一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
IA2016-57(2016-11)

属性情報を秘密分散した閾値型認証システムの設計

伊藤　友浩†　　小谷　大祐††　　岡部　寿男††

† 京都大学大学院情報学研究科　〒 606–8501 京都府京都市左京区吉田本町
†† 京都大学学術情報メディアセンター　〒 606–8501 京都府京都市左京区吉田本町
E-mail: †ito@net.ist.i.kyoto-u.ac.jp, ††{kotani,okabe}@media.kyoto-u.ac.jp

あらまし　ID 連携は，単一の認証サービスの結果を連携しているサービスが利用するため，認証サービスへの不正アクセスや，ネットワークの障害が起こると，連携している全てのサービスの利用に影響が出る．これを防ぐために複数の認証サービスを利用すると，ユーザの属性情報を分散させなければならず，プライバシー保護上の課題がある．著者らは，閾値型認証と秘密分散を用いて上記の問題点を解決するシステムを提案している．本稿では，分散時でも管理者が能動的に属性情報を管理でき，属性情報更新時も整合性が維持できるように提案手法の詳細な設計を行う．
キーワード　属性情報，秘密分散，閾値型認証

Design of a Threshold-based Authentication System Which Provides Attributes Using Secret Sharing

Tomohiro ITO†, Daisuke KOTANI††, and Yasuo OKABE††

† Graduate School of Informatics, Kyoto University　Yoshida-Honmachi, Sakyo-ku, Kyoto, 606–8501 JAPAN
†† Academic Center for Computing and Media Studies, Kyoto University　Yoshida-Honmachi, Sakyo-ku, Kyoto, 606–8501 JAPAN
E-mail: †ito@net.ist.i.kyoto-u.ac.jp, ††{kotani,okabe}@media.kyoto-u.ac.jp

Abstract　In identity federation, each service provider verifies the identity of a user based on authentication performed by an authentication server called an Identity Provider (IdP). When the IdP suffer troubles like an unauthorized person has cracked into the IdP or the IdP is unreachable due to a network problem, all services in the federation are affected by them. Simple replication of servers for the IdP causes privacy concern because raw attribute values of users must be copied to many servers, including some servers that may not be fully trusted. In order to maintain the function as an IdP even under such troubles, we propose a system in which servers of the IdP are distributed and cooperate using threshold-based authentication and secret sharing. In this paper, we design the system from the viewpoint of management of attribute values so that an administrator of the IdP can update the values anytime while keeping consistency of attribute values among the servers in the IdP.
Key words　Attributes, Secret Sharing, Threshold-based Authentication

1. はじめに

インターネットの普及により，ユーザは数多くのサービスをインターネット上で受けることが可能になった．しかし，それぞれのサービスで個別の認証情報を求められることから，ユーザは数多くの認証情報を管理しなければならず，サービス提供側もユーザの認証情報を安全に管理しなければならないことから，双方にとって大きな負担となっていた．

この問題点については，SAML [1] や OpenID [2] などの ID 連携技術を用いてシングルサインオンを導入することで解決することができる．シングルサインオンは，各サービス提供者が行っていた認証部分を分離，統合し，認証サービスで行われた認証結果を各サービス提供者が信頼するシステムである．これにより，ユーザは認証情報を 1 つ保持するだけで良く，サービス提供者も認証情報を預かる必要はなくなる．

しかし，シングルサインオンにも課題が残っていると我々は考えている．1 つ目の課題として，認証サービスでサーバのハードウェア障害やネットワーク障害が起きた場合，ユーザは連携している全てのサービスにログインすることができなくなることが挙げられる．この課題に対しては認証サーバの冗長化

が有効であるが，単純な冗長化を行うと，不正アクセスを試みる攻撃者はいずれか 1 台の認証サーバの脆弱性を突けば良いので，認証サーバの管理者権限を奪取されるリスクが高まり，連携している全サービスのユーザアカウントが不正利用される恐れが高まる． 2 つ目の課題として，現在のシングルサインオンの 1 実装モデルである Shibboleth [3] モデルでは，認証サーバがユーザの属性情報を保持しサービス提供者に提供する．したがって，攻撃者に認証サーバの管理者権限を奪取された場合，認証サーバにアカウントがある全ユーザの属性情報が漏洩する恐れがある．認証サーバの冗長化を行うと，同様の理由で管理者権限を奪取されるリスクが高まるため，属性情報が漏洩するリスクも高まる．

我々は 1 つめの課題に対して閾値型認証を用いることで， 2 つめの課題に対して属性情報の秘密分散管理を行うことで解決することを提案している [4], [5]．本提案手法は，閾値以下の分散認証サービスが障害下にあっても連携サービスを利用でき，閾値以下の分散認証サービスの管理者権限が奪取されたとしてもユーザのアカウントが悪用されず，かつユーザの属性情報が漏洩しないシステムを目指している．我々は，文献 [4] で閾値型認証と属性情報の秘密分散管理を用いることを提案し，文献 [5] で閾値型認証のプロトコルの詳細な設計を行った．本稿では，属性情報の秘密分散管理について，要件定義を行い，プロトコルの設計，考察を行う．

以下，本稿の章構成について述べる．まず，2. 章で関連研究を紹介した後，3. 章で閾値型認証のプロトコルについて説明する．4. 章で属性情報の秘密分散管理についてプロトコルの設計を行い，5. 章で考察する．最後に，6. 章でまとめを行う．

2. 関 連 研 究

本章では，先行研究について紹介した後，属性情報を秘密分散管理する我々の提案に特に関連性が高い分散属性認証方式を紹介する．次いで，提案手法に使用する秘密分散について説明する．

2.1 認証の強化と属性情報の保護

ID 連携の環境においては，認証サービスのログインに成功すると全ての連携サービスにログインに成功することと等しいので，認証の強化という点で従来からのリスクベース認証や多要素認証が用いられている．リスクベース認証はユーザが過去に使用した IP アドレスや端末を記憶しておき，通常とは異なるログインの場合，二段階目（二要素目）の認証を求めるシステムである．多要素認証の実装としては，スマートフォンを使った所有物認証を行うことができる Tiqr [6] がある． Tiqr は ID 連携環境で多要素認証の一つとして実際に運用されている．提案手法の閾値型認証は，複数の分散認証サーバでそれぞれ認証を行うため，多要素認証を取り入れることが可能であり，認証の強化に役立つ．しかし，本研究で想定している認証サーバの脆弱性を突いた管理者権限奪取に対しては多要素認証では対処することができず十分ではない．

一方，連携 ID のプライバシー保護の観点から，BlindIDM [7] が提案されている．これは，認証サービスを外部にアウトソーシングする Identity as a Service（ IDaaS ）環境にプロキシ再暗号化を用いて，外部事業者がユーザの属性情報を見ることができないようにシステムである．提案手法でも分散認証サーバを外部事業者のもとに置くことを想定しており，属性情報の暗号化を行って預けていることから同等のプライバシー保護が行えている．しかし，BlindIDM プロトコルではアウトソーシング前の認証機関 Host Organization も利用しなければならず，ここで障害などが起こってしまうと全ての連携サービスに影響してしまう．したがって，本研究で考える要求要件の実現手法としては十分ではない．

2.2 分散属性認証方式

属性情報の秘密分散管理についての関連研究として，分散属性認証方式 [8] が提案されている．分散属性認証方式では，ユーザの属性情報のプライバシー保護について着目し，属性情報の選択的な提示と属性提供機関からの漏洩困難性の向上を達成するために，属性情報を秘密分散し管理する．

分散属性認証方式のプロトコル [9] は図 1 のようになっている．まず，(1) User は User Assistant と呼ばれるアプリケーションにログインする．(2) User は User Assistant を通して Service Point へ利用要求を送り，(3) Service Point は，サービス利用に必要な属性値を選ぶ関数を提示する．(4) User Assistant は各 Shared Attribute Authority に関数と自身の証明書を提示し，(5) 認証後，分散属性証明書をそれぞれ受け取る．(6) User Assistant は各分散属性証明書を検証した後，証明書をまとめて匿名属性証明書を作成した上で， Service Point へ提出する． Service Point は各分散属性証明書の署名を検証してから，属性値の復元の計算を行い， User に属性情報に応じたサービスを提供する．なお， このプロトコル中では， Dealer と User Assistant は不正を行わないという仮定が置かれている．

分散属性認証方式では User Assistant が認証機関， Shared Attribute Authority が（分散）認可機関と役割が分かれているのに対し，現行のシングルサインオンで利用されている Shibboleth モデルでは，認証サーバが認証兼認可機関を担っている．また， Shibboleth モデルでは User がブラウザを用いて認証を行うため，分散属性認証方式の User Assistant が存在しない．さらに，データの受け渡しは User を通して行われるので， User は属性値を偽造や不正を行う可能性がある．したがって，分散属性認証方式を現行のシングルサインオンモデルに適応することができない．よって，本研究では User が不正を行うことができず，かつ属性情報漏えいの困難性をあげ，現行のシングルサインオンモデルに適応できる手法を提案する．

2.3 秘 密 分 散

秘密分散とは，秘密にしたい情報を複数の情報に分割することで暗号化を行い，分割した情報を集めることで元の情報を復元することができる暗号のことである．集めた分散情報の数が一定数に満たない場合は，元の情報を復元することができない．本稿では Shamir の （k，n） 閾値秘密分散法 [10] を用いる．

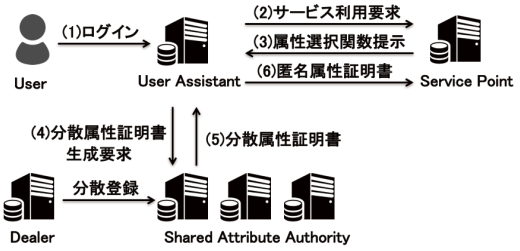

図 1　分散属性認証方式

3. 閾値型認証

本章では閾値型認証について説明する．まず，閾値型認証の概要について説明し，課題を解決できることを示した後，具体的な提案手法について説明する．

3.1 概　　要

閾値型認証の概要について図 2 を用いて説明する．従来のシングルサインオンでは認証サービスが単一であるため，障害が起こると連携サービスを利用できず，管理者権限を奪取されると連携サービスのアカウントを悪用される恐れがあった．閾値型認証では複数の認証サービスを用意し，各認証サービスで認証を行い，閾値以上の認証成功結果が返ってくればサービスを提供すると定める．本研究ではこの複数の認証サービスのことを，認証権限が分散していることから分散認証サーバと表す．図では分散認証サーバを 3 つ準備し，閾値を 2 と定めた．したがって，1 つの分散認証サーバが障害で利用できなくても他の分散認証サーバを利用でき，また，1 つの認証サービスの管理者権限が奪われたとしても，閾値の条件を満たすことができず，アカウントが悪用できない．

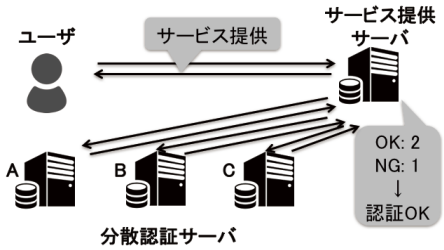

図 2　閾値型認証

3.2 提 案 手 法

本章では，閾値型認証のプロトコルについて説明する．閾値型認証の提案手法について図 3 を用いて説明する．

準備

- 認証サービス提供機関は複数の認証サービス（分散認証サーバ）を準備し，一定数の分散認証サーバから認証成功結果が返ればログイン成功とみなす．
- 認証成功結果の一定数を何個以上とするかは認証機関が決定し，事前にサービス提供者に周知しておくものとする．

手順

① ユーザはサービス提供サーバにサービス利用要求を送る．サービス提供サーバは利用できる分散認証サーバのリストを提示し，ユーザは使用したい分散認証サーバを複数指定する．

② サービス提供サーバはユーザに認証要求を渡し，指定した分散認証サーバへリダイレクトする．ユーザは認証要求を提示し，分散認証サーバで認証を行う．分散認証サーバは認証応答を作成し，ユーザをサービス提供サーバへリダイレクトする．

③ サービス提供サーバはユーザが指定した分散認証サーバに対して手順 ② をそれぞれ行い，分散認証サーバから一定数以上の認証成功の結果が返ってきた場合，ログイン成功とみなす．

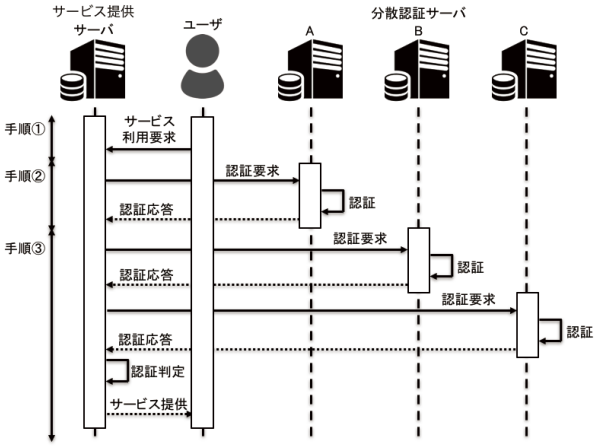

図 3　閾値型認証プロトコル

4. 属性情報の秘密分散管理

本章では，属性情報の秘密分散管理のプロトコルについて議論する．まず，属性情報の秘密分散管理の概要について述べ，次に，プロトコルを設計する上での要件を示し，提案手法について説明する．

4.1 概　　要

属性情報の秘密分散管理の概要を図 4 に示す．現在の Shibboleth モデルでは，認証サーバの管理者権限を奪取されると，認証サービスに登録されている全ユーザの属性情報が漏洩の恐れがある．また，認証サービスの冗長化を行うと，攻撃者はいずれかのシステムの脆弱性を突けば良いため，管理者権限を奪取されやすくなり，属性情報の漏洩のリスクが高まる．提案手法では複数の認証サービスを用意し，各分散認証サーバにユーザの属性値を秘密分散して登録しておく．各分散認証サーバのログインに成功すると秘密分散された属性値が手に入るとし，ユーザは秘密分散の閾値以上の認証サービスにログインすることで，属性値を復元することができる．このようにすれば，閾値未満の認証サービスの管理者権限が奪取されたとしても，属性値は秘密分散で暗号化されているので漏洩しない．また，閾値以上であればどの認証サービスを利用しても良いため，認証サービスの冗長化もできている．

4.2 要 件 定 義

属性情報の秘密分散管理プロトコルの要件定義とそれを満たすアイデアについて以下に記す．

- 属性情報の更新，削除が任意のタイミングで可能である

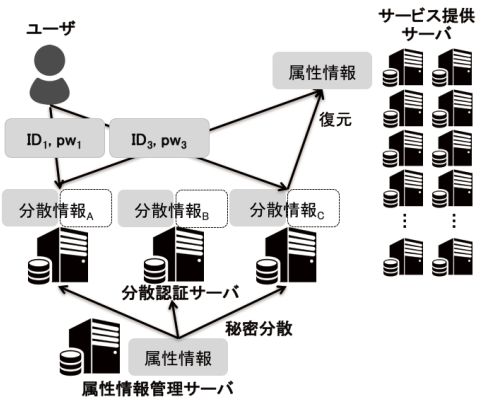

図 4　属性情報の秘密分散管理

こと

- 分散認証サーバには他の分散認証サーバの同じユーザの手がかりとなる情報を置かないこと

属性情報を厳密に管理する場合，属性情報が変更，もしくは失われた時に，管理者はすぐに更新，もしくは削除を行わなければならない．また，別々の分散認証サーバであっても同じユーザである手がかりを置いてしまうと，属性値を復元されてしまう恐れがあり望ましくない．

提案手法では，属性情報管理サーバが各ユーザの分散認証サーバごとにランダムな ID を振る．ユーザのデータを更新したい場合， ID と属性名を指定することで一意に指定することが可能で，同じユーザのデータかは見分けがつかない．

- 属性情報管理サーバで付与した属性値がサービス提供サーバに改ざんされずに届くこと
- 閾値未満の分散認証サーバの管理者権限が奪取されたとしても属性値の偽造ができないこと

Shibboleth モデルではユーザが認証サーバから受け取った認証応答をサービス提供サーバに提出する．したがって，悪意のあるユーザは認証応答を偽造し，属性値を改ざんする可能性がある．また，悪意のあるユーザが閾値未満の分散認証サーバの管理者権限を奪取し，自分の属性値を偽造する恐れがある．

提案手法では，属性情報管理サーバが属性値を秘密分散し，分散値に対して署名をつけることで属性値の偽造を防ぐ．

- 更新時に属性情報の整合性が崩れないこと

属性情報を更新する際に，障害などの原因で一部の分散認証サーバの更新が上手くいかなかった場合，データの不整合が起こり，認証サービスが利用できなくなる恐れがあり，これを防ぐ仕組みが必要である．

提案手法は，各分散認証サーバで属性情報を秘密分散した値を管理しているため，本システムを 1 つの大きなシステムとしてみた場合，分散データベースの構造をしていると考えられる．したがって提案手法では，分散データベースのデータ登録，更新，削除時に利用するプロトコルとしてよく知られている Two-phase commit protocol [11], [12] を用いる．

4.3 提 案 手 法

属性情報の秘密分散管理についての提案手法を属性情報の登録，更新，削除時と利用時に分けて以下に記す．なお，登録，更新，削除時は図 5 に，利用時は図 6 に対応している．

4.3.1 登録，更新，削除時

準備

- アカウントを新規に登録する際については，分散認証サーバごとにランダムな ID を作成しておき，属性情報管理サーバはこれのマッピングを知っておく．

手順

① （登録，更新時の場合）属性情報管理サーバは，対象の属性値について秘密分散を行い，分散した値についてそれぞれ署名をつける．

② 属性情報管理サーバは，Two-phase commit protocol を用いて属性情報の登録，変更，削除を行う．

②-1 属性情報管理サーバは，各分散認証サーバに ID ，対象の属性名，（登録，更新時の場合は）分散済の属性情報とそれに対する署名を送り， commit できるか問い合わせる．

②-2 各分散認証サーバは， commit できるかどうか確認し，属性情報管理サーバに応答する．

②-3 属性情報管理サーバは，全ての応答が commit 可能であった場合，各分散認証サーバへ commit を指示し，それ以外は abort を指示する．

②-4 各分散認証サーバは，手順 ②-3 の指示に従って commit / abort をする．（commit の指示で） commit できた場合，分散認証サーバは commit が完了したことを属性情報管理サーバへ伝える．

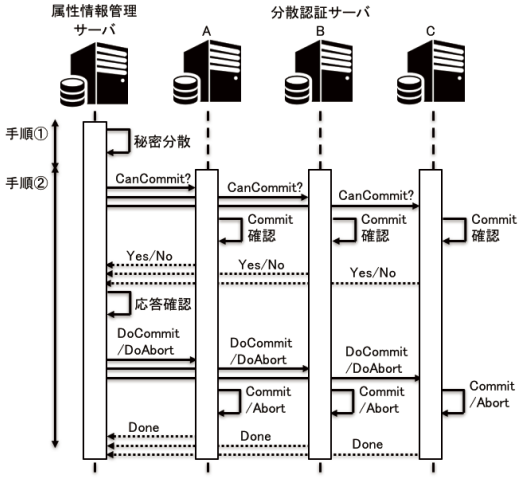

図 5　属性情報の登録，更新，削除時

4.3.2 利　用　時

準備

- サービス提供サーバは各分散認証サーバと属性情報管理サーバの公開鍵を持つ．

手順

① ユーザはサービス提供サーバにサービス利用要求をする．サービス提供サーバは分散認証サーバのリストを提示し，ユーザはその中から（閾値以上の）分散認証サーバを指定する．

② サービス提供サーバは認証要求を作成，ユーザへ渡し，ユー

ザを指定した分散認証サーバへリダイレクトする．
③　ユーザは認証要求を分散認証サーバへ提示し，分散認証サーバで認証を行う．認証に成功した場合，分散認証サーバは，認証要求を元にして認証応答を作成する．
④　分散認証サーバは，認証応答をユーザへ渡しサービス提供サーバへリダイレクトする．サービス提供サーバは認証応答を保持し，まだリダイレクトしていないサーバがある場合，手順 ② に戻り，全ての分散認証サーバへリダイレクト済みの場合は手順 ⑤ へ進む．
⑤　サービス提供サーバは各認証応答の分散認証サーバの署名と各分散属性値に対する属性情報管理サーバの署名を検証する．検証に合格したならば，秘密分散値からユーザの属性情報を復元し，ユーザにサービスを提供する．

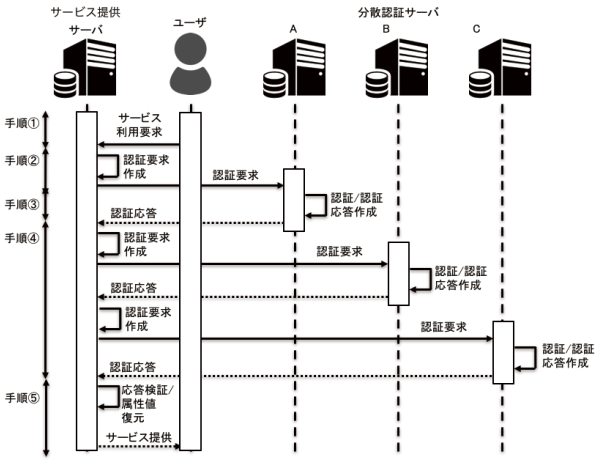

図 6　属性情報の利用時

5. 考　　察

提案手法についての考察を属性情報の登録，更新，削除時と利用時に分けて以下に記す．

5.1　登録，更新，削除時

5.1.1　各サーバ間の通信制御

各サーバ間でどのような通信制御を行うべきか，またそれを行うためにどのような準備が必要か考察する．

属性情報管理サーバについては，ユーザの機微な情報が入っているため外部から保護する必要がある．プロトコル中では，必ず属性情報管理サーバから通信を始めるので，ファイヤーウォールで動的フィルタリングを行う案が挙げられる．また，属性値登録時に偽の分散認証サーバと通信しないように，各分散認証サーバの公開鍵を知っておき，サーバ証明書を検証できる必要がある．

各分散認証サーバについては，偽の属性情報を登録，または不当に属性値を削除されないように．属性情報管理サーバの公開鍵を知っておき，サーバ証明書を検証できる必要がある．

5.1.2　属性情報更新時の分散認証サーバの状態

分散認証サーバが落ちている時に更新できるか，更新する時は全てのサーバを更新しなければならないかについて考察する．分散認証サーバが落ちている時に更新する場合の例として，3 個の分散認証サーバで閾値が 2 であり，1 個サーバが落ちている時に更新したことを想定する．この場合，更新した 2 個のサーバのうち 1 つが障害などで利用できなくなってしまった場合，残っているサーバは新しいデータと古いデータとなるため，連携サービスが利用できなくなり，目的としている障害への耐性が備わらない．ただし，属性情報の削除に関しては属性情報を使えないようにすれば良いため，閾値未満のサーバしか落ちていないのであれば，特に考慮せずに行って良い．

これらの議論点は分散データベースの議論点と一致すると考えられる．属性値の整合性が崩れないようにするためには，all Commit or Abort が原則ではあるが，部分 Commit を認めるならばシステムがそれを吸収できる仕組みが必要である．例えば，落ちているサーバ名と更新対象だった ID，属性値を属性情報管理サーバで保持しておき，サーバ復旧時に順次更新を行うなどである．

提案手法では，属性情報の不整合で利用できないということを防ぐために，Two-phase commit protocol を用いて原則として全てのサーバを更新することを想定している．分散認証サーバが落ちている場合は Two-phase commit protocol が abort し，管理者が検知することができる．したがって，更新，削除の緊急度に応じて，サーバ復旧まで待つか，管理者が上記のようなアイデアを用いて部分更新，削除を行うか判断して解決することとする．

5.1.3　属性情報分散時の属性値の形式

属性情報管理サーバが，分散認証サーバにどのような形式で属性値を預ければ良いか考察する．

まず，閾値未満の分散認証サーバの管理者権限を奪取した悪意のあるユーザについて考える．管理者権限を奪取したサーバは任意の偽造分散値に正当な署名をつけることが可能であるので，悪意のあるユーザは秘密分散の復元値，つまり自身の属性値を任意に偽造することが可能である．この場合，分散した属性値が改ざんされたかが検出できないことが問題なので，属性情報管理サーバで分散した属性値に対して属性情報管理サーバの署名を付けることでこの問題を防ぐ．

次に，サービス提供サーバでのプライバシー保護について考える．ユーザの属性値が同一でも秘密分散値がユーザ毎に異なる場合，サービス提供サーバにおいて，分散属性値をカウンティングすることによって，あるユーザがいつ，何回利用したかなどの情報がサービス提供サーバに伝わってしまう．一方で，属性値が同一であれば秘密分散値も同じであるとすると，悪意のある攻撃者に分散認証サーバが乗っ取られた場合，頻度分析から属性値が漏えいしてしまう恐れが残る．したがって，各属性値に対して同一の属性を持つ人数を調査し，最も少ない人数にあわせてその他の属性値の人のグループ分けを行い，グループ毎に秘密分散のさせ方を変化させる．このようにすれば，どの属性値のグループも同じ人数であり，頻度分析ができず，サービス提供サーバでのマッピングも起こらない．

5.1.4　秘密分散時のシェアの数，閾値

秘密分散時のシェアの数，閾値について妥当な値があるか考察する．シェアの数，閾値については，認証サービス提供者，

ユーザそれぞれに要望があると考えられる．認証サービス提供者にとっては，閾値が高いほどユーザの属性情報は漏洩しにくくなる．一方で，シェアの数を多くすると，外部に数多くのシステムを持たなくてはならなくなりコストがかかる．ユーザにとっても，閾値が高ければ自身の情報が漏洩しにくいことはメリットであるが，利用時に数多くの認証を行わなければならず手間がかかる．このように，安全性と利便性がトレードオフの関係となっているため，お互いが納得する閾値を決める必要がある．

提案手法では今後，実装，実験を行った上で，ユーザの心理と安全性を評価し，妥当な値を指し示したい．

5.2 利 用 時

5.2.1 属性情報の復元場所

秘密分散させた属性情報はいずれかの場所で計算を行い，復元する必要がある．分散させた属性情報の復元場所として考えられるのは，属性情報利用時に関わる組織のいずれかであり，分散認証サーバ，ユーザ，サービス提供サーバのいずれかである．

分散認証サーバを属性情報の復元場所として設定すると，復元場所の分散認証サーバの管理者権限を奪取された場合，閾値未満にも関わらず属性情報が漏洩してしまう．提案手法の要件を満たせないことから，本プロトコルでは利用できない．

ユーザで行う場合，ユーザが属性情報を偽造する恐れがあるので，復元した値の正当性を保証する仕組みを導入する必要がある．例えば，復元した属性値が，<属性値+有効期限+署名>となるようにすれば，正当性を保証することができるが，属性の資格を剥奪された後も有効期限内であれば使用できたり，復元値が漏洩した場合，他人が属性値を使用したりする恐れが出てくる．これらの問題を防ぐ 1 例としては，復元値を<属性値+ nonce +有効期限+署名>の形式にして nonce で失効処理を行う．しかし， nonce はユーザごとに一定であるので，サービス提供サーバにマッピングを許し，ユーザがサービスを何回利用したかの情報が把握できてしまうので，プライバシー保護上の課題が解決できない．

したがって，提案手法ではサービス提供サーバで属性値の復元を行う．サービス提供サーバで行うメリットとしては，秘密分散時の値を保証すれば良いため，分散認証サーバが信頼できればユーザの改ざんを疑わなくて良い．また， Shibboleth モデルではユーザがブラウザを用いて認証を行うので，ブラウザに複雑な処理を要求しなくて済む点が挙げられる．デメリットとしては，全利用者の復元計算を行わなければならず，サービス提供サーバに負荷がかかる点がある．

6. お わ り に

本稿では，属性情報の秘密分散管理について，要件定義とそれを満たすアイデアについて述べた上で，プロトコルの詳細な検討を行い，考察を行った．今後は，設計したプロトコルを元に，実装と評価を行う予定である．

文 献

[1] S. Cantor, I.J. Kemp, N.R. Philpott, and E. Maler, “Assertions and protocols for the oasis security assertion markup language,” OASIS Standard (March 2005), pp.1–86, 2005.

[2] D. Recordon and D. Reed, “OpenID 2.0: a platform for user-centric identity management,” the second ACM workshop on Digital identity management, pp.11–15, 2006.

[3] S. Cantor, “Shibboleth Architecture, Protocols and Profiles,” Internet2-MACE, 10 September 2005, pp.1–19, 2005.

[4] 伊藤友浩，岡部寿男，“複数の IdP を用いたシングルサインオンの提案と実装，” 電子情報通信学会 2016 年総合大会予稿集，vol.2016，p.130，2016．

[5] 伊藤友浩，岡部寿男，“属性情報を秘密分散した閾値型認証システムの検討，” 電子情報通信学会ソサイエティ大会講演論文集，vol.2016，no.2，p.352，2016．

[6] R.M. Van Rijswijk and J. Van Dijk, “tiqr: a novel take on two-factor authentication,” USENIX LISA’11, pp.81–97, 2011.

[7] D. Nuñez and I. Agudo, “BlindIdM: A privacy-preserving approach for identity management as a service,” International Journal of Information Security, vol.13, no.2, pp.199–215, 2014.

[8] 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コンピュータセキュリティ（CSEC），vol.2005，no.70，pp.321–328，2005．

[9] 松本 勉, 四方順司, 堀 正義, 大野一樹, 塩田明弘, “分散属性認証方式の実装と評価，” Proc. of Computer Security Symposium 2006 (CSS2006)，vol.2006，no.11，pp.489–494，2006．

[10] A. Shamir, “How to share a secret,” Communications of the ACM, vol.22, no.11, pp.612–613, 1979.

[11] J.N. Gray, “Notes on data base operating systems,” Operating Systems: An Advanced Course, eds. by R. Bayer, R.M. Graham, and G. Seegmüller, pp.393–481, Springer, 1978.

[12] B.W. Lampson and H.E. Sturgis, “Crash recovery in a distributed data storage system,” Xerox Palo Alto Research Center Palo Alto, California, pp.1–28, 1979.