情報処理学会論文誌 Vol. 51 No. 2 604–612 (Feb. 2010)

一意なアクセスと属性間関係性の検証可能な属性情報分散管理方式

柿崎 淑郎[†1] 吉田 慶章[†1,*1] 辻 秀一[†2]

属性情報は個人などの主体が持つ属性，権限，職責，資格，地位などであり，電子商取引や Web サービスなどにおける認可の根拠として用いられる．属性情報は個人情報の一部として厳密に扱われる場合も多く，情報流出時の影響範囲を考慮した管理が必要である．分散管理は集中管理に比べ，情報流出時の影響を抑えることはできるが，管理される情報が複数箇所に分散されるため，情報の利便性は低下する．本提案方式は，分散管理された属性情報を一意に指し示す方法として URI，属性情報どうしの関係性を示す方法として RDF を用いることで，属性情報の有効利用と有効性検証を可能とする属性情報の分散管理方式である．分散管理されている属性情報へのアクセス方法を ID プロバイダが集中管理し，アクセスをリダイレクトすることで，属性情報の利用者に対して集中管理と同じ利便性を保ちつつ，分散管理と同じ安全性を実現とする．また，属性情報は分散管理されるため，情報流出が発生した場合の影響を局所化できるとともに，RDF によって分散管理されている属性情報どうしの関係を示すことで，属性情報の検証者はより厳密な属性情報の検証が可能となる．

A Decentralized Management for Attribute Information with Unique Access and Verification of Relationship between Attributes

Yoshio Kakizaki,[†1] Yoshiaki Yoshida[†1,*1] and Hidekazu Tsuji[†2]

The attribute information of a subject may be an authority, a qualification, a position, etc., and it is used to authorize the e-commerce and the Web service, etc. This attribute information is often treated as part of an individual's identity information, so we should consider about information leakage. In this paper, we propose a decentralized management for attribute information which is able to verify effectiveness using URI and RDF. Our method makes it possible that an identity provider manages URI for authority which manages attribute information, and redirects access to its authority. And it is the same convenience centralized management, and the same security as decentralized management.

1. はじめに

属性情報は個人などの主体が持つ属性，権限，職責，資格，地位などであり，電子商取引や Web サービスなどにおける認可の根拠として用いられる．個人が持ちうる属性情報としては，氏名，性別，生年月日，住所，職業，所属など様々なものがある[1)]．また，一部の属性情報は個人と密接に関わり合っていることもあり，プライバシの問題が発生することがある．そのため，このような属性情報は個人情報の一部として厳密に扱われる場合も多い．一方で，電子商取引や Web サービスなどにおける認可の根拠として，入会審査や登録作業などで属性情報を要求されることが多く，広範な場面で属性情報が利用されている現状がある．特に近年においては，個々人に適応したサービスを提供する Web サービスが増えており，属性情報の活用が活発に行われている．

属性情報を管理する場合，大きく分けて，集中管理と分散管理の 2 つに分けられる．集中管理は 1 つの機関が多くの属性情報を集中的に管理する方式であるため，利便性や再利用性は高い反面，膨大な情報を管理するに足る信頼と安全性の確保が必要である．たとえば，情報流出が発生した場合，集中管理であるがゆえに，多数の情報が流出する問題がある．対して，分散管理は複数の機関で分担して属性情報を管理する方式であるため，1 つの機関に属性情報が集中することはない．また各機関が互いに独立であれば，情報流出時の影響範囲は限定的である．

しかしながら，属性情報が複数の異なる機関によって分散管理される場合，属性情報が複数箇所に分散するため，どの機関がどの属性情報を管理しているかを把握しておく必要がある．属性情報は他の属性情報と依存・派生関係を持つことがあるため，関係性のある属性情報を検証するためには，各属性情報をどの機関が管理しているかを調べ，複数箇所に分散している属性情報のすべてを取得する必要があり，属性情報の取得・検証時に問題がある．

†1 東京理科大学
Tokyo University of Science
†2 東海大学
Tokai University
*1 現在，日本アイ・ビー・エム株式会社
Presently with IBM Japan, Ltd.

この問題に対して，属性情報の分散管理として URI を用いる方法を検討してきた[2]．属性情報に URI を割り当てることで，一意にアクセス可能とし，分散管理されている属性情報の有効利用を実現している．また，関係性のある属性情報の関係確認と有効性検証を行うために，属性情報の厳密な検証方法について検討を行った[3]．

本論文では，分散管理された属性情報を一意に指し示す方法として URI，属性情報どうしの関係性を示す方法として RDF を用いることで，属性情報の有効利用と有効性検証を可能とする属性情報の分散管理方式を提案する．本提案方式は，分散管理されている属性情報へのアクセス方法として，ID プロバイダを介してリダイレクトすることで，属性情報の利用者に対して集中管理と同じ利便性を保ちつつ，分散管理と同じ安全性を実現する．また，複数の異なる属性認証機関によって管理されている属性情報どうしの関係を RDF によって示すことで，属性情報の検証者はその属性情報が発行された根拠を確認し，より厳密な属性情報の検証が可能となる．

2. 関 連 研 究

属性情報の集中管理に関する研究として，千葉らによる「属性情報プロバイダ」の提案がある[4]．属性情報プロバイダは個人属性を安全に交換，管理する情報化基盤であり，属性情報プロバイダに属性情報を登録することで，属性情報の実用性や信頼性を向上させている．しかし，属性情報を集中管理することで，情報流出時には影響範囲が拡大するため，属性プロバイダの管理責任や安全性の保証など，運用面における問題が残る．そのため，集中管理を行う属性プロバイダは強固に安全かつ高負荷に耐える運用が求められる．

対して分散管理を行う場合，情報流出時の影響範囲を限定的にとどめ，負荷分散による信頼性の向上が図れる一方で，どの情報をどの機関が管理しているかを知る必要があり，利便性に課題がある．P2P の分野では，この問題を解決するために，分散ハッシュテーブルを用いて，情報を効率的に管理する技術が用いられている．しかしながら，特に ID 管理技術では，扱う情報が個人情報やプライバシ情報などであるために，情報流出のリスクや管理している実体がどこか分からないなどの心理的な不安が少なくない．そのため，個人情報や属性情報などは，分散ハッシュテーブルなどによる効率的な分散管理を行うことは困難である．

本提案方式では，属性情報はその属性情報を証明する機関に各々分散して管理させ，その属性情報がどこにあるかを集中的に管理する方式を用いることで，集中管理と分散管理の問題を解決する試みを行っている．

セマンティック Web の分野では，RDF（Resource Description Framework）[5] による情報のネットワークである Linked Data に注目が集まっている．Berners-Lee は Linked Data を次のように定義している[6]．1. URI による事物への名前付け，2. HTTP URI による名前参照，3. URI を参照したときの有益情報の提供，4. 外部へのリンクを含むこと．情報に URI が与えられると，その情報は一意に特定可能となる．特に，Web 上で利用される情報に URI を与えることで，コンピュータ可読であり処理可能な情報のネットワークが形成され，Web 上の膨大な情報が利用しやすくなる．

近年では，Web サービスを中心に急速な普及を見せている URI を用いた分散型 ID 認証システムとして OpenID がある[7]．OpenID では自分のブログや Web サイトの URL を ID として利用することができ，その ID を用いて OpenID 対応の複数のサービスをシングルサインオンで利用できる．また，OpenID の拡張使用であり，属性交換を可能とする Attribute Exchange や，アクセス権限を付与する認可プロトコルの OAuth など，Web 技術との親和性が高い URI ベースの ID 管理技術が注目されている．

3. 属性情報の分散管理方式

3.1 概　　要

本提案方式は，分散管理された属性情報を一意に指し示す方法として URI を利用し，属性情報どうしの関係性を示す方法として RDF を利用することで，属性情報の分散管理において属性情報の有効利用と有効性検証を提供する．

属性情報を指し示す URI は ID プロバイダと属性認証機関の双方に与えられ，ID プロバイダに割り当てられた URI から，属性認証機関に割り当てられた URI へとリダイレクトさせる．これによって，属性情報を管理する属性認証機関が変更になったとしても，ID プロバイダからのリダイレクト先が変わるだけで，ID プロバイダに割り当てられた URI が変わらない限り，属性情報を要求する場合の URI は変わらない．

また，属性情報どうしの関係性を示すために RDF を用いる．RDF には対象の属性情報と関係のある属性情報と対象の属性情報を証明する属性証明書に関する情報が書かれている．これによって，属性情報の検証者は，属性情報が発行された根拠を確認することができ，より厳密に属性情報を検証することができる．

本提案方式の利点を以下にあげる．

- URI によって一意にアクセスが可能．
- 属性情報は複数の属性認証機関によって分散管理が可能．
- 複数の属性認証機関によって管理されている属性情報の関係を検証可能．

3.2 準　　備

3.2.1 プレイヤ

本論文で用いる 3 つのプレイヤを以下のように定義する．

IdP（Identity Provider）は属性情報を実際に管理している *AP* への情報を管理するプレイヤであり，実際の属性情報は管理しない信頼できる第三者機関である．*IdP* は X.509 公開鍵証明書[8] を発行する認証局または，それに類する ID プロバイダであり，公知の比較的大規模な機関である．

AP（Attribute Provider）は実際の属性情報を管理するプレイヤであり，X.509 属性証明書[9] を発行する属性認証機関とする．*AP* は信頼できる第三者機関である．

Client は属性情報の取得を要求するプレイヤであり，関係性のある属性情報を発行しようとする *AP* や属性情報を利用するサービス提供者である．

3.2.2 URI

URI（Unique Resource Identifier）は決められた書式によって，リソースを一意に指し示す識別子である[10]．すべての属性情報に URI を割り当てることで，それらは一意にアクセス可能となる．本提案方式では URI を以下のように設定する．

`scheme://Authority/UniqueID/Attribute`

ここで，`scheme` は URI で示された情報を取得する手段で，本提案方式では `https` を用いる．`Authority` は *IdP* または *AP* であり，`UniqueID` は `Authority` が管理している利用者を識別するユニークな番号である．`Attribute` は `Authority` が管理している属性情報である．

本提案方式で用いる URI は上記の書式を用いるが，一般的には，恒久的かつ固有であれば，異なる書式であっても問題はない．

3.2.3 要求と返答

Client が以下の属性情報を取得するための要求を行う．

`https://idp1/12345/address`

これをリクエスト URL と呼ぶ．認証とリダイレクトが行われ，最終的な返答として，要求した属性情報に関する情報が RDF として返される．また，この属性情報の属性証明書を取得する場合は，以下の要求を行う．

`https://idp1/12345/address/cert`

これらの返答に関する情報は 3.4 節で説明する．

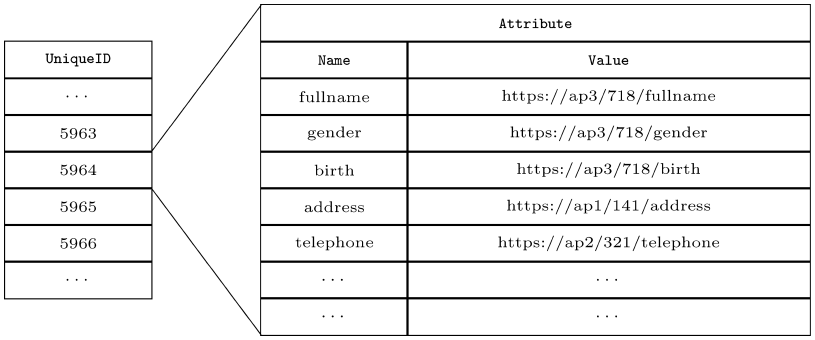

図 1　`idp1` の属性情報ディレクトリ例
Fig. 1　An example of an attribute information directory of `idp1`.

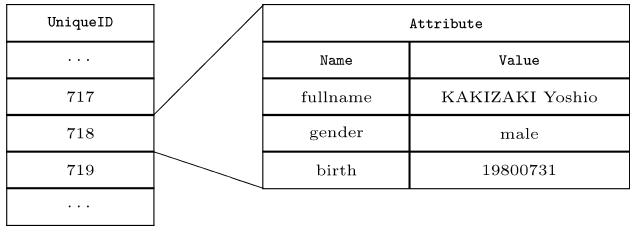

図 2　`ap3` の属性情報ディレクトリ例
Fig. 2　An example of an attribute information directory of `ap3`.

3.3 属性情報ディレクトリ

属性情報ディレクトリには属性に関する情報が格納されるが，*IdP* と *AP* では異なる情報が格納される．*IdP* の属性情報ディレクトリ例を**図 1** に，*AP* の属性情報ディレクトリ例を**図 2** に示す．

図 1 は，*IdP* である `idp1` における `UniqueID` が 5964 の利用者の属性情報がどの *AP* によって管理されているかを示している．この例では `Attribute` である fullname，gender，birth は同じ *AP* である `ap3` に管理されており，address と telephone はそれぞれ異なる *AP* に管理されている．*IdP* の属性情報ディレクトリには fullname，gender，birth などの基本的な属性情報[11] を格納するための領域があらかじめ準備してあり，その他の属性情報を格納する場合は，領域を追加して属性情報を格納する．この属性情報ディレクトリには，

```
<rdf:Description rdf:about="定義する属性の URI">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:value>属性値</rdf:value>
  <dcq:requires rdf:resource="関係のある属性の URI"/>
</rdf:Description>
```

図 3　属性情報の関係を示すクラスの表現
Fig. 3　An expression of the class for relation between attributes.

利用者の申告によって，実際の属性情報を管理している ap3 へのリダイレクト URL が格納される．そのため，利用者が登録していない属性情報については，空欄とする．

図 2 は，*AP* である ap3 における UniqueID が 718 の属性情報を示している．図 1 で説明したように，fullname，gender，birth の属性情報を管理しているが，それ以外の address や telephone などの属性情報は管理していない．*AP* の属性情報ディレクトリは *IdP* の属性情報ディレクトリとは異なり，管理している利用者の属性情報のみを格納し，それ以外の属性情報を格納する領域は準備しない．

属性情報ディレクトリに属性情報を格納する場合について説明する．*IdP* の属性情報ディレクトリは，利用者が属性情報の格納先である *AP* へのリダイレクト URL を登録しない限り空欄である．そのため，利用者は *IdP* から *AP* へのリダイレクトを利用する場合，*IdP* に *AP* へのリダイレクト URL を登録しなければならない．

3.4　属性情報の関係の表現

3.2.2 項で述べたように，すべての属性情報には URI が割り当てられている．これによって，すべての属性情報は一意にアクセス可能となるが，各属性情報どうしの関係については表現されていない．属性情報は他の属性情報と依存や派生などの関係を持つことがある．そのため，本提案方式では RDF によって，属性情報間の関係を表現する．RDF（Resource Description Framework）[5] は WWW 上でリソースに関する情報を表すための枠組みである．RDF は URI によってリソース（主語）を識別し，プロパティ（述語）とプロパティ値（目的語）でリソースを記述する．

ある属性情報をクラスで定義する場合，dcq:requires プロパティを用いて，**図 3** のように表現する．dcq:requires プロパティは Dublin Core[⋆1] の拡張プロパティの 1 つであり，「リソースはプロパティ値を**必要とする**」を示す．この dcq:requires プロパティを用いることで，属性情報間の関係を示す．図 3 の例では「定義する属性」と「関係のある属性」が 1 対 1 であるが，1 対多の場合は，そのすべての「関係のある属性」を dcq:requires プロパティで表現する．

ある属性情報のインスタンスを表現する場合，その属性情報と関係しているすべての属性情報を dcq:requires プロパティを用いて表現する．この関係している属性情報は，インスタンスである属性証明書を発行するにあたって検証を行った属性情報であり，依存・派生関係にある属性情報である．

これらのクラスやインスタンスは，その属性情報を管理する *AP* が定義する．*AP* が管理・発行する属性情報について，その発行に際して参照した関連情報をクラスで表現する．クラスは管理・発行する属性情報の種類ごとに存在する．ただし，属性情報が複数の *AP* によって発行される場合は，それらの *AP* で共通のクラスを 1 つ定義する．たとえば，各市役所・区役所が *AP* として，住所の属性情報を発行する場合などが該当する．対して，インスタンスは管理・発行される属性情報ごとに存在する．これは，インスタンスである属性証明書を発行するにあたって検証を行った属性情報および属性証明書が，各利用者ごとに異なるからである．

3.5　パラメータの定義

IdP から *AP* へのリダイレクトには以下の 6 つのパラメータを用いる．

- sign
- identifier
- url
- time
- unique
- nonce

sign は以下のように定義される．

$$\texttt{sign} \leftarrow S(H(\texttt{identifier}||\texttt{nonce}||\texttt{url})) \quad (1)$$

ここで，$S(\cdot)$ はディジタル署名であり，$H(\cdot)$ は一方向性ハッシュ関数である．また，|| は連接を示す．identifier は署名者を識別するユニークな番号であり，url は *AP* へのリダイレクト URL である．nonceは以下のように定義される．

$$\texttt{nonce} \leftarrow \texttt{time}||\texttt{unique} \quad (2)$$

ここで，time は nonce 生成時の UTC 時間であり，一定時間利用されなかった nonce の削除に利用する．unique はユニークな英数字の文字列である．

⋆1 http://dublincore.org/documents/dces/

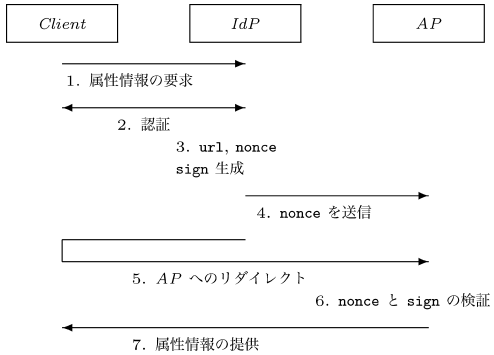

図 4　リダイレクトのプロトコル
Fig. 4　Protocol flow for redirection.

完全なリダイレクト URL は上記のパラメータを用いて，以下のように定義される．

```
https://Authority/UniqueID/Attribute
?idp_identifier=identifier&idp_nonce=nonce&idp_sign=sign
```

ここで，1 行目が `url` であり，`identifier` は署名者である *IdP* を識別するユニークな番号であり，`nonce` と `sign` は前述の定義により生成される．なお，各パラメータを含んだ完全なリダイレクト URL は，URI において使用禁止である値を含まないように，URL エンコードを適切に行う．

3.6　プロトコル

属性情報を取得するための *IdP* から *AP* へのリダイレクトを含むプロトコルを**図 4** に示す．また，図 4 中の各番号に対応する，以下の処理を行う．

(1)　*Client* は *IdP* に属性情報を要求する．
(2)　*IdP* は *Client* を認証する．
(3)　*IdP* は `nonce`，`url`，`sign` を生成する．
(4)　*IdP* は `nonce` を *AP* に送信する．
(5)　*IdP* は完全なリダイレクト URL を生成し，*Client* を *AP* へリダイレクトさせる．
(6)　*AP* は `nonce` と `sign` を検証する．
(7)　*AP* は *Client* に属性情報を提供する．

手順 (2) において，*IdP* は *Client* の公開鍵証明書を用いて，認証を行う．このプロトコルでは，属性情報の要求に際して，必ず *IdP* から *AP* へのリダイレクトが必要となる．そのため，*IdP* はどの *Client* がどのような属性情報を要求しているかをすべて把握している．もし，*Client* が悪質または不正な要求を続ける場合，*IdP* はその *Client* を拒否することができる．手順 (6) において，*AP* は完全なリダイレクト URL に含まれるパラメータの検証を行う．まず，*AP* は完全なリダイレクト URL に含まれている `nonce` が，手順 (4) で *IdP* から送信された `nonce` と一致するかどうかを検証する．一致しない場合，再送攻撃や不正な要求であるため，*Client* からの要求を拒否する．一致した場合，手順 (4) で *IdP* から送信された `nonce` を消去し，完全なリダイレクト URL に含まれている `sign` の検証を行う．*AP* は完全なリダイレクト URL に含まれている `identifier` から，署名者を特定し，署名検証に必要な公開鍵証明書を入手する．*AP* は完全なリダイレクト URL に含まれている `url`，`identifier`，`nonce` を用いて，以下の検証を行う．

$$V(\texttt{sign}) \stackrel{?}{=} H(\texttt{identifier}||\texttt{nonce}||\texttt{url})$$

ここで，$V(\cdot)$ は署名者の公開鍵を用いたディジタル署名の検証である．この検証に問題がなければ，*Client* からの要求を正当なものとし，*Client* に属性情報の提供を行う．もし，`url` を書き換えて，異なる属性情報の取得を試みた場合，`sign` の検証は正しく検証できないため，*Client* からの要求を拒否する．

3.7　属性情報の有効性検証

属性情報を取得した *Client* がその属性情報の有効性を検証する場合を考える．ここでは属性情報を取得した *Client* を検証者と呼ぶこととする．

まず，検証者は取得した属性情報についての RDF を取得し，その RDF に示されているインスタンスである属性証明書を取得する．検証者は取得した属性証明書を検証することで，有効性や正当性を調べることができる．もし，取得した属性情報が発行された根拠とされる属性情報の検証も必要とする場合，検証者は RDF に書かれている関係性のある属性情報とその属性証明書の取得を要求し，検証する．これを繰り返すことにより，必要があれば，最初に取得した属性情報を発行するに至った根拠とされる属性情報のすべてを検証することが可能であり，より厳密で完全な有効性検証が実現される．

3.8　適　用　例

複数の属性情報を同時に持っている場合にのみ付与される属性情報がある．この属性情報の発行と検証を適用例として説明する．

ここでは *IdP* として *idp1* および *ap1* から *ap5* の 5 つの *AP* をプレイヤとして設定する．*ap1* は名前や性別などの基本的な属性情報を管理している．*ap2* は高速道路を管理する道路局の *AP* であり，高速道路通行料割引権を発行することができる．*ap3* は運転免許

表 1　idp1/111 のディレクトリ情報
Table 1　Directory information for idp1/111.

Attribute	属性値
fullname	https://ap1/718/fullname
gender	https://ap1/718/gender
...	...
driverlicence	https://ap3/234/driverlicense
handicap	https://ap4/543/handicap
disease	https://ap5/961/disease
...	...

表 2　*AP* のディレクトリ情報
Table 2　Directory information for *AP*.

Authority	UniqueID	Attribute	属性値
ap3	234	driverlicense	第一種普通
ap4	543	handicap	1 級
ap5	961	disease	心臓病

証を発行する交通局の *AP* である．*ap4* は障害者認定を行う保険局の *AP* であり，*ap5* は医療診断書を発行する病院の *AP* である．また，障害者が運転免許証を保持している場合，高速道路通行料割引権を得ることができる．

属性情報の発行

いま，*Alice* は障害者であり，かつ運転免許証を持っているため，高速道路通行料割引権を得ようとしている．*Alice* の属性情報を管理している *idp1* の *Alice* に関するディレクトリ情報の一部を**表 1** に示し，高速道路通行料割引権の発行に必要な各 *AP* が管理する属性情報を**表 2** に示す．

Alice は道路局の *ap2* に対して，高速道路通行料割引権の発行に必要な属性情報である障害者認定と運転免許証を証明する．*Alice* は *ap2* に対して以下を提示し，*ap2* は *idp1* に対して属性情報の要求を行う．

https://idp1/111/driverlicense
https://idp1/111/handicap

idp1 は *ap2* を認証し，それぞれ以下にリダイレクトさせる．

https://ap3/234/driverlicense
https://ap4/543/handicap

```
<rdf:RDF>
<rdf:Description rdf:about="http://ns/schema#handicap">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:value>1 級</rdf:value>
  <dcq:requires rdf:resource="http://ns/schema#disease"/>
</rdf:Description>
<ap4:handicap rdf:about="https://ap4/543/handicap/cert"/>
<ap5:disease rdf:about="https://ap5/961/disease/cert"/>
</rdf:RDF>
```

図 5　適用例における *ap4* の RDF の例
Fig. 5　An example of RDF from *ap4*.

ap3 と *ap4* はそれぞれリダイレクトが正しいかを検証し，*ap2* に要求された属性情報を提供する．*ap2* は *ap3* と *ap4* から得た属性情報を検証し，*Alice* に高速道路通行料割引権を発行する．

ここで，*ap4* から得た属性情報をその RDF に従って，厳密に検証する場合を考える．*ap4* から提供される RDF の例を一部簡略化して**図 5** に示す．図 5 は *ap4* が発行した UniqueID が 543 の障害者認定について説明している．図 5 の 2 行目は rdf:Description プロパティによって，この RDF は handicap 属性，つまり障害者認定について説明されていることが分かる．4 行目より，この障害者認定の属性値は「1 級」であることが示されており，5 行目は dcq:requires プロパティによって，handicap 属性には disease 属性が必要であることが示されている．8 行目より，*ap4* は障害者認定を行うに際して，*ap5* が発行した disease 属性である ap5/961/disease/cert を根拠としていることが分かる．また，実際に発行された障害者認定の属性証明書は，7 行目に示される URI より取得可能であることを示している．この図 5 で示される RDF は障害者認定を行った *ap4* が発行・保持するものであり，*ap4* によるディジタル署名または XML 署名が行われている．

発行された高速道路通行料割引権は以下のとおりである．

https://ap2/131/discount

ap2 より発行された高速道路通行料割引権に関する RDF の例を**図 6** に示した．図 6 より，高速道路通行料割引権を発行するにあたって，driverlicense 属性（運転免許証）と handicap 属性（障害者認定）が必要であることが示されている．また，実際に高速道路通行料割引権を発行した根拠は，ap3/234/driverlicense/cert および ap4/543/handicap/cert であることが示されている．

```
<rdf:RDF>
<rdf:Description rdf:about="http://ns/schema#discount">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:value>半額</rdf:value>
  <dcq:requires rdf:resource="http://ns/schema#driverlicense"/>
  <dcq:requires rdf:resource="http://ns/schema#handicap"/>
</rdf:Description>
<ap2:discount rdf:about="https://ap2/131/discount/cert"/>
<ap3:driverlicense rdf:about="https://ap3/234/driverlicense/cert"/>
<ap4:handicap rdf:about="https://ap4/543/handicap/cert"/>
</rdf:RDF>
```

図 6　適用例における *ap2* の RDF の例
Fig. 6　An example of RDF from *ap2*.

Alice は発行された高速道路通行料割引権を *idp1* に登録する．

`https://idp1/111/discount` $\overset{\text{リダイレクト}}{\longrightarrow}$ `https://ap2/131/discount`

属性情報の利用

Alice が高速道路通行料割引権を行使する場合を考える．*Alice* は第 2 高速道路で高速道路通行料割引権を行使するため，第 2 高速道路の検証者に以下を提示する．

`https://idp1/111/discount`

検証者は *idp1* に対して属性情報を要求し，*idp1* は検証者を認証し，以下にリダイレクトさせる．

`https://ap2/131/discount`

検証者は *ap2* から提供される RDF による属性情報を検証し，正しければ *Alice* に高速道路通行料割引を実施する．この際，検証者は発行された属性情報である高速道路通行料割引を厳密に検証する場合，*ap2* から提供される図 6 の RDF によって，関係のある属性情報である障害者認定と運転免許証を *ap4* と *ap3* からそれぞれ取得し，検証することができる．

4. 考察と評価

4.1 URI の意義

IdP はハブのように振る舞い，リクエスト URL から実際に管理している *AP* へとリダイレクトする．そのため，属性情報を実際に管理する *AP* が変更になった場合においても，*IdP* へのリクエスト URL は変わらないので，リクエスト URL へ属性情報の取得を要求すれば，*AP* へと適切にリダイレクトされる．

また，*IdP* から *AP* へのリダイレクトは一方向性であるので，*AP* から *IdP* へのリダイレクトや逆参照はできない．これはつまり，『「利用者 A」の「属性 1」は何であるか』という要求はできるが，『この「属性値」は誰の属性であるか』という要求はできないことを意味する．

4.2 分散管理の利点と情報流出による影響範囲

本提案手法は，分散管理する手法における属性情報へのアクセス方法と属性情報の関係性の提示による有効性検証を可能としている．一般に集中管理の場合は，1 つの機関が多くの情報を扱うため，その安全性と可用性に問題が生じる．1 つの機関が多くの情報を扱うため，情報流出が発生した場合，その影響範囲は広範にわたることが予想される．また，可用性も同様であり，1 つの機関が停止した場合の影響範囲は広範であると予想される．

対して分散管理の場合は，1 つの機関が停止した場合においても，その影響範囲は集中管理の場合に比べ，局所的に抑えられる．本提案手法では *IdP* が停止すると，リダイレクトを含むプロトコルが利用できなくなるが，*IdP* が停止していなければ，*AP* が停止していても，属性情報の行使ができなくなるわけではないため，機能停止による影響範囲を局所的にとどめることができる．また，*AP* が停止した場合においても，各 *AP* どうしは独立であるため，影響が他の *AP* に及ぶことを防ぐことができる．

本提案手法における情報流出の影響は，集中管理の場合における影響に比べ，局所的にとどまる．まず，*IdP* から情報流出が発生した場合の影響範囲を考える．なんらかの理由で *IdP* のディレクトリ情報が流出した場合，*IdP* が管理している利用者の属性情報を実際に管理している *AP* の URL が知られる．この状況下で，流出した情報をもとに，*AP* に属性情報を要求しても，プロトコルに基づいた正しい要求でなければ，拒否される．つまり，*IdP* のディレクトリ情報からだけでは属性情報そのものを取得することはできない．

情報流出が発覚した段階で，*IdP* は利用者の属性情報を管理しているすべての *AP* に対して，`UniqueID` の変更を依頼し，ディレクトリ情報を更新する．これによって，流出したディレクトリ情報から，実際の属性情報への関連は断たれる．*AP* の `UniqueID` が変更された状態であっても，*IdP* からのリダイレクト先が変更になるだけであって，*IdP* や *IdP* での `UniqueID` が変わらない限り，属性情報を取得しようとする *Client* には何ら変化はない．これは *IdP* が恒久的な URI を持つことによる利点である．

次に，*AP* から情報流出が発生した場合の影響範囲を考える．なんらかの理由で *AP* のディレクトリ情報が流出した場合，*AP* が管理している利用者の属性情報をが知られる．しかしながら，*AP* の `UniqueID` と *IdP* の `UniqueID` は異なっているので，流出した属性情報

が誰のものであるかは分からない．これは，リクエスト URL からリダイレクト URL を作成できるが，リダイレクト URL からリクエスト URL を求めることはできないからである．

4.3　RDF による属性情報どうしの関係の提示と検証

属性証明書は属性を証明する X.509 証明書であるが，属性証明書は属性証明書どうしの関係を示すことはできない．たとえば，A 市から発行された「A 市民」の属性証明書と B 区から発行された「B 区民」の属性証明書は，属性値が異なるので，同じ意味をなさない．しかしながら，実際には「A 市民」も「B 区民」も同じ「住民」を表している．本提案方式ではこの解決に RDF を用いた．RDF にはクラスの定義とインスタンスの定義があり，属性証明書はインスタンスである．本提案方式では，A 市から発行された「A 市民」および B 区から発行された「B 区民」は，それぞれ同じ「住民」クラスから派生したクラスであり，そのインスタンスとして，それぞれ「A 市民」属性証明書と「B 区民」属性証明書があることを RDF で示すことを可能としている．

また，この RDF にはクラスの定義として，発行に必要とする属性情報に関する情報が書かれている．そのため，属性情報を取得した *Client* はその属性情報が発行されるに至った根拠を確認することができる．3.7 節で説明したように，取得した属性情報とその属性情報が発行されるに至った根拠となる属性情報を含むすべての関係ある属性情報を検証可能である．このため，本来は有効期間が長い属性情報がなんらかの理由で失効した場合に，その属性情報に依存している他の属性情報がまだ失効していない場合であっても，厳密にその有効性を検証することが可能となる．

4.4　リダイレクト URL による不正

リダイレクト URL を書き換え，異なる属性情報の取得を試みることが考えられる．この場合，`Authority`，`UniqueID`，`Attribute` のいずれか，または複数を書き換え，不正に属性情報の取得を試みる．しかしながら，完全なリダイレクト URL に含まれる `sign` は式 (1) が示すように，`url` を含んでいるため，`sign` は正しく検証できない．

次に，`sign` を正しく検証させるために，`identifier` を自分自身に書き換え，署名者を偽って `sign` を不正に生成する場合を考える．この場合，`sign` に含まれるすべての情報を偽造可能だが，3.6 節の手順 (4) において送信されるべき `nonce` が送信されていないため，*AP* は要求を拒否する．

同様にして，不正な *IdP* を用いて，不正なリクエストを行うときを考える．*AP* は `sign` を検証する際に，3.6 節の手順 (6) に従って，リダイレクト URL に含まれる `identifier` から署名者を特定するが，不正な *IdP* の場合，公知ではない *IdP* であることが分かるため，*AP* は要求を拒否する．

完全なリダイレクト URL をすべて偽造し，*AP* に直接要求を行う場合を考える．この場合も同様に，`sign` および `nonce` の検証によって，要求は拒否される．

すでに利用された完全なリダイレクト URL を再送し，再び属性情報を取得することを考える．この場合，3.6 節の手順 (6) において `nonce` は削除されているので，完全なリダイレクト URL を再送しても，同様に再び属性情報を取得することはできない．

4.5　可　用　性

IdP が停止した場合，リダイレクトを含むプロトコルが利用できなくなるため，属性情報の取得ができなくなる．これは集中管理の場合と同じ可用性である．一方，*AP* が停止した場合，停止した *AP* の属性情報は取得できなくなる．しかしながら，各 *AP* どうしは独立であるため，他の *AP* はそのまま利用可能である．これは分散管理の場合と同じ可用性である．全体の可用性は *IdP* と *AP* の直列接続になるため，集中管理と分散管理のいずれの方式よりも低下する．特に，本提案方式では *IdP* がプロトコルにおける重要な機能を担っているため，*IdP* の可用性を上げることで，全体の可用性を改善することが可能である．

5.　ま　と　め

本論文では，分散管理された属性情報を一意に指し示す方法として URI，属性情報どうしの関係性を示す方法として RDF を用いることで，属性情報の有効利用と有効性検証を可能とする属性情報の分散管理方式を提案した．

属性情報を集中管理する場合，管理する機関へのアクセス集中や情報流出時の影響範囲の問題があるが，分散管理によって負荷分散と情報流出時のリスク軽減ができる．これに加えて，本提案方式によって，分散管理されている属性情報へのアクセスは，ID プロバイダを介してリダイレクトすることで，属性情報の利用者に対して集中管理と同じ利便性を確保しつつ，安全性への配慮が可能となった．

また，複数の異なる属性認証機関によって管理されている属性情報どうしの関係を RDF によって示すことで，属性情報の検証者はその属性情報が発行された根拠を確認し，より厳密な属性情報の検証が可能となる．

今後の展望として，URI による分散型 ID 認証システムである OpenID との連携を考えていきたい．特に，OpenID 拡張使用である Attribute Exchange や Simple Registration Extension などの属性交換拡張仕様との関係を整理し，連携を検討していきたい．

参 考 文 献

1) 電子商取引推進協議会：属性認証ハンドブック (2005). http://www.ecom.jp/results/results16.html
2) 柿崎淑郎，辻　秀一：一意にアクセス可能な属性情報の分散管理方式，情報処理学会研究報告，2008-IS-105(10), pp.61–68 (2008).
3) 柿崎淑郎，辻　秀一：属性間関係性を用いた属性認証における失効遅延削減方式，情報処理学会論文誌，Vol.49, No.2, pp.893–901 (2008).
4) 千葉昌幸，漆嶌賢二，前田陽二：属性情報プロバイダ：安全な個人属性の活用基盤の提言，情報処理学会論文誌，Vol.47, No.3, pp.676–685 (2006).
5) Manola, F. and Miller, E.: RDF Primer, W3C Recommendation (2004). http://www.w3.org/TR/rdf-primer/
6) Berners-Lee, T.: Linked Data (2007). http://www.w3.org/DesignIssues/LinkedData.html
7) Recordon, D. and Reed, D.: OpenID 2.0: A platform for user-centric identity management, *Proc. 2nd ACM workshop on Digital identity management* (*DIM '06*), New York, NY, USA, ACM, pp.11–16 (2006).
8) Housley, R., Polk, W., Ford, W. and Solo, D.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC3280 (2002).
9) Farrell, S. and Housley, R.: An Internet Attribute Certificate Profile for Authorization, RFC3281 (2002).
10) Berners-Lee, T., Fielding, R. and Masinter, L.: Uniform Resource Identifier (URI): Generic Syntax, RFC3986 (2005).
11) Hoyt, J., Daugherty, J. and Recordon, D.: OpenID Simple Registration Extension 1.0 (2008). http://openid.net/specs/openid-simple-registration-extension-1_0.html

(平成 21 年 5 月 25 日受付)
(平成 21 年 11 月 6 日採録)

柿崎　淑郎（正会員）

1980 年生．2003 年東海大学工学部電子工学科卒業．2008 年同大学院博士課程修了，博士（工学）．現在，東京理科大学工学部第一部電気工学科助教．情報セキュリティ，情報システム等の研究に従事．電子情報通信学会，IEEE 各会員．

吉田　慶章

2007 年東海大学電子情報学部情報メディア学科退学．2009 年東海大学大学院工学研究科修士課程修了，修士（工学）．現在，日本アイ・ビー・エム株式会社に勤務．2008 年情報処理学会第 69 回全国大会大会優秀賞受賞．2009 年情報処理学会山下記念研究賞受賞．在学時はセマンティックウェブ，自然言語処理，特にオントロジを用いた知識支援の研究に注力．

辻　　秀一（正会員）

1969 年大阪大学基礎工学部電気工学科卒業．1974 年同大学院基礎工学研究科博士課程修了，工学博士．1974〜2000 年三菱電機（株）に勤務．この間，研究所および開発部門にて，ヒューマンインタフェースや人工知能システム等の研究開発に従事．1997〜2000 年電子商取引実証推進協議会へ出向．2000 年 4 月より東海大学に勤務．現在，情報通信学部組込みソフトウェア工学科教授．電子情報通信学会，人工知能学会，電気学会，IEEE 各会員．