなわち、$w = g_2^{sk}$ である）。
- **署名生成アルゴリズム**：メッセージ M=(m_1, m_2,…, m_n)と公開鍵 w、秘密鍵 sk と公開定数(p, g_1, g_2, h_0, h_1,…, h_n)を入力として、乱数 t, s を選び、下記を計算する。

$$A=\left(g_1 h_0^s h_1^{m_1} \cdots h_n^{m_n}\right)^{\frac{1}{sk+t}}$$

  署名 S として（A,t,s）を出力する。
- **署名検証アルゴリズム**：メッセージ M と公開鍵 w、署名（A,t,s）と公開定数(p, g_1, g_2, h_0, h_1,…, h_n)を入力とし、ペアリング演算を e として下記が成り立つことを確認する。

$$e(A, wg_2^t) = e\left(g_1 h_0^s h_1^{m_1} \cdots h_n^{m_n}, g_2\right)$$

- **選択的開示アルゴリズム**：メッセージ M と公開鍵 w、署名（A,t,s）、開示するメッセージの集合 Mo=(m_{i1}, m_{i2},…, m_{ik})、乱数（r_1, r_2）と公開定数(p, g_1, g_2, h_0, h_1,…, h_n)を入力として下記を計算する。

$$b = g_1 h_0^s h_1^{m_1} \cdots h_n^{m_n}$$
$$A' = A^{r_1}$$
$$\bar{A} = A'^{-t} b^{r_1}$$

---

[38] ある関数 e が、e(xy, z) = e(x, z)･e(y, z)かつ e(x, yz) = e(x, y)･e(x, z)を満たすとき、e はペアリング演算であるという。このとき、$e(x^a, y^b) = e(x, y)^{ab}$ を満たす。

$$d = b^{r_1} h_0^{-r_2}$$

$$r_3 = \frac{1}{r_1},\ s' = s - r_2 r_3$$

これを用いて、$\mathcal{D}$=(i_1, i_2,…, i_k) として、

「$\left(\frac{\bar{A}}{d} = A'^{-t} h_0^{r_2}\right) \wedge \left(g_1 \prod_{i \in \mathcal{D}} h_i^{m_i} = d^{r_3} h_0^{-s'} \prod_{i \notin \mathcal{D}} h_i^{-m_i}\right)$　となる

$(m_i)_{i \notin \mathcal{D}}, t, r_2, r_3, s'$ を知っている」ことのゼロ知識証明 π を計算する [39]。

この結果、算出されたデータ

$$(A', \bar{A}, d, \pi)$$

を導出署名 So として出力する。

- **選択的開示検証アルゴリズム**：Mo=(m_{i1}, m_{i2},…, m_{ik})と公開鍵、導出署名 So=$(A', \bar{A}, d, \pi)$と公開定数(p, g_1, g_2, h_0, h_1,…, h_n)を入力として、

  $$e(A', w) = e(\bar{A}, g_2)$$

  が成り立つことを確認するほか、A'が 1 でないことと、上記のゼロ知識証明(π)が正しいこともそれぞれ確認する。

選択的開示プロトコルの考え方について述べる。全開示であれば、署名検証アルゴリズムのように、M と A と s と t を開示し、それが署名検証式を満たすことを確認することによって、CP の署名がついていることを確認できる。選択的開示では M が部分的にしか開示されず、加えて紐づけ不可能な証明をしたいので、M も$\bar{A}$も s も t もそのまま開示することはできない。

そこで、乱数 r_1 を用いて A を A'に変換し、これを公開することにする。さらに、乱数 r_1,r_2 を用いて、s と t のかわりに$\bar{A}$と d を公開する。また、M を選択開示した部分と選択開示していない部分に分割する。これらの変換が正しく行われていることを保証しているのがゼロ知識証明のπであり、公開した A',$\bar{A}$, d を用いて、開示されていない情報を知っていることを証明している。検証者はこの証明に対応する検証式が成り立つことと、変換された A'がペアリング関数を通じて適切な性質を持っていることを検証する。この 2 点を検証することで証明者が$(m_i)_{i \notin \mathcal{D}}$である残りの開示されない属性値の集合と発行者の署名(A,t,s)を知っていて、かつ、その署名は正しい M の署名になっていることを確認することができる。すなわち、開示された Mo にも発行者の署名がついていたことを示す証明になっている。

---

[39] この証明は Generic Protocols for non-linear relations[18]を活用して構成できるが、ここでは詳細を省く。