社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－IS－105 (10)
2008／8／29

一意にアクセス可能な属性情報の分散管理方式

柿崎 淑郎† 辻 秀一‡

†東京理科大学
102-0073 東京都千代田区九段北 1-14-6
kakizaki@ee.kagu.tus.ac.jp

‡東海大学
259-1292 神奈川県平塚市北金目 1117
htsuji@keyaki.cc.u-tokai.ac.jp

あらまし 属性情報の集中管理方式は，利便性が高い反面，情報漏洩の発生によって，同時に多数の情報が漏洩する問題がある．分散管理方式の場合，ある機関で情報漏洩が発生しても，他の機関で管理されている情報は漏洩しないが，どの機関がどの情報を管理しているかは知っておく必要性がある．本稿では一部の属性情報を異なる機関に管理委譲しても，その属性情報に一意にアクセス可能とし，属性情報の集中管理方式と分散管理方式の長所を併せ持つ，属性情報の管理方式を提案する．

A decentralized management method for uniquely accessible attribute information

Yoshio KAKIZAKI† Hidekazu TSUJI‡

†Tokyo University of Science
1-14-6 Kundankita, Chiyoda-Ku, Tokyo 102-0073, Japan
kakizaki@ee.kagu.tus.ac.jp
‡Tokai University
1117 Kitakaname, Hiratsuka-Shi, Kanagawa 259-1292, Japan
htsuji@keyaki.cc.u-tokai.ac.jp

Abstract The centralized management of attribute information is convenient; however it has a problem that a lot of information is leaked when data leakage is happened. In decentralized management, the information which is managed by other authorities is not leaked when the information is leaked from a certain authority. However it is necessary to know which authority manages information. In this paper, we propose a decentralized management method for uniquely accessible attribute information though the management of some attribute information is delegated to other authorities.

1 はじめに

属性情報とは，その主体が持つ属性・権限・職責・資格・地位などである．属性を持つ主体は個人であることが多いが，組織が持つ場合もある．個人が持ち得る属性情報として，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深く密着したものもあり，しばしばプライバシの問題が発生することがある．これらの属性情報は個人情報の一部として扱われることも多い．しかしながら，入会審査などの登録作業時に属性情報を要求されたり，権限や身分を証明するために，属性情報が必要になるなど，多くの場面で属性情報が利用されている現状がある．特に近年では，個々人に適応したサービス提供を行う 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方式は大きく分けて 2 つある．1 つは集中管理方式であり，もう 1 つは分散管理方式である．集中管理方式は 1 つの機関が集中的に属性情報を管理する方式である．この方式の場合，全ての属性情報が 1 つの機関に集約されているため，

属性情報の利便性や再利用性は高いが，その反面，膨大な情報を管理するに足る信頼の確保が必要であり，ひとたび情報漏洩が発生すれば，その影響範囲が広範囲に及ぶ懸念がある．分散管理方式の場合，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く，情報漏洩発生時の影響範囲は集中管理方式に比べて，軽微となる．しかしながら，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り，属性情報の活用の面で問題が残る．

本稿では集中管理方式における情報漏洩発生時の影響範囲の問題と分散管理方式における情報の散逸の問題を解決するため，両方式の長所で両方式の欠点を補い合えるように，一部の属性情報を異なる機関に管理委譲しても，その属性情報に一意にアクセス可能とする管理方式を提案する．属性情報を実際に管理している機関がどこであろうとも，URIによって一意にアクセス可能とし，分散管理される属性情報の利便性を向上させるとともに，複数の機関に属性情報を管理委譲させることができるため，ある機関から情報漏洩が発生しても，その影響範囲を最小限に止めることが可能となる．

2 関連研究

千葉らは個人属性を安全に交換・管理する情報化基盤として，属性情報プロバイダを提案している [2]．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，情報漏洩発生時の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．

他方，属性情報を分散管理する試みとして，松本らによって，秘密分散方式を用いた分散属性認証方式が提案されている [3]．この方式は属性情報を管理する機関からの情報漏洩に対して耐性を持たせるために，秘密分散法を利用して，属性情報を分散した分散属性情報とユーザのIDを組にして複数の機関に登録する．ユーザは分散属性情報を示す分散属性証明書を一定の数だけ集めて，匿名属性証明書として提示することで，必要以上のプライバシ情報を流出させず，属性証明書を提示しているユーザがサービスを利用するユーザと同じことを確認することができる．

セマンティックウェブの分野では，Berners-LeeがLinked Dataを提案している [4]．Linked Dataでは4つのルールとして，識別のためにURIを用いることなどが示されており，注目が集まっている [5]．また，近年注目を集めているOpenID[1]はURIを用いた分散ID認証技術である．

3 一意にアクセス可能な分散管理

3.1 概要

本提案方式は一部の属性情報を異なる機関に管理委譲しても，その属性情報に一意にアクセス可能とし，属性情報の集中管理方式と分散管理方式の長所を併せ持つ，属性情報の管理方式である．以降で述べる属性情報の範囲は個人を対象とする．ある個人に対する属性情報は全てURIが与えられ，一意にアクセス可能である．また，属性情報の集中管理による一極集中の問題を解決するために，属性情報を分散管理できるように，複数の機関に属性情報を管理委譲を行う．管理委譲された属性情報は，実際に管理されている機関へとリダイレクトされることで，アクセス可能とする．属性情報の取得はアクセスポリシによって許可か不許可かが判断され，不必要な情報公開を抑制する．

本提案方式における利点を以下に挙げる．

- URI表現による一意なアクセスが可能
- リダイレクトによる他機関への管理委譲
- アクセスポリシによる情報公開の制御

また，本稿で使う用語を以下のように定義する．

IdP アイデンティティプロバイダ（Identity Provider）であり，ID情報や属性情報を集中管理する信頼できる第3者機関である．*AP*よりも上位の機関である．

AP 属性プロバイダ（Attribute Provider）であり，属性情報を集中管理する信頼できる第3者機関である．*IdP*よりも下位の機関である．

[1] http://openid.net/

要求者　属性情報を取得しようとするプレイヤであり，取得した属性情報を利用してサービスを展開するサービス提供者である．

3.2 URIによる表現

3.2.1 URI

URI (Uniform 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り，RFC3986で標準化されている [6].

リソースを一意に指し示すことができるURIの性質を利用し，属性情報を一意にアクセス可能とする．以下に，属性情報を一意にアクセス可能とするURIの構成を示す．

```
scheme://Authority/UniqueID/Attribute
```

scheme　URIで示されたリソースを取得するための手段であり，一般的にhttpやftpが用いられる．本提案方式ではhttpを用いることを想定しており，通信路の安全性を考慮し，httpsを用いることとする．

Authority　ID情報や属性情報を集中管理する*IdP*または属性情報を集中管理する*AP*である．

UniqueID　*Authority*上において管理している利用者を識別する番号であり，重複や再利用はされないユニークな番号である．

Attribute　*Authority*上で管理されている属性情報である．

3.2.2 REST

REST (Representational State Transfer) [7] はリソースを特定するURIとリソースにアクセスし操作するためのプロトコルを定義したWebアーキテクチャスタイルであり，Webサービスでよく利用されている．RESTのプロトコルにはHTTPやHTTPSが利用され，GET（取得），POST（更新），PUT（削除），DELETE（新規作成）のHTTPメソッドが利用される．また，RESTはHTTPを利用するため，HTTP準拠の認証が利用可能である．

3.2.3 リクエストとレスポンス

本方式はURIに対してRESTによるリクエストを行い，URIに紐付けられた属性情報をレスポンスとして受け取る方式である．

リクエスト

リクエストは以下のように行われる．

```
GET /000001/gender HTTP/1.1
Host: idp.test
```

上記のリクエストは*Authority*としてidp.testを指定し，*UniqueID*が000001の*Attribute*であるgenderを要求している．

レスポンス

レスポンスとして返されるステータスコードを以下に示す．

- 200 OK
  - 許可
  - 条件付許可（許可の場合）
- 303 See Other
  - リダイレクトの場合
- 403 Forbidden
  - 要確認
  - 条件付許可（不許可の場合）
  - 不許可
  - 存在しないユーザ

「200 OK」が返された場合，アクセスは許可され，要求したリソースであるgenderの属性情報が返される．属性情報の管理委譲によって他の*AP*にリダイレクトされる場合，「303 See Other」が返される．属性情報の管理委譲は後述の3.4節で説明する．アクセスが許可されない場合，一律に「403 Forbidden」が返され，要求したリソースであるgenderの属性情報は返されない．アクセスが許可されるか否かは後述の3.5節で説明するアクセスポリシによって規定される．

3.3 属性情報ディレクトリ

IdP が属性情報を管理する場合のディレクトリ構成について説明する．*IdP* によって管理されている利用者を識別するために，各利用者にはユニークな *UniqueID* が割り振られ，*UniqueID* によって管理される．各 *UniqueID* のディレクトリには属性情報を示す *Attribute* が存在する．*Attribute* は基本領域と拡張領域に分けられ，基本領域には基本的な属性情報である名前，性別，生年月日などが格納される．拡張領域には基本領域に含まれない特殊な属性情報が格納され，その属性情報毎に新しいディレクトリを構成し，属性情報を格納する．属性情報が管理委譲されている場合，*Attribute* の属性値はリダイレクト先の URI となる．属性情報の管理委譲は後述の 3.4 節で説明する．各属性情報にはアクセスが許可されるか否かを決定するためのアクセスポリシが付けられている．アクセスポリシは後述の 3.5 節で説明する．

AP が属性情報を管理する場合のディレクトリ構成は，図 1 とほぼ同じである．*AP* のディレクトリ構成は基本領域と拡張領域の区別はないが，各属性情報には URI によって一意にアクセス可能である必要がある．また，*AP* は属性情報の管理委譲は行わないため，属性値にリダイレクト先の URI が記載されることはない．

3.4 属性情報の管理委譲

IdP は全ての属性情報を管理することもできるが，その場合，集中管理方式と同様である．いくつかの属性情報を他の *AP* に管理を委譲させることで，属性情報を分散管理方式のように扱うことができるようになる．すなわち，*IdP* は全ての属性情報を管理しなくてもよくなるため，*IdP* への属性情報の一極集中を避けることが可能となる．

IdP に管理されている属性情報を他の *AP* に権利委譲する場合，管理委譲する *Attribute* の値を属性値ではなく，管理委譲する *AP* 内の属性情報へとリダイレクトするための URI とする．この基本的な考え方は 303 URIs [8] と同様であり，*IdP* が持つ属性情報の URI から，管理委譲する *AP* 内の属性情報の URI へリダイレクトする．

表 1: IdP.test/01234 のディレクトリ情報

属性名	属性値	ポリシ
fullname	https://AP.test/718/fullname	許可
gender	male	許可
birth	https://AP.test/718/birth	不許可

3.5 アクセスポリシ

IdP または *AP* が管理する属性情報の属性値へのアクセスは，アクセスポリシによって制御される．属性情報を指し示す URI は誰でも知り得るが，その属性値の取得可否はアクセスポリシによって判断される．アクセスポリシを以下に定義する．

- 許可
- 条件付許可
- 要認証
- 不許可

許可　アクセスを許可するポリシであり，その属性値は公開されているという意味を持つ．

条件付許可　ある条件下においてアクセスを許可するポリシであり，ホワイトリスト方式によってアクセスを制御する．ホワイトリストに記載されている場合は許可となり，そうでない場合は不許可となる．

要認証　属性値を取得しようとするクライアントに対して，アクセス権限があるか否かを認証することで，アクセスの可否を決定する．

不許可　アクセスを許可しないポリシであり，その属性値は非公開であることを意味する．

3.6 動作例

IdP である IdP.test における *UniqueID* が 01234 の *Attribute* を表 1 に示す．また，管理委譲先の *AP* である AP.test における *UniqueID* が 718 の *Attribute* を表 2 に示す．

属性情報を取得する際のフロー図を図 2 に示し，動作例を以下に説明する．

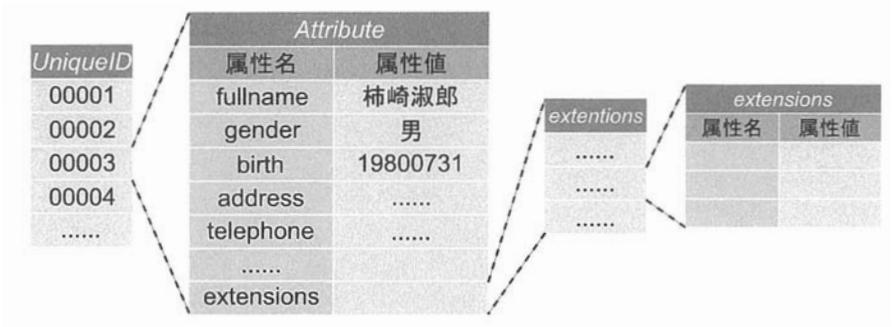

図 1: 属性情報ディレクトリの構成例

表 2: AP.test/718 のディレクトリ情報

属性名	属性値	ポリシ
fullname	Yoshio KAKIZAKI	許可
birth	19800731	不許可

属性値が返る場合

図 2 の上段はアクセスポリシによって許可され，属性値が返る場合の動作例を示している．プレイヤは属性情報を取得しようとする要求者と *IdP* である IdP.test の 2 者である．

まずはじめに，要求者は取得したい属性情報を IdP.test にリクエストする．IdP.test は要求された属性情報のアクセスポリシを確認する．要求者が要求している属性情報は `IdP.test/01234/gender` であり，表 1 によれば，アクセスポリシは「許可」である．そのため，IdP.test は要求者に対して，要求された属性情報を返す．

リダイレクトされる場合

図 2 の中段はアクセスポリシによって許可され，他の *AP* にリダイレクトされる場合の動作例を示している．プレイヤは属性情報を取得しようとする要求者と *IdP* である IdP.test および *AP* である AP.test の 3 者である．

まずはじめに，要求者は取得したい属性情報を IdP.test にリクエストする．IdP.test は要求された属性情報のアクセスポリシを確認する．要求者が要求している属性情報は `IdP.test/01234/fullname` であり，表 1 によれば，アクセスポリシは「許可」である．ここで，`IdP.test/01234/fullname` は `https://AP.test/718/fullname` へのリダイレクトである．そのため，IdP.test は要求者に対して，リダイレクト先の URL を返す．

要求者は返されたリダイレクト先である `https://AP.test/718/fullname` にリクエストする．AP.test は要求された属性情報のアクセスポリシを確認する．要求者が要求している属性情報は `https://AP.test/718/fullname` であり，表 2 によれば，アクセスポリシは「許可」である．そのため，AP.test は要求者に対して，要求された属性情報を返す．

アクセスポリシで拒否される場合

図 2 の下段はアクセスポリシによって拒否される場合の動作例を示している．プレイヤは属性情報を取得しようとする要求者と *IdP* である IdP.test の 2 者である．

まずはじめに，要求者は取得したい属性情報を IdP.test にリクエストする．IdP.test は要求された属性情報のアクセスポリシを確認する．要求者が要求している属性情報は `IdP.test/01234/birth` であり，表 1 によれば，アクセスポリシは「不許可」である．そのため，IdP.test は要求者に対して，アクセスができない旨の通知を返す．

4 検討および考察

4.1 IdP と AP の要件

属性情報を管理する *IdP* と *AP* は共に信頼できる第 3 者機関（Trusted Third Party; TTP）であることを想定している．

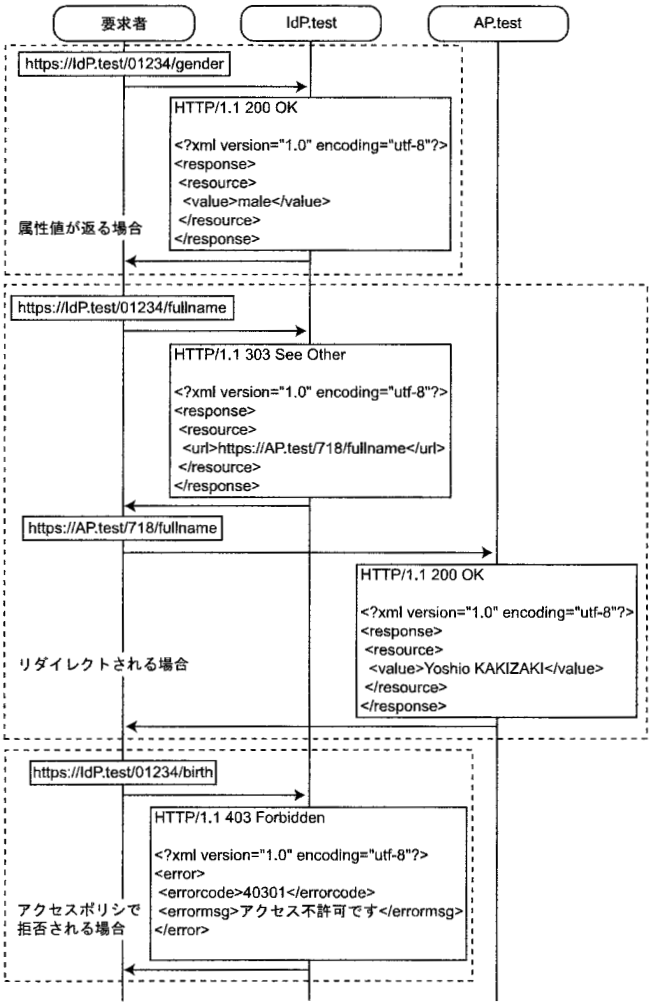

図 2: 動作例

IdP の要件としては，公開鍵証明書などとの紐付けを行うなど，少なくとも ID 情報を証明できることが挙げられる．また属性情報の管理委譲を行わない場合，属性情報も証明できる必要がある．属性情報はポリシに基づいた情報公開を行うことが必要である．

AP の要件としては，属性値を証明できることが挙げられる．これは属性証明書などを利用する．また，属性情報はポリシに基づいた情報公開を行うことが必要である．

4.2 URI の有効性

属性情報を他の *AP* に管理委譲している場合，*IdP* はハブのように振る舞い，属性情報へのリクエストを実際に管理している *AP* へとリダイレクトする．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* にリクエストをすることによって，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行われるが，*AP* から *IdP* へのリダイレクトや逆参照は行えない．これはつまり，『「利用者 A」の「属性 1」は何であるか』というリクエストはできるが，『この「属性値」は誰の属性であるか』というリクエストはで

きないことを意味する．

本稿では属性情報の取得のみを検討したが，RESTによって更新，削除，新規作成も可能である．URIとRESTによって，利便性の高い属性情報の管理と運用が可能になるものと思われる．これについては，今後の検討課題である．

4.3 ポリシの整合性

IdP または *AP* が管理する属性情報の属性値へのアクセスは，アクセスポリシに従って制御されている．属性情報の管理を委譲している場合に，*IdP* から *AP* にリダイレクトされるが，この際に，ポリシ整合性の問題が発生する可能性がある．

3.6 節を例として説明する．表 1 では「不許可」であるにも関わらず，表 2 における birth のアクセスポリシが，「許可」である場合を想定する．この場合，`https://IdP.test/01234/birth` にアクセスした場合，アクセスポリシによって拒否されるが，何らかの手段で，属性情報の管理委譲先である `https://AP.test/718/birth` を取得して直接アクセスした場合，属性情報が取得されてしまう問題が発生する．本来ならば，*IdP* はこの属性情報の公開を「不許可」としているため，属性情報の管理を委譲されている *AP* もこの属性情報の公開を「不許可」としなくてはならない．

また，属性間に関係性がある場合においては，別のポリシ不整合問題が発生する．属性情報にはある条件を満たすことによって発生する属性情報も存在する．そのような属性情報を属性間の関係性を持った属性情報と呼ぶ [9]．例えば，図 3 に示されるように，*Attribute3* は *Attribute1* と *Attribute2* を同時に持つことによって，発生する属性情報である．この条件下で，厳密に *Attribute3* の有効性を検証する場合，*Attribute1* および *Attribute2* が有効であることを検証しなくてはならない．そのため，これら 3 つの属性情報を取得する必要性がある．しかしながら，*Attribute2* が *Attribute3* よりも厳しいアクセスポリシで管理されている場合，*Attribute3* を厳密に検証しようとする検証者は，*Attribute2* を取得することができず，有効性検証ができなくなる．

このように，*IdP* と *AP* のアクセスポリシは整合性を取っておく必要性がある．

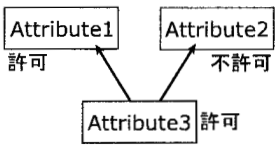

図 3: 属性間に関係性がある場合

5 まとめ

本稿では属性情報の集中管理方式と分散管理方式の長所を併せ持つ，一部の属性情報を異なる機関に管理委譲しても，その属性情報に一意にアクセス可能とした管理方式を提案した．属性情報を実際に管理している機関がどこであろうとも，URI によって一意にアクセス可能とし，分散管理される属性情報の利便性を向上させた．

今後の展開として，OpenID への適応可能性を検討する．また，本稿では検討を行わなかった属性情報の更新，削除，新規作成についても，今後の検討が必要である．

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] 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.

[3] 松本勉, 四方順司, 清藤武暢, 古江岳大, 上山真貴子. 分散属性認証方式に対する基本検討. 情報処理学会研究報告, 2005-CSEC-30(45), 2005.

[4] T. Berners-Lee. Linked Data, 2007. `http://www.w3.org/DesignIssues/LinkedData.html`.

[5] A. Shakya and H. Takeda. A Report on Linked Data. 人工知能学会セマンティックウェブとオントロジー研究会, SIG-SWO-A801-07, 2008.

[6] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax, RFC3986, 2005.

[7] Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine, 2000.

[8] L. Sauermann and R. Cyganiak. Cool URIs for the Semantic Web. Technical Report, W3C Interest Group Note 31 March 2008. http://www.w3.org/TR/cooluris/.

[9] 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.