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

2006－ＱＡＩ－21（１）
2006／10／23

プライバシー保護に配慮したShibbolethにおける属性交換の拡張

藤原翔一朗† 古村 隆明†† 岡部 寿男††

† 京都大学 情報学研究科
†† 京都大学 学術情報メディアセンター
E-mail: †{fuji-sho,emon}@net.ist.i.kyoto-u.ac.jp, ††okabe@i.kyoto-u.ac.jp

あらまし 様々なインターネット上のサービスを属性交換等のアイデンティティ連携技術を用いてシームレスかつ安全に利用するための枠組みとして，SAMLやその拡張であるShibbolethが開発されている．ところが，これらの枠組みで規定されている属性交換の方式ではユーザの属性とその値を直接交換するようになっており，ユーザの属性が必要とされる条件を満たすかどうかという情報を直接交換できない．ゆえに，必要以上に詳細な属性情報をサイト間で交換しざるを得ない状況が属性交換の過程で生じうる．そこで，本稿ではShibbolethに注目し，そのような条件情報を交換するための属性交換プロトコルの拡張を提案する．

キーワード Shibboleth, SAML, Liberty, XACML, プライバシー，属性交換

A Privacy Oriented Extension of Attribute Exchange in Shibboleth

Shoichirou FUJIWARA†, Takaaki KOMURA††, and Yasuo OKABE††

† Graduate School of Informatics, Kyoto University
†† Academic Center for Computing and Media Studies, Kyoto University
E-mail: †{fuji-sho,emon}@net.ist.i.kyoto-u.ac.jp, ††okabe@i.kyoto-u.ac.jp

Abstract Frameworks like SAML or Shibboleth have been developed so that various internet services can be used more sucurely and seamlessly. In these frameworks, however, a user must exchange one's attributes by immidiate value between one's identity provider and sevice providers. Without any extensions, they cannot directly communicate conditions about users' attributes that should be satisfied to authorize them. Consequently, there are cases where users must present detailed information that service providers don't actually require. In this paper, we discuss the issue. We focus on Shibboleth and propose an extension of the attribute exchange protocol of Shibboleth, that will solve the issue.

Key words Shibboleth, SAML, Liberty, XACML, Privacy, Attribute Exchange

1. まえがき

近年，WWW関連技術を利用して提供されるサービス(以下，WWWサービスと呼ぶ)の普及はめざましく，ネットショッピングや大学・企業内の個人用コンテンツポータルなど多岐に渡っている．これらWWWサービスは，様々な利便性を我々に提供する一方で，提供されるサービスの質・量の多様化はアカウント，属性情報の管理という負担をユーザに強いている．加えて，個人情報保護法[8]の施行等を背景としてWWWサービスで利用されるこれらの情報の適切な管理に対する重要性は増している．連携アイデンティティ(Federated Identity)とは，各WWWサービスに登録しているアカウント，属性情報を連携させシングルサインオン(Single Sign On, SSO)や組織間での属性交換(Attribute Exchange)を実現する技術であり，アカウント，属性情報管理にまつわるこれらの問題を解決する機能として期待されている．現在，この機能を提供する枠組みとして，SAML[1]，Liberty[2]，Shibboleth[3]，WS-Federation[4]等が開発されている．これらの枠組みはインターネットをインフラとして利用するのでSSL/TLS[5]やXML署名[6]・暗号[7]といったセキュリティ技術によって通信が保護されている．しかしながら，通信で交換された情報がその後どう扱われるかは情報を受け取ったエンティティ(entity)に委ねられている．交換された情報に関するセキュリティやプライバシーの保護が適切に行われているかについては当該エンティティを信頼するしかない．それゆえ，交換される情報が安全に管理されることも大事であるが，それと同様に必要以上の情報をエンティティ間で交換しないことも大事である．

本稿では，WWWサービス連携の枠組みとしてShibboleth

に注目し，必要以上の属性情報を交換しないという観点においてプライバシー保護に配慮した属性交換の拡張を提案する．まず，2. で Shibboleth とその属性交換方式を紹介する．そして，3. では既存の Shibboleth の属性交換が抱える問題を示し，4. でその問題に対する解決策を考察する．その上で，5. において Shibboleth における属性交換の拡張を提案する．最後に，6. で本稿のまとめと関連仕様・研究との比較，今後の展望を述べる．

2. Shibboleth と属性交換

2.1 Shibboleth

Shibboleth は，Internet2/MACE [9] におけるプロジェクトの一つであり，アクセス制御下の Web リソースを組織間で共有するオープンソースなミドルウェアの開発を目標としている．そのために，同プロジェクトは SAML ベースの枠組み及びプロジェクト名と同じ名前を持つ実装を開発し，提供している．実装の最新版は 1.3 で，SAML 1.1 がベースである．そして現在 SAML 2.0 をベースにした Shibboleth 2.0 が開発中である．以下，特に断らない限り Shibboleth とは Shibboleth 1.3 に関連する枠組み及び仕様のことを指すものとする．

Shibboleth のアーキテクチャは，WWW サービスを利用するユーザ，アイデンティティプロバイダ(Identity Provier，以下 IdP)，サービスプロバイダ(Service Provider，以下 SP) および **WAYF** サービス(Where Are You From Service) という 4 種類のエンティティから構成される．IdP はユーザの属性情報に責任を持ち，所属ユーザの本人性を検証する認証(authentication) を行う．SP はユーザが利用を試みる資源をホストし，その資源アクセスを許可する認可(authorization) を担当する．IdP はユーザの属性情報を管理するサービスを提供しているという点で SP でもあるといえる．WAYF サービスはユーザが所属する IdP を SP に知らせる支援をする．WAYF サービスについては，これ以外の機能は基本的に持たないので本稿ではこのエンティティの介在は考える必要がない．よって，本稿で考える Shibboleth のアーキテクチャは図 1 のようになる．

Shibboleth は IdP と SP 間のメッセージ交換プロトコルを規定し，認証，認可の方法は IdP と SP のローカルに委ねている．メッセージ交換プロトコルではユーザのセキュリティ情報を言明するアサーション(assertion) が交換される．Shibboleth が規定しているアサーションは，認証情報，属性情報，認可情報をそれぞれ言明する認証アサーション(Authentication Assertion)，属性アサーション(Attribute Assertion)，認可決定(注1)アサーション(Authorization Decision Assertion) の 3 種類である．アサーションを含む Shibboleth のプロトコルメッセージは SOAP [10] 等のプロトコルを用いて通信される．

Shibboleth における SSO の流れは以下のようになる (図 2)．

（1） ユーザはブラウザを通して SP が管理する資源へのアクセスを試みる．

（2） SP はブラウザとのセキュリティコンテキストを持っていないので，ブラウザをリダイレクトするなどの方法を用いて IdP へ認証アサーションを要求する．

（3） SP からの要求を受け取った IdP は，ユーザの認証を済ませていなければ認証を行い，認証方法，時間，ユーザ識別子などを含んだ認証アサーションを作成して何らかの方法で SP へそれを渡す．

（4） SP は認証アサーションの内容に従いユーザが適切に認証されているかを検証してユーザが要求した資源に対するアクセスの認可決定を行う．

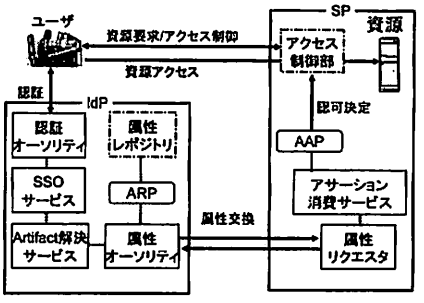

図 1 Shibboleth アーキテクチャ

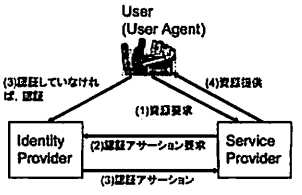

図 2 Shibboleth SSO

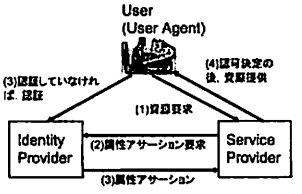

図 3 Shibboleth における属性交換

2.2 Shibboleth における属性交換

ユーザの匿名性を保つために，IdP は SP に対してセッション単位のランダムなユーザハンドルをユーザの識別子として提供する．このハンドルは，ユーザの実際のアイデンティティに関する情報を (ユーザの所属 IdP 以外には) 含まない．ゆえに SP は認可決定の材料として IdP からユーザの属性情報を得て，これを用いる．属性情報は IdP と SP が同様に解釈できるものでなければならない．普遍的な属性セットとして，[11], [12] 等が用意されている．Shibboleth における属性交換の仕様は SAML のそれと同じである．以下，ユーザと IdP，SP の間にすでに認証がなされているとして，Shibboleth における属性交換の流れを説明する (図 3)．もし認証がなされていない場合には，属性アサーションと共に認証アサーションを交換すれば

(注1)：認可をするかどうかの判断を認可決定 (authorization decision) と呼ぶ

よい．

（1） ユーザはブラウザを通して SP が管理する資源へのアクセスを試みる．

（2） SP はユーザがその資源への認可に必要な属性を持っているかどうか分からないので，ブラウザを経由するか直接 IdP と SP 間での通信を行うかして SP から IdP へ属性を要求する．

（3） IdP は SP から要求された属性を開放しても良いかどうかをビルトインの属性開放ポリシー(Attribute Release Policy, ARP) 等を利用して確かめ，開放しても良いならば属性 (と属性値) を属性アサーションの形式にして SP に渡し，開放できないならばエラーメッセージを SP に渡す．ユーザの認証がなされていない場合はそれを行う．

（4） SP は IdP から受け取ったメッセージから属性を得て，ビルトインの属性受理ポリシー(Attribute Acceptance Policy, AAP) 等を利用した認可決定を行う．さらにはその属性を認可決定以外の処理に用いるかもしれない．

3. Shibboleth 属性交換における問題

Shibboleth の枠組みの下 IdP と SP 間で交換される属性情報は多かれ少なかれユーザのアイデンティティに関連する情報である．そのためこの属性情報が (利用後は破棄されるように規定されているような場合にも関わらず) 蓄積されると属性の保持者を特定されうる．ユーザの匿名性 [13], [14] を可能な限り守るためにも IdP は必要以上に詳細なユーザの属性情報を公開すべきではない．例えば，“A が 20 歳以上かどうか” という情報が必要とされているのに対して “A が 24 歳である” という情報を提供するのは必要以上に詳細な情報の提供にあたる．

しかるに，現状の Shibboleth および SAML の枠組みにおいては属性値の直接交換以外の方法によって属性情報を IdP と SP 間で交換するためには何らかの仕様拡張が必要となる．実際，属性交換を規定する<shib:AttributeQuery>(注2) の仕様において “20 歳以上” 等の属性が満たすべき条件を記述するフィールドは規定されていなく，IdP から SP へ渡されるレスポンスにおいてもそのような情報を記述するフィールドは存在しない．このような属性値の直接交換で交換された情報は，認可に不必要な情報を含む場合がある．

例えば，K 大学は自身の IdP を持ち，K 大学の学生は，ID とクレデンシャルとは別に，IdP に対して「所属学部・研究科」と「身分」の 2 つの属性を登録しているものとする．そして，K 大学は企業 C が提供するオンラインジャーナルをライセンス契約して情報学研究科の 4 回生以上の学生及び工学研究科の修士以上の学生に使用を許可するとする．ここでは，オンラインジャーナルをホストしているプロバイダが SP である．この状況で，企業 C がオンラインジャーナルに対するユーザアクセスの認可決定を適切に行うためには IdP から属性情報を得る必要があるが，現状の Shibboleth の仕様では，K 大学 IdP と C 社の SP 間での属性交換においては両属性をその値ごと渡さねばならない．すなわち，K 大学の IdP に「所属学部・研究科」に情報学研究科，「身分」に修士課程 2 回生という属性値を登録している学生 F が C 社提供のオンラインジャーナルを利用するには属性交換を通して自分が情報学研究科修士課程 2 回生ということを C 社の SP に伝えなければならない．この “情報学研究科修士課程 2 回生” という情報は，求められている “情報学研究科の 4 回生以上の学生か，工学研究科の修士以上の学生” という情報に比べて必要以上に詳細である．

4. 属性交換の拡張に関する考察

前節で挙げた IdP と SP 間における属性交換の問題に対処するためには，属性がある条件をみたすかどうかという情報を交換できるように Shibboleth を拡張する必要がある．

Shibboleth と同様に SAML を拡張した Liberty においても [18] の “4.1.1” において上述の問題に対処が可能であるような属性交換の拡張が提案されている．拡張の概要は，SP が IdP に対して属性に関するテストを要求し，SP がそのテスト結果として true または false を返すというものである．この方法ならば，例えばあるユーザ A が 20 歳以上でなければ閲覧できない SP の資源を要求した場合に “A が 20 歳以上” か確認するテストを SP から IdP に対して要求し，IdP が SP に対して true を返すことで A の実際の年齢を SP に伝えることなく認可決定を行うことができる．ただし，テストの内容を記述する方法は仕様外であり，このテストを利用するにはその仕様外のテスト記述方法を IdP と SP で共有しておかねばならない．上記の例ならば “A が 20 歳以上” という記述はユーザ A の年齢が 20 歳以上であるという言明を意味するという共通の理解を SP と IdP 間で持っておかねばならない．IdP と SP 双方が XACML [16] 等の高機能なポリシー言語を理解できる場合ならば，[17] のように，そのポリシー言語でテストを記述しそれを交換するよう拡張すればよい．しかしながら，一般的に現実の IdP と SP は必ずしもそういう状況にはない．また，テスト結果として true か false の 2 値しか与えられないという仕様は何らかの理由でテスト結果を公開できない場合，どちらかの値にその意味を含めるか，テストに直接関係しない部分でエラーと処理しなければならない．

5. 条件提示を通じた Shibboleth 属性交換

我々は，前節の考察を踏まえ Shibboleth における IdP と SP 間での属性交換の拡張を提案する．本拡張は，従来の属性交換に加えて以下のような属性情報交換を可能にする．

> 認可を要する資源に対してある IdP に属するユーザがアクセスを試みると SP はユーザを認可するために必要な属性に関する条件を IdP に渡す．IdP は該当ユーザがその提示された条件を満たしているかを判定し，“満たす” を意味する “true”，“満たさない” を意味す

(注2)：尚，本稿では以下の XML 名前空間プレフィクスを用いる

- xsd: XML Schema([15]) の名前空間．
- shib: Shibboleth の仕様であることを示す名前空間．
- ext: 本稿における拡張であることを示す名前空間．

る “false”，“回答不可” を意味する “unanswerable” に相当するメッセージを SP へ返す．SP はそのメッセージを用いて認可決定を行う．

本拡張を利用した属性交換のシナリオは以下のようになる．

（1） ユーザはブラウザを通して SP が管理する資源へのアクセスを試みる．

（2） SP はユーザがその資源への認可に値する属性及びその値を持っているかどうか分からないので，ブラウザを経由するか直接 IdP と SP 間での通信を行うかしてユーザが認可されるために必要な条件を IdP へ渡す．

（3） IdP は SP から受け取った条件に対して “true/false” を返すことができるかどうかを確かめる．IdP は，“true/false” を返すことが可能ならば条件を判断した結果を SP へ返し，不可能ならば “unanswerable” を SP に返す．

（4） SP は IdP から受け取ったメッセージに基づいて認可決定を行う．

本拡張においては基本的な条件記述言語を仕様で規定する．それによって IdP と SP 間で条件記述について特別な共通理解を得る必要無しに普遍的に本拡張を利用できるようになる．特別に定めた条件記述言語を利用するにあたっては従来どおりの特化した拡張を行う．

以下，我々は本拡張を次の観点から記述する．

- 拡張プロトコル
- プロトコル処理
- プロトコル周辺

5.1 拡張プロトコル

5.1.1 条件提示

本拡張が提供する機能は，将来の Shibboleth で標準に利用できるべきである．しかしながら，本稿における本拡張の設計では既存の Shibboleth との相互運用性を考えて Shibboleth の拡張ポイントを利用する．Shibboleth 1.3(すなわち SAML 1.1) では，要求を記述する<shib:Request>下の要素<shib:Query>が拡張ポイント(注3)であるのでこれを利用する．本拡張の要素<shib:Query>下に加える部分の XML スキーマ記述は図 4 のようになる．図 4 中の<ext:ConditionExpression>が提示条件である．要素<ext:Annotation>は提示条件その他に対する注釈でプログラムによる解釈はされない．要素<ext:Extension>は拡張フィールドであり，IdP と SP 間で特に定めた方式で記述された提示条件のやり取りなどに用いる．属性 ConditionId は条件の ID を示すものであり，提示条件とその返答の対応をとるために用いられる．

5.1.2 条件記述

条件の内容は<ext:ConditionExpression>以下に記述する．そこに記述する条件は，概念的には属性に対する言明 (例えば “年齢が 20 歳以上”) である述語 (predicate) を “AND”・“OR”・“NOT” といった論理関数 (logical function) で再帰的に結合することによって構成される (例えば “年齢が 20 歳以上 AND 性別が女性”)．

(注3)：SAML 2.0 ならば<shib:AttributeQuery>下の要素<shib:Extensions>

```
<xsd:element name="RequiredCondition"
                        type="RequiredConditonType"/>
<xsd:complexType name="RequiredConditionType">
    <xsd:element ref="ConditionExpression"/>
    <xsd:element name="Annotation" type="string"/>
    <xsd:element name="Extension" type="anyType"/>
    <xsd:attribute name="ConditionId" type="string"
                            use="required"/>
</xsd:complexType>

<xsd:element name="ConditionExpression"
                      type="ConditionExpressionType"/>
<xsd:complexType name="ConditionExpressionType">
  <xsd:element ref="Predicate"/>
</xsd:complexType>
```

図 4 <shib:Query>下に加える拡張のスキーマ

```
<ext:Predicate function="and">
  <ext:Predicate function="ge" border="20">
    <shib:AttributeDesignator AttributeName="年齢"
                    AttributeNamespace="hoge"/>
  </ext:Predicate>
  <ext:Predicate function="match" value="女性">
    <shib:AttributeDesignator AttributeName="性別"
                    AttributeNamespace="hoge"/>
  </ext:Predicate>
</ext:Predicate>
```

図 5 述語の例

```
<shib:Attribute AttributeName="ConditionResult"
                    AttributeNameSpace="hoge">
  <shib:AttributeValue>
    <ext:result ConditionID="id" reason="user-policy">
      "unanswered"</ext:result>
  </shib:AttributeValue>
</shib:Attribute>
```

図 6 条件提示に返答する属性の例

本拡張で定める条件記述言語に求められる条件記述能力は本質的には XACML における *Condition* に準ずるものである．Shibboleth と XACML は [19] にあるように相補的に利用可能であることを考慮すると，IdP への提示条件を XACML における表現に変換することが容易である方が望ましい．よって，本拡張における述語と論理関数は XACML の *predicate* 風に定義する．

図 5 は，ge という演算を用いて名前空間 hoge の “年齢” 属性の値と値 20 を比較した結果と match という演算を用いて名前空間 hoge の “性別” 属性の値と女性という値を比較した結果に対し，and という演算を用いてその評価を行う述語の例である．

5.1.3 提示条件への回答

提示された条件に対しては “true/false/unanswerable” を表

す属性を定義し，その属性に関するステートメントを含む属性アサーションを返す．“unanswerable” は，“true/false” の評価結果を返すことがプライバシーポリシー上の理由等により不可能な場合使用する．そして，“unanswerable” となった原因を示す属性も平行して定義，利用する．

例えば，ConditionID が id である条件の評価結果として，user-policy が原因で “unanswered” を返すことを示す属性 ConditionResult は図 6 のようになる．

5.1.4 Metadata

当然ながら，拡張プロトコルを利用するためには当事者の IdP と SP 双方がこのプロトコルを理解できなければならない．どのエンティティが拡張プロトコルを利用できるかという情報は metadata の要素<shib:Extensions>にその旨を記述することによって交換する．仕様外の演算を用いる場合にもここにその旨を記述する．

5.2 プロトコル処理

本拡張プロトコルの大まかな流れは本節の冒頭で述べた通りである．

SP は，条件提示と同時に通常の属性要求のメッセージを IdP に送ることが出来る．その場合，要求を処理する IdP は，条件評価の結果 “true” を返すことが出来る場合のみに通常の属性要求を処理する．IdP は，<shib:Request>下の拡張ポイントに適切な名前空間を持つ<ext:ConditionExpression>を発見すれば，それを処理する．要素<ext:ConditionExpression>の処理規則は，以下の通りである．

- 要素<ext:Predicate>で示される述語の評価結果は “true/false/unanswerable” のいずれかである．“unanswerable” の場合は，その原因も返す．
- 理解できない述語は “unanswerable” と評価され，その原因として述語が理解できない旨を返す．
- 要素<ext:ConditionExpression>の評価結果は，直下の<ext:Predicate>の評価結果と一致する．“unanswerable” の場合はその理由も適切な形式で返答に記述される．

5.3 プロトコル周辺

5.3.1 SP 側

本拡張では，Shibboleth ビルトインの AAP も拡張される．Shibboleth 仕様外の認可決定機構を用いる場合，本拡張プロトコルを利用するには仕様外の認可決定機構における認可条件の記述言語と本拡張の条件記述言語を相互に変換する機能が必要になる．そのような変換を行う一つの方法として，[19], [21] 等のように XSLT [20] を利用する方法が挙げられる．

5.3.2 IdP 側

IdP においては，提示条件の評価を行うコンポーネントが必要になるが，属性オーソリティを拡張してこのコンポーネントとしての機能を持たせる．

それに加えて提示された条件を評価した結果を返しても良いかどうかを判定・制御する機能が必要になる．我々は Shibboleth ビルトインの ARP を拡張することでこれに対応する．具体的には，SP ごとに評価してもよい述語を設定できるようにする．

SP 側と同様に，Shibboleth の仕様外の属性開放管理コンポーネントを用いる場合にはデータの整合性を提供するための措置が必要となる．

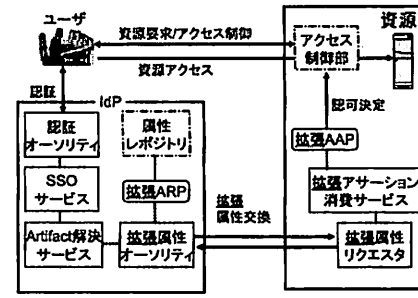

図 7 本拡張における Shibboleth アーキテクチャ

5.3.3 アーキテクチャ

本拡張下での Shibboleth アーキテクチャは図 7 のようになる (拡張が行われるコンポーネントの名前の先頭には “拡張” がついている)．

6. おわりに

本稿では，既存の Shibboleth の抱える属性交換に関する課題として，属性とその値を直接交換することしか出来ないという仕様を挙げた．そして，その仕様の問題への対処について既存の方法・研究を踏まえて考察をした．その上で，認可決定に必要な属性値の条件を属性要求の代わりに IdP へ送り，IdP は提示された条件に適合するか否か，あるいはその条件質問には答えられないという旨のレスポンスを SP に返すという属性交換の拡張を提案した．

本拡張は，XACML の<Condition>下に記述される情報を交換できるようにしたものといえる．また，本拡張は IdP から SP へ渡す情報を少なくするという点で IdP およびユーザ側の立場に立ったものであるが，[22] のような SP 側の立場に立ったプライバシー保護手法と本拡張を相互利用すれば Shibboleth のプライバシー保護機能をさらに強化することができると考えられる．

今後は本拡張の実装と平行して Delegation との兼ね合いや適切な APP/ARP ルールの拡張・設計手法に関する研究を行う予定である．

文 献

[1] “Security Assertion Markup Language SAML, OASIS Security Services (SAML) TC”. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.

[2] “Liberty Alliance Project”. http://www.projectliberty.org/.

[3] “Shibboleth Project”. http://shibboleth.internet2.edu/.

[4] “Web Services Federation Language”. http://www-128.ibm.com/developerworks/library/specification/ws-fed/.

[5] “The TLS Protocol Version 1.0, RFC 2246” (1999). http://www.ietf.org/rfc/rfc2246.txt.

[6] “XML Signature WG, W3C”. http://www.w3c.org/Signature/.

[7] “XML Encryption WG, W3C”. http://www.w3.org/Encryption/2001/.

[8] “個人情報の保護に関する法律”. http://law.e-

gov.go.jp/htmldata/H15/H15HO057.html.

[9] "MACE - Middleware Architecture Committee for Education". http://middleware.internet2.edu/MACE/.

[10] "SOAP:Simple Object Access Protocol". http://www.w3.org/TR/soap/.

[11] "eduPerson Object Class". http://www.educause.edu/eduperson/.

[12] "InCommon Federation: Common Identity Attributes". http://www.incommonfederation.org/docs/policies/federatedattributes.html.

[13] "Security and Privacy Considerations for the OASIS Security Assetion Markup Language (SAML) V2.0" (2005). http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf.

[14] A. Pfitzmann and M. Hansen: "Anonymity, Unlinkability, Pseudonymity, and Identity Management -A Consolidated Proposal for Terminology" (2005). http://marit.koehntopp.de/pub/anon/Anon_Terminology.pdf.

[15] "XML Schema, W3C". http://www.w3.org/XML/Schema.

[16] "eXtensible Access Control Markup Language (XACML)". http://www.oasis-open.org/committees/xacml/.

[17] "SAML 2.0 profile of XACML v2.0" (2005). http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf.

[18] "Liberty ID-WSF Data Services Template Version: 2.1-17" (2006). http://www.projectliberty.org/specs/draft-liberty-idwsf-dst-v2.1-17.pdf.

[19] W. Hommel: "Using xacml for privacy control in saml-based identity federations", LNCS 3677, pp. 160–169 (2005).

[20] "The Extensive Stylesheet Language Family (XSL), W3C". http://www.w3.org/Style/XSL/.

[21] W. Hommel and H. Reiser: "Federated identity management in business-to-business outsourcing", 12th Workshop of the HP OpenView University Association (2005).

[22] U. Mbanaso, G. Cooper, D. Chadwick and S. Proctor: "Privacy Preserving Trust Authorization Framewrok using XACML", IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2006), IEEE Computer Society, pp. 673–678 (2006).