Vol.2014-IOT-26 No.2
2014/6/28

Shibboleth IdPにおける LoAを考慮した認証方式グループ化機能の開発

河野 圭太[1,a)]　中村 素典[2]

概要：利用する情報サービスの保護レベルに応じて身元保証レベル（Level of Assurance：LoA）が異なる認証方式を使い分けることにより，安全かつ実用的な認証連携を実現できる．しかしながら，大学等の学術機関において多数の導入実績がある Shibboleth を用いてこれを実現するためには，Service Provider（SP）が連携先 Identity Provider（IdP）の認証方式（およびその LoA）全てを事前に把握しておく必要があり，現実的ではない．本研究では，Shibboleth IdP における認証方式のグループ化機能を開発することにより，この問題の解決を図る．

キーワード：Shibboleth，LoA，IdP，SAML，統合認証基盤

Development of Authentication Method Grouping Function Considering LoA on Shibboleth IdP

Keita Kawano[1,a)]　Motonori Nakamura[2]

Abstract: Using multiple authentication methods with different Levels of Assurance (LoAs) enables secure and practical federated authentication. Shibboleth, widely deployed at academic institutions like universities, has a practical issue, to realize this. Service Providers (SPs) are supposed to understand all the authentication methods (and their LoAs) on the correspondent Identity Providers (IdPs) beforehand. We develop an authentication method grouping function on Shibboleth IdP, to overcome this issue.

Keywords: Shibboleth, LoA, IdP, SAML, integrated authentication infrastructure

1. はじめに

近年，様々な分野において情報システムの利活用が進む中で，利用者が用いる情報サービスの数が増加している．そのため，多くの組織では，統合認証基盤を導入し，利用者の利便性とシステムの安全性を確保している [1], [2]．統合認証基盤を用いることで，利用者は，信頼できるいつものサーバによる一度の認証で，複数の情報サービスを利用（シングルサインオン）することができる．

学術界においては，認証フェデレーションと呼ばれる共同体の構築により，組織の枠を超えた情報サービスの相互・共同利用に関する取り組みも進んでおり，日本では，「学認」がその役割を担っている [3]．学認では，海外の主要な学術フェデレーションと同様に，Security Assertion Markup Language（SAML）に基づく組織間のシングルサインオン・属性交換が実現されており，技術的には，Internet2 が開発した Shibboleth が使用されている [4], [5]．そのため，大学等の学術機関において，学認による組織間連携との親和性を考慮し，Shibboleth を用いた統合認証基盤の構築事例が多数報告されている [6], [7], [8]．

このような認証連携が進むにつれて，統合認証基盤における安全性の重要度が増している．例えば，一般的な，ID とパスワードによる認証方式を採用した統合認証基盤においては，1 組の ID とパスワードの漏洩が，その利用者が利

[1] 岡山大学
Okayama University, Okayama 700–8530, Japan
[2] 国立情報学研究所
National Institute of Informatics, Tokyo 101–8430, Japan
[a)] keita@cc.okayama-u.ac.jp

用できる複数の情報サービスの不正利用につながるため，より安全性を考慮した運用が求められる [9]．一方で，安全性向上のために，強度の高い認証方式を採用することは，利用者の利便性低下につながりかねない．

このような背景を受けて，身元保証レベル（Level of Assurance：LoA）に基づき，利用する情報サービス（保有する情報資産）の保護レベルに応じた認証方式を選択する方法が確立されつつある [10], [11]．例えば，より重要度が低い情報資産しか保有しない情報サービスを利用する場合には，従来の ID とパスワードによる認証を許容する一方で，より重要度が高い情報資産を保有する情報サービスを利用する場合には，スマートカード認証や，バイオメトリクス認証等，強固な認証方式を要求する．これにより，利用者の利便性を著しく低下させることなく，必要な安全性を確保することができる．

しかしながら，上述のように大学等の学術機関において多く採用されている Shibboleth 認証サーバ（Identity Provider：IdP）では，LoA の考慮に対する十分な実装がなされておらず，実用上の課題が残されている．

とりわけ，Shibboleth IdP では，認証方式（認証コンテキスト）間の LoA を比較する機能が実装されていないため，LoA を考慮したシングルサインオン実現のためには，LoA を要求する情報サービス（Service Provider：SP）が，IdP で提供している該当 LoA 以上の認証方式全てを，認証要求中の許容認証方式リストに含める必要がある [12], [13]．組織間での認証連携を考慮した場合，IdP・SP 間で綿密な情報共有が求められるこの制限は，許容できるものではない．

そこで，本研究では，Shibboleth IdP において認証方式のグループ化を実現する機能を開発し，この問題の解決を図る．本機能により，SP 側では，LoA 等のグループ形式による認証方式の要求が可能になり，認証フェデレーションを伴う環境において，実用的な運用レベルで LoA を考慮した認証連携を実現できる．

以下，2 章では，Shibboleth および LoA の概要と，Shibboleth における LoA 対応の問題点について述べる．3 章では，2 章の問題点を解決するために本研究で開発する Shibboleth IdP における認証方式グループ化機能について述べる．最後に，4 章で，まとめと今後の課題を述べる．

2. Shibboleth と LoA

本章では，まず，2.1 節，2.2 節において，Shibboleth の概要および Shibboleth IdP における認証手順の概要を述べる．また，2.3 節において，LoA の概要を述べる．さらに，2.4 節において，Shibboleth を用いて LoA を考慮した認証連携を行う際の問題点を明らかにする．

2.1 Shibboleth

Shibboleth は，OASIS で策定された SAML 標準に基づき，組織間のシングルサインオン・属性交換を実現するための，オープンソースソフトウェアである [14]．

図 1 に，Shibboleth の動作原理を示す [*1][16]．図 1 は，利用者が，ある組織の SP を利用するために，自組織の IdP で認証を受け，認証結果を SP に提示する様子を示している．Shibboleth では，IdP，SP に加えて，利用者による自組織 IdP の発見を支援するための Discovery Service（DS）が用いられる．

まず，(1) 利用者が SP のリソースにアクセスしようとすると，(2) 利用者からの要求は，利用者のブラウザを介して DS へリダイレクトされる．(3) DS で利用者自身による IdP の選択が行われた後，(4) 利用者からの要求は，利用者のブラウザを介して SP へリダイレクトされる．(5) SP では選択された IdP に対する認証要求が生成され，利用者のブラウザを介して IdP へリダイレクトされる．

また，(6) IdP での認証に成功すると，(7) 認証結果と提供属性が含まれた認証応答が SP にリダイレクトされる．(8) SP での認可制御の結果を受けて，利用者が SP のリソースにアクセスできるようになる．なお，既に利用者が IdP で認証を受けている場合には，(6) の処理は省略される．

SAML 標準において，手順 (5) で生成される認証要求には，SP が許容できる認証方式（認証コンテキスト）のリストを含めることができる．これらは，認証要求（AuthnRequest）に含まれる RequestedAuthnContext 内の AuthnContextClassRef または AuthnContextDeclRef として，指定される [12]．また，SAML 標準においては，RequestedAuthnContext 内の Comparison を指定することにより，比較表現を用いた認証方式の要求が可能になる．

Shibboleth IdP の実装では，AuthnContextClassRef または AuthnContextDeclRef として，認証要求内で指定された文字列と，ログインハンドラ（IdP が利用者に提供す

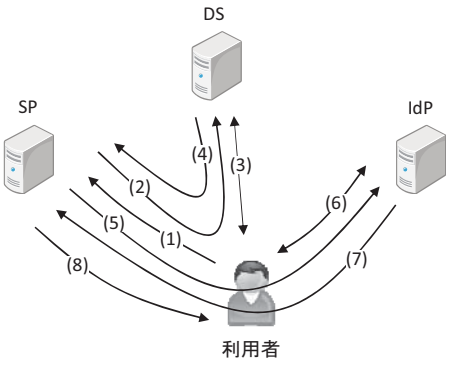

図 1 Shibboleth の動作原理.
Fig. 1 Operating principle of Shibboleth.

[*1] 本稿では，SAML V2.0 の Web Browser SSO Profile を前提とする [15].

る各種認証方式に対応するハンドラ）ごとに対応付けられた文字列を比較することにより，認証方式を同定する．SP が許容できない認証方式は，IdP の AuthenticationEngine において，フィルタリングされる．なお，Shibboleth IdP の実装では，Comparison に関しては，「exact」（厳密一致）のみが考慮されている．

2.2 Shibboleth IdP の認証処理

図 2 に，Shibboleth IdP における認証処理の概要を示す．図中の IdP では，ID・パスワード認証，クライアント証明書認証が実装されており，それぞれのログインハンドラは，PasswordProtectedTransport，TLSClient として定義されている [*2]．また，SP では，クライアント証明書認証，スマートカード認証が許容されており，認証要求中の RequestedAuthnContext に TLSClient, Smartcard を含めている．なお，図中の斜線部は，後述する LoginContext の読み書きが発生する処理を示している．

まず，(1) SP からのリダイレクトにより IdP に到達した認証要求は，(2) IdP の ProfileRequestDispatcherServlet を経て，SSOProfileHandler で処理される．ここでは，認証処理の状態を保持するための LoginContext が生成され，(3) 要求は AuthenticationEngine へとリダイレクトされる [*3]．このとき，認証要求に含まれる RequestedAuthnContext の内容が，LoginContext に保存される．

次に，(4) AuthenticationEngine において，LoginContext に保存された RequestedAuthnContext の内容に基づき，SP からの要求を満たさない認証方式がフィルタリングされる．図 2 の例では，クライアント証明書認証（TLSClient）のみが利用可能な認証方式となる．AuthenticationEngine では，フィルタリング後に残された認証方式から，実際に利用する認証方式を選択し，それに対応するログインハンドラ（ここでは，RemoteUserLoginHandler）へ処理を引き渡す．なお，該当の利用者が，フィルタリング後に残された認証方式のいずれかで既に認証を受けている場合，その認証結果を再利用（実際の認証処理を省略）できる [17]．

さらに，RemoteUserLoginHandler では，(5) RemoteUserAuthServlet へのリダイレクトが行われ，(6) クライアント証明書を用いた認証が行われる．要求は再度 AuthenticationEngine で処理され，実際に使用された認証方式が LoginContext に保存された SP からの要求を満たすかどうかを確認すると共に，その情報を LoginContext に保存する．また，AuthenticationEngine では，今後の再利用に備えて，使用された認証方式による認証の成功を記録しておく．

その後，(7) 再度 ProfileRequestDispatcherServlet にリダイレクトされた要求は，(8) SSOProfileHandler で処理される．ここでは，認証の後処理として LoginContext を破棄すると共に，LoginContext に保存された使用認証方式の情報（AuthnContext）を含む認証応答を作成する．(9) 作成された認証応答は，SP へとリダイレクトされる．

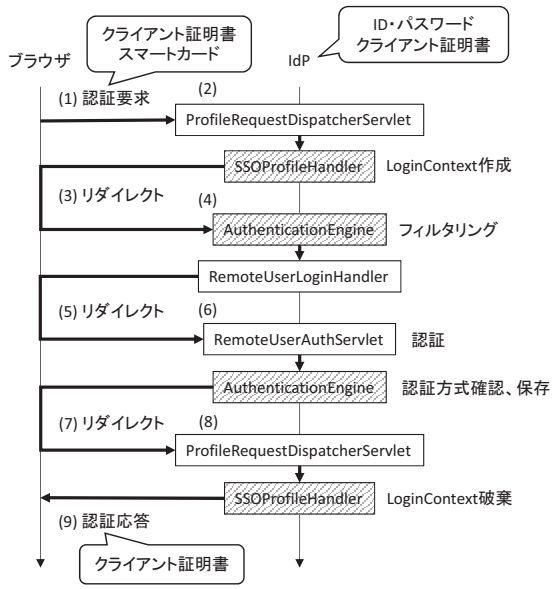

図 2 Shibboleth IdP の認証処理.
Fig. 2 Authentication process of Shibboleth IdP.

[*2] 実際には，urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport のような形式で定義される [13].
[*3] 以降の処理では，利用者と LoginContext の紐付けのため，Cookie が利用される．

2.3 Level of Assurance (LoA)

Shibboleth 等の活用により，SP は IdP を信用し，その認証結果を受けて，認可制御を実施する．このとき，SP の観点からは，IdP の認証結果がどの程度信頼できるかを把握できることが求められる．例えば，重要度が高い情報資産を保有する SP の管理者は，IdP からの認証結果に高い信頼性を要求する．

このような情報サービスの保護レベルを考慮した認証連携を実現するため，身元保証レベル（Level of Assurance：LoA）が用いられる [10], [11], [18]．認証結果の信頼性には，ID の作成方法，配布方法，認証メカニズムの強度等，複数の要素が影響するが，これらに対する要求事項を複数のレベルに分け，IdP・SP 間で共有する [*4]．例えば，ITU-T X.1254 においては，**表 1** のように 4 レベルの LoA が規定されている [10]．高レベルの LoA ほど，ID の作成方法，配布方法，認証メカニズムの強度等に，厳しい制約が課せられる．IdP では，SP が要求したレベル以上の LoA を有する認証方式を用いて，利用者の認証を行う．

なお，IdP・SP 間で，LoA 等のポリシーを個別に規定することもできるが，認証フェデレーションにおいては，信

[*4] 単純化のため，本節以外では，認証メカニズムの強度が LoA を支配すると仮定している．

表 1 ITU-T X.1254 における LoA の分類.
Table 1 Classification of LoAs in ITU-T X.1254.

Level	Description
1 - Low	Little or no confidence in the claimed or asserted identity
2 - Medium	Some confidence in the claimed or asserted identity
3 - High	High confidence in the claimed or asserted identity
4 - Very high	Very high confidence in the claimed or asserted identity

頼フレームワーク内で一律のポリシーを策定し，運用レベルを統一することが求められる [18], [19], [20].

2.4 Shibboleth における LoA の考慮とその問題点

2.1 節で示した RequestedAuthnContext を用いることにより，Shibboleth においても LoA を考慮した認証連携を実現できる．具体的には，SP が要求するレベル以上の LoA を有する認証方式を，RequestedAuthnContext 内の AuthnContextClassRef として IdP への認証要求に含めることで，これに対応する方法が考えられる．ここで，前述のとおり，Shibboleth IdP では，RequestedAuthnContext 内の Comparison を考慮した処理が実装されておらず，各種認証方式の優劣を指定する機能も存在していないため，該当する認証方式を全て列挙する必要がある．

しかしながら，このためには，SP が連携先の IdP で実装されている認証方式（およびその LoA）全てを事前に把握しておく必要があり，組織間での認証連携を前提とする認証フェデレーションでは現実的ではない．各種ログインハンドラに対応付ける AuthnContextClassRef や，SP が認証要求に含める AuthnContextClassRef を，LoA を用いた記述に改める方法も考えられるが，個別の認証方式による制限をかけたい場合や，連携先（個別の SP や信頼フレームワーク）ごとに LoA の分類が異なるような環境に対応できないなど，運用上の問題が発生する．

3. LoA を考慮した認証方式グループ化機能

2.4 節で明らかにした問題点を解決するため，本研究では，Shibboleth IdP における認証方式のグループ化機能を開発する．まず，3.1 節において，開発する認証方式グループ化機能の概要を述べる．また，3.2 節において，本機能の処理中に発生する形式変換のタイミングについて言及する．最後に，3.3 節において，本機能の実装について述べる．

3.1 認証方式グループ化機能の概要

2.4 節で述べたように，現状の Shibboleth IdP を用いて LoA を考慮した認証連携を実現するために，個別認証方式の形式により認証方式を要求する方法は，現実的ではない．

そこで，本研究では，Shibboleth IdP において，LoA の形式で要求された認証方式を，IdP で提供する個別認証方式へと変換する機能を開発し，この問題の解決を図る．ここでは，汎用性を考慮し，個別認証方式のグループ化を行い，グループ形式による認証方式の要求に対応する機能として，これを実装する．グループ化機能の実現により，LoA に基づかない，個別ポリシーによる認証方式群の指定や，連携先ごとの LoA 分類方法の違いに対応できる．例えば，自組織内ではスマートカード認証を組織内の LoA2 として扱う一方で，所属する認証フェデレーションに対しては認証フェデレーション内の LoA1 として扱う，などの対応が可能である．

図 3 に，従来の個別認証方式の形式による認証方式の要求方法と，本研究のグループ形式による認証方式の要求方法を示す．図 3 の例では，2 つの IdP（IdP1，IdP2）が，それぞれ ID・パスワード認証とクライアント証明書認証，ID・パスワード認証とスマートカード認証を，LoA1，LoA2 に対応する認証方式として提供している．また，2 つの SP（SP1，SP2）が，それぞれ LoA1，LoA2 による認証を要求している．

図 3 に示すように，従来の方法ではそれぞれの SP がそれぞれの IdP で提供されている認証方式の詳細を把握し，認証要求を出す必要がある一方で，本研究では認証フレームワーク等で事前に合意された LoA（グループ形式）による認証方式の要求が可能である．

図 4 に，本研究におけるシステム開発イメージを示す．まず，SP からリダイレクトされた認証要求に対して，LoA 等のグループ形式で指定された SP が要求する認証方式を，IdP で提供されている個別認証方式の形式へと変換する．また，認証要求との整合性を取る必要性から，認証応答についても，個別認証方式の形式からグループ形式への逆変換を実施する．

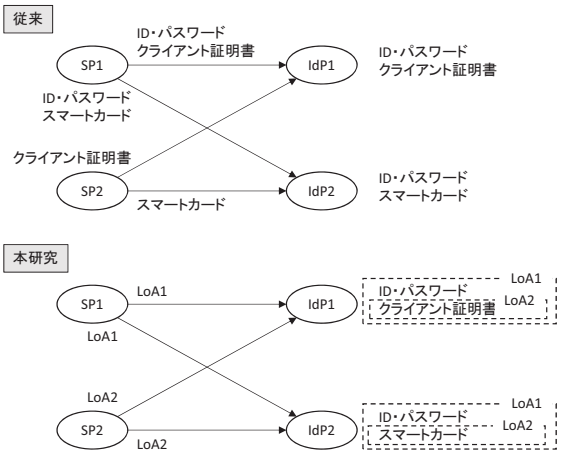

図 3 認証方式の要求方法.
Fig. 3 Request method for authentication method.

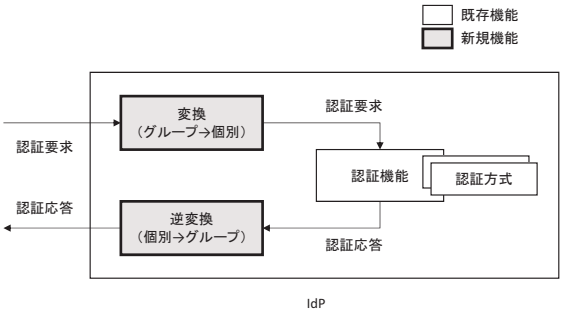

図 4 システム開発イメージ.
Fig. 4 System development image.

これにより，従来の機能によるシングルサインオンの実現等を保ったまま，認証方式のグループ化が可能になる．特に，従来の認証機能において，個別認証方式の形式による処理を保つことにより，複数のグループに所属する認証方式が存在する場合でも，厳密なシングルサインオンを提供できる．本研究では，既存の認証機能による処理が，2.2 節で示したように，LoginContext の内容に基づき実行される点に着目し，LoginContext の内容を書き換えることによって，これらの処理を実現する．

なお，本研究では，実運用開始後のメンテナンス性を考慮し，Tomcat のフィルタ機能を用いて，これらの機能を実装する．フィルタ機能の開発に際しては，Shibboleth IdP やその拡張機能のソースコードを参考にした [21], [22], [23].

3.2 形式変換のタイミング

本研究における認証方式グループ化機能の開発においては，LoginContext に保存された SP が要求する認証方式の形式変換タイミングが重要である．本機能を用いた場合にも，IdP・SP 間で正しく認証連携を継続するためには，適切なタイミングでフィルタ機能を実行し，形式を変換する必要がある．

まず，認証要求内の RequestedAuthnContext をグループ形式から個別認証方式に変換する機能については，2.2 節で示したように，IdP における認証方式のフィルタリングが AuthenticationEngine で実施されることから，この直前（2.2 節における，手順（3）と（4）の間）に実施する．

また，認証応答内の AuthnContext を個別認証方式からグループ形式に逆変換する機能については，同様に 2.2 節で示したように，AuthenticationEngine において認証成功の記録，SSOProfileHandler において利用した認証方式を含む応答が作成されることから，この間（2.2 節における，手順（7）と（8）の間）に実施する．これにより，IdP では，個別認証方式ごとに認証の成功を記録することで適切なシングルサインオンを実現する一方で，認証応答に対しては認証要求に整合させてグループ形式での認証ステートメントを発行できる．

3.3 実装

本研究では，前節までに述べた認証方式グループ化機能を実現するフィルタを開発した．なお，**表 2** に，本開発で用いた IdP と SP の仕様を示す．共に，Citrix XenServer 6.0.2 上の仮想サーバとして動作させた．

本機能は，大きく，既存認証機能実行前の変換処理と，既存認証機能実行後の逆変換処理に分けられるが，グループ情報の管理機能等が重複することから，本研究では，これらを単一のフィルタとして実装し，認証状態の有無によって実行する関数を分けることにした．

まず，共通処理として，XML で定義されたグループ情報を取り込む機能を実装した．ここでは，java.util.HashMap を用いて，変換処理，逆変換処理のそれぞれに対して，グループ形式と個別認証方式（複数可），個別認証方式とグループ形式（複数可）のマッピングを持つようにした．

変換処理では，変換処理用のマッピングを利用して LoginContext 内の requestAuthenticationMethods を変換すると共に，変換前の requestAuthenticationMethods を LoginContext 内に保存した．LoginContext には，propsMap という汎用的に利用できる java.util.Map が用意されていたため，変換前の情報はここに保存することにした．

また，逆変換処理では，逆変換処理用のマッピングを利用して，propsMap に保存していた変換前の requestAuthenticationMethods との整合性を取る処理を実装した．すなわち，実際に使用された認証方式が変換前の requestAuthenticationMethods に含まれない場合に，使用個別認証方式をグループ形式へと逆変換するようにした．逆変換処理を行った後も変換前の requestAuthenticationMethods と使用認証方式（グループ形式）が一致しない場合には，認証要求とは異なる方式で認証されたことになるため，認証エラーとするようにした．

これらの機能を有するフィルタを，テスト環境の IdP に組み込み，動作確認実験を行った．ここでは，3.2 節で述べたタイミングでフィルタが動作するよう，url-pattern を/AuthnEngine および/profile/SAML2/Redirect/SSO とした．動作確認実験の結果，本研究で開発したフィルタにより，グループ形式による認証方式の要求を IdP が的確に処理し，

表 2 IdP と SP の仕様.
Table 2 Specification of IdP and SP.

役割	OS とソフトウェアのバージョン
IdP	CentOS release 6.5
	Apache httpd 2.2.15
	Apache Tomcat 6.0.24
	OpenJDK 1.7.0_51
	Shibboleth IdP V2.4.0
SP	CentOS release 6.5
	Apache httpd 2.2.15
	Shibboleth SP V2.5.3

適切なシングルサインオンが実現できることを確認した．

4. おわりに

本研究では，Shibboleth IdP における LoA を考慮した認証方式のグループ化機能を開発した．

まず，既存の IdP において LoA を考慮した認証連携を実現することには，SP が連携先の IdP に関する詳細な情報を把握しておく必要がある等，運用上の問題があることを明らかにした．その問題を解決するため，本研究では，既存の認証機能による個別認証方式による認証を維持したまま，グループ形式による認証方式の要求に基づく認証連携を実現するためのフィルタを開発した．

今後は，本機能を拡張し，同レベルの LoA として複数の個別認証方式が定義されている場合等，認証方式が複数利用できる場合の選択方法について検討する予定である．

謝辞 本研究の一部は JSPS 科研費 26330158 の助成を受けたものである．

参考文献

[1] 秋山豊和，寺西裕一，岡村真吾，坂根栄作，長谷川剛，馬場健一，中野博隆，下條真司，長岡亨：大阪大学における全学 IT 認証基盤の構築，情報処理学会論文誌，Vol.49，No.3，pp.1249–1264 (2008).
[2] 内藤久資，梶田将司，小尻智子，平野靖，間瀬健二：大学における統一認証基盤としての CAS とその拡張，情報処理学会論文誌，Vol.47，No.4，pp.1127–1135 (2006).
[3] 国立情報学研究所：学術認証フェデレーション 学認 GakuNin (online), 入手先 ⟨http://www.gakunin.jp/⟩ (2014.05.07).
[4] OASIS: OASIS Security Services (SAML) TC — OASIS (online), available from ⟨https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security⟩ (2014.05.07).
[5] Shibboleth Consortium: Shibboleth (online), available from ⟨https://shibboleth.net/⟩ (2014.05.07).
[6] 只木進一，江藤博文，大谷誠，渡辺健次：認証基盤の効率化と「学認」への対応，情報処理学会研究報告，Vol.2012-IOT-17，No.10 (2012).
[7] 松平拓也，笠原禎也，高田良宏，東昭孝，二木恵，森祥寛：大学における Shibboleth を利用した統合認証基盤の構築，情報処理学会論文誌，Vol.52，No.2, pp.703–713 (2011).
[8] 足立紘亮，新村正明：複数の IdP へのシングルサインオンを可能にする認証システムの提案，情報処理学会研究報告，Vol.2011-IOT-13，No.17 (2011).
[9] 秋山豊和，西村健，山地一禎，中村素典：Web ブラウザ拡張機能を用いた認証連携基盤の機能拡張フレームワークの提案，電子情報通信学会技術研究報告，IA2012-3，pp.13–18 (2012).
[10] ITU-T: Entity authentication assurance framework, Recommendation ITU-T X.1254 (2012).
[11] Chehab M.I. and Abdallah A.E.: Assurance in Identity Management Systems, Proc. IAS 2010, pp.216–221 (2010).
[12] Cantor S., Kemp J., Philpott R. and Maler E.: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, saml-core-2.0-os (2005).
[13] Kemp J., Cantor S., Mishra P., Philpott R. and Maler E.: Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, saml-authn-context-2.0-os (2005).
[14] Shibboleth Consortium: Shibboleth Consortium - What's Shibboleth (online), available from ⟨https://shibboleth.net/about/⟩ (2014.05.07).
[15] Hughes J., Cantor S., Hodges J., Hirsch F., Mishra P., Philpott R. and Maler E.: Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, saml-profiles-2.0-os (2005).
[16] Shibboleth Consortium: Shibboleth Consortium - How Shibboleth Works (online), available from ⟨https://shibboleth.net/about/basic.html⟩ (2014.05.07).
[17] Shibboleth Consortium: IdPAuthnSession - Shibboleth 2.x - Confluence (online), available from ⟨https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthnSession⟩ (2014.05.07).
[18] 中村素典：LoA1 認定プログラム (online)，入手先 ⟨http://www.nii.ac.jp/service/openforum/setsumeikai2013/?action=common_download_main&upload_id=925⟩ (2014.05.07).
[19] Open Identity Exchange: What is a Trust Framework? — Open Identity Exchange (online), available from ⟨http://openidentityexchange.org/what-is-a-trust-framework⟩ (2014.05.07).
[20] Maler E., Nadalin A., Reed D., Rundle M. and Thibeau D: The Open Identity Trust Framework (OITF) Model (online), available from ⟨http://download.microsoft.com/download/D/8/E/D8E4FB66-563F-42C3-A08A-DAD15C50BD81/Open Identity Trust Framework Model_WP.pdf⟩ (2014.05.07).
[21] Shibboleth Consortium: SourceAccess - Shibboleth 2.x - Confluence (online), available from ⟨https://wiki.shibboleth.net/confluence/display/SHIB2/SourceAccess⟩ (2014.05.07).
[22] SWITCH: uApprove - /src/main/java/ch/SWITCH/aai/uApprove - リポジトリ - SWITCH Forge (online), available from ⟨https://forge.switch.ch/redmine/projects/uapprove/repository/show/src/main/java/ch/SWITCH/aai/uApprove⟩ (2014.05.07).
[23] 学認：FPSP (Filter Per SP, ユーザに対する特定 SP へのアクセス制限) プラグイン - GakuNinShibInstall - meatwiki (online), 入手先 ⟨https://meatwiki.nii.ac.jp/confluence/pages/viewpage.action?pageId=12158554⟩ (2014.05.07).