プライバシ情報逆流出に対する SAML/Shibboleth の仮名性強化手法
Toward Robust Pseudonymity in Shibboleth/SAML Federation against Backflow of Personal Information

大神　渉 †　　古村　隆明 ‡　　岡部　寿男 †
Wataru Oogami†　　Takaaki Komura‡　　Yasuo Okabe†

1　はじめに

Web サービスの利用が拡大し，多くの人がその恩恵を受けている．一人のユーザが複数のサービスを同時に使うことも多く，中でも SNS やネットショッピングなど個人情報をやり取りするサービスがその利便性を背景に急速に広まっている．しかし，現在のサービスモデルではサービスの提供者 (Service Provider,SP) に対し，ユーザが直接個人情報をやりとりせねばならず常にリスクを負わなくてはならない．また，異なるサービスを使おうとすれば，ユーザは異なる ID とパスワード (ID/PW) を用いる必要があり，不便であると同時に ID/PW そのものの安全性も考慮しなくてはならない．このように ID/PW を含めた個人情報の管理は難しく，情報漏えいなどにつながったケースも多い．しかし，酒類の購入など，ユーザに対してサービスの提供をする上で制限を加える必要もあり，サービスを提供する側がそのユーザの本人性またはその属性を把握することは必須である．

これらのユーザとサービス提供者の利便性を改善するために考えられたシステムとして，Single Sign On (SSO) がある．本稿では SSO 技術の中でも特に資源認可の柔軟性が高い SAML/Shibboleth を用いる．SSO とは Identity Provider (IdP) という認証専用のサービスを立ち上げ，各サービスにアクセスするための認証を一手に引き受けることで，ユーザが 1 度のログインで複数のサービスを利用できるシステムである．これによりユーザだけでなく SP にとっても個人情報の管理から解放され，情報漏えいに強くなる．IdP では従来ユーザが使っていた ID/PW に変えて，ユーザの識別子としての ID を発行する．この ID を使って SP はログを取る．従来の SAML/Shibboleth では SP がこのログを流出した場合にも Shibboleth の枠組みの外の攻撃者にそれを解析させない工夫がされていた．また，ある SP が他の SP と結託してログ上のユーザを特定する，いわゆる名寄せに対する対策として ID を使い分けている．そのため，この SP の利用ログは IdP を除く誰にも解析できない．しかし SP の利用ログはこれまでの SAML/Shibboleth において IdP に渡り解析することで IdP が本来知りえないユーザのプライバシ情報を手にすることについて考えられていない．本稿では SP から IdP に利用ログが流出することを ’’ 逆流出 ” として定義する．逆流出によって IdP が手に入れる事のできる情報はユーザの利用履歴であり，これは間接的にユーザの趣味・嗜好・病歴などのプライバシ情報である．これらの問題は IdP が発行した ID を SP がそのまま用いてログを取ることに起因する問題である．

本稿ではこの問題を SAML の標準エンティティである AP を利用することで ID を変換して解決する手法を提案する．既存の AP の実装には proxyIdP 方式及び SWITCH VO 方式が広く用いられていることから，それらを使った際の ID 変換方法に対する詳細設計を行い，両方式の比較及び考察を行った．

以降，２章で SSO と本研究で扱う Shibboleth の概要及び必要な要素を紹介する．続いて，３章で本研究が扱う問題を具体的に定義し，４章でこれらの問題を解決するための考察とそれを実現するために SAML の標準エンティティである AP(Attribute Provider) に注目し，ユーザのプライバシ情報における仮名性を高めることでその情報をより安全に扱う枠組みについて言及する．

2　SSO とそのプライバシ

SSO とはユーザが一度認証を受けるだけで許可されている複数のサービスを利用できるようになる (認可) システムである．SSO を実現することで，ユーザはサービスを受けるたびに認証を繰り返す必要がなくなる．そのため，ユーザはサービス毎に異なるパスワードをもつ必要がなくなり，サービスを提供する側は漏洩などのリスクを最小限に軽減することができる．

2.1　関連技術・研究

SSO の実現には主に，OpenID[1] や OAuth[2], SAML/Shibboleth [3] 等が用いられている．これらの技術では認証と認可を分離して管理することで SSO を

† 京都大学情報学研究科知能情報学専攻
Graduate School of Informatics, Kyoto University
‡ 京都大学情報環境機構 IT 企画室
Institute for Information Management and Communication

実現している．OpenID[1] は URL を用いて ID 認証を行っており，ID を使用するユーザの本人性を認証することで多くのサービスへのアクセスを可能にする技術である．しかし，この技術ではサービス側が持つユーザの個人情報に別のサービスがアクセスする際の認可について柔軟なアクセス制御ができないという短所がある．そこで，OAuth[2] では，ユーザ情報にアクセスするための権限をユーザが選択してサービス側に移譲することで対応している．この場合，細かなアクセス制御はできるが，サービス側に ID を知らせなくてはならず，利用者情報を少なからずサービス側に提供しなくてはならない．Shibboleth は Internet2/MACE[4] におけるプロジェクトの 1 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．そのためサービス側に ID を秘匿してサービスを受けることが可能であり，認可に関してユーザ固有の属性情報を用いることで細かな制御を可能にしている．本稿ではこうした点から Shibboleth を用いてプライバシ情報を保護する手法を検討する．

2.2 先行研究

これまで Shibboleth ではプライバシ上いくつか問題点も指摘されてきた．例えば，属性交換をする際，IdP が SP に対し必要以上に詳細な個人情報を属性として提示する問題である．これについては藤原ら [5] が Shibboleth に関する拡張を提案している．さらにその拡張の問題点として，IdP に対し必要以上に認可条件を知らせなければならないという点に対し高木ら [6] はマジックプロトコルを用いて不正者を明らかにし，問題を解決する拡張について提案している．Shibboleth では，直接個人情報をエンティティ間でやり取りすることはないため，前述のように属性交換を行う上でそれぞれのエンティティに対し，プライバシの問題が起こりやすい．例えば藤原ら [5] の研究は IdP を介して，ユーザ個人の情報が SP 側へ流出する問題を扱っており，高木ら [6] は，SP の秘匿情報が認可の過程で IdP 側に伝わる問題を扱っている．

2.3 ID とその仮名性

SAML/Shibboleth では IdP と SP 間で様々な属性を交換してサービスに対する認可条件として用いている．属性の中でも ID はそのフェデレーション内のユーザの識別子として用いられるものであり，どのユーザにも必ず ID が IdP から付与されている．ここでいう ID はユーザがログインするときに必要とされる ID とは別のものである．便宜上，ユーザがログインするときに用いる ID は “実 ID”とし，ここから後，“ID”は IdP と SP 間でやりとりされる属性情報中の ID を指し示すこととする．ID は IdP が発行し，SP に属性として渡される．SP では IdP から発行された ID を使い，サービスの利用履歴とそのユーザの ID を記録する．これは SP でユーザが不正を行った場合にフェデレーション内で IdP と SP が協力することで個人を特定するためである．従来の Shibboleth においては，SP のこのようなログに対し，複数の SP が結託してユーザを特定することでその機微情報に触れる不正，つまり名寄せに関する対策が取られている．SAML/Shibboleth 内で用いられる ID には主に次の 3 種類である．

Principal Name

どの SP に対しても同一の ID を発行するため，名寄せに対し仮名性が低い．ある個人だけが利用出来るサービスなどで用いられる．

TargetedID

eduPersonTargetedID[7] に代表される ID．SP 毎にハッシュされた ID であるため，名寄せに対する仮名性が高く，同じ SP を利用する毎に同じ ID が割り当てられる．図書の推薦機能など，匿名性が必要であるが同一ユーザに対して有効なサービスを提供する場合に特に用いられる．

TransientID

SP 毎，利用毎に異なる ID が発行される一時的な ID であるため，名寄せに対する仮名性が高い．電子書籍の貸し出しサービスなど，認可条件を満たせば，サービスに ID を特に必要としないサービスの提供に用いられる．

2.4 プライバシ情報

Shibboleth/SAML では SSO システムという性質上，様々なプライバシ情報を保護するよう設計がなされている．前述した名寄せに対応した ID もその 1 つである．本稿で扱うプライバシ情報は SP における利用履歴そのものである．本稿では仮名性の高い TargetedID と TransientID によって保存された利用履歴を「利用ログ」と呼ぶ．誰が（どの ID で）どのような物を購入したかという情報はプライバシ情報であり，誰であっても知るべきものではない．しかし，不正をしたユーザを特定する場合には利用ログを解析出来ることも求められており，保護のバランスは難しい．従来の SAML/Shibboleth ではこのプライバシ情報について名寄せ，それから Shibboleth フェデレーションの外部のユーザがログを解析できないよう設計されている．さらに，IdP と SP の協力により，不正ユーザを突き止めることが可能である．

3　プライバシ情報逆流出の問題

本研究では，SP の利用ログがフェデレーション内の IdP に渡ることでユーザのプライバシが侵害される問題を解決する．IdP は元来ユーザに関する属性情報を持っているが，そのサービスの利用履歴はプライバシ情報であり，保護されるべきものである．しかし従来の SAML/Shibboleth では考えられなかったケースの一つとして SP から IdP への情報の逆流出により，これまでは考えられてこなかった IdP へのプライバシ情報流出につながるのである．つまり，IdP がこの利用ログを手にした場合，IdP は前述のとおり認証を行い，ユーザに ID を付与している．そのため，ユーザと ID との対応付けが可能であり，どの種類の ID を使っても SP やユーザの意思に関係なく利用ログからユーザのサービス利用履歴という個人のプライバシに触れることが可能であり，重要なプライバシ問題である．(図 1)

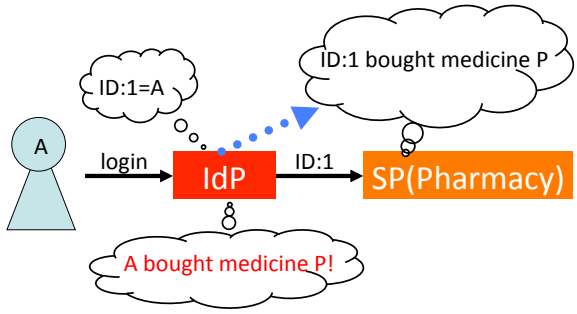

図 1: The problem concerned in this paper

逆流出の原因

利用ログは，個人情報保護法第一章第二条 1,2 に則ると「容易に個人と結びつけることができる」情報ではなく，法律上個人情報として扱われない．当然 SP のログ管理は甘くなり，流出する危険性が高いものである．また，SP 全てが利用ログの管理を厳重にすることは，現実的ではなく，漏洩という形以外にも極端な場合に盗まれると言ったケースも考えられないわけではない．また，Web サービスが拡充していく上で，SP ログを統計情報として SP が公開する事を考えた場合にも利用ログ上でのユーザの仮名性の確保は必要である．このように SP の利用ログはフェデレーション内であっても安全に扱われなくてはならない．

4　解決のための考察

4.1　アイデア

前述の問題を解決する方法として，ID を一度変換することで，IdP が割り振った ID と SP 内で用いられる ID の関係を IdP 側が把握しないようにする方法が考えられる．例えば図 2 のように IdP と SP の間に ID 変換機構を設けることで，IdP による SP の利用ログの参照を不可能にする．

これらの問題に加え，インシデント時の制約から，

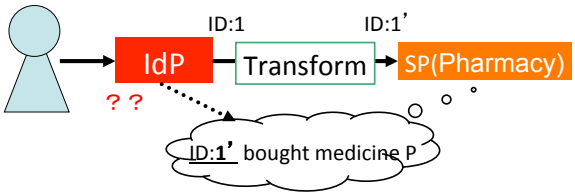

図 2: The idea for solving problems

変換後の ID は以下のようないくつかの制限が課せられる．

- ID は SP から見て直接特定のユーザと結びつけることができない
- 利用ログ上での ID は IdP から見て直接特定のユーザと結びつけることができない
- 利用者が不正した場合等ポリシーで予め決めておいた場面においてフェデレーション内で協力することで，ID から個人を特定することができる

このような制約を満たすよう変換した ID を **hashedTargetedID, hashedTransientID** と呼び，それぞれ従来の TargetedID と TransientID の代替として用いることでそれぞれの ID がもつ匿名性を強化する．またこれら 2 つを総称して単に **hashedID** と呼称する．また，従来手法で発行されたそれぞれの ID を変換する機構は AP(Attribute Provider) という SAML 標準のエンティティを用いることであくまで SAML/Shibboleth の枠組み上でこのアイデアを実現できるよう考える．

4.2　AP の実現

AP(Attribute Provider) は SAML 標準のエンティティである．IdP とは別にユーザに付加的な属性を管理するためのエンティティであり，IdP と SP の中間に位置する．AP には主に 2 つの実装が考えられている．

ProxyIdP 方式

この方式は図 3 に示す方式で，IdP と SP 間の通信

は必ず AP を通して行われるため直感的に理解しやすい実装である．その反面，ユーザは従来の DS で IdP を選ぶのに加え，その前に自分が経由する AP を選ばなくてはならず，利便性が損なわれる可能性がある．

SWITCH VO 方式

この方式は図 4 に示す方式である．IdP と SP 間の通信は従来方式と変わらないが，SP と AP の間でバックチャンネルによる通信を行うことで IdP を経由せずに付加的な属性を得ることができる．この場合，ブラウザを使うユーザは DS で 1 度だけ IdP を選ぶという従来と変わらない操作で特に意識することなく AP による恩恵を受けることができる．

図 3: ProxyIdP method

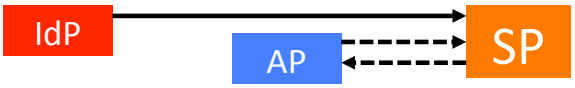

図 4: SWITCH VO method

２つの方式にはすこしずつ差異があるため，その長所短所を見極めて実装する必要がある．

4.3 hashedID の構築

AP を使って実際に hashedID を実装するときの詳細設計について述べる．hashedID を ProxyIdP 方式で実現しようとすれば，AP はただ IdP から発行された ID を変換するだけでよい (図 5). また，ProxyIdP 方式でインシデント対応するときも同様に AP を通して IdP に問い合わせることでユーザを特定することが可能である (図 6). このようにして生成された hashedID は IdP に対しプライバシ情報を秘匿しつつ，SP に対しては特に挙動を変化させることなく使うことが出来る．しかし，ユーザ負担の少ない SWITCH VO 方式の AP を用いる場合には以下の制約を考慮した設計を行わなくてはならない．

- SP には変換前の ID を受け取る必要があるがその ID でログを取ってはならない

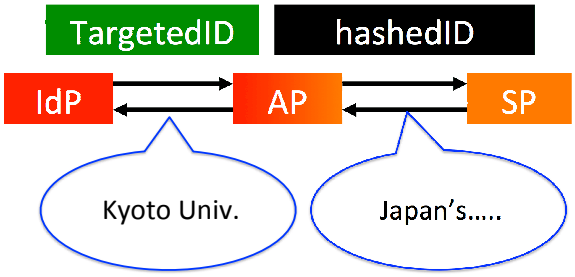

図 5: Example using proxyIdP

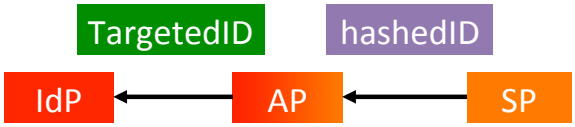

図 6: ProxyIdP method when an incident occurs

- SP が変換前に IdP から受け取った ID は保持してはいけない

そこで，IdP は発行する ID を予め SP にはわからない方法で暗号化して送信することを考える．この場合，属性情報中に ID がなくなってしまうので TransientID をつけて暗号化した ID を SP に送ることにする．このように SWITCH VO 方式においては ProxyIdP 方式とは異なり，まったく別の実装を考える必要がある．以下では，SWITCH VO 方式における hashedID を生成する手順とインシデント時の各エンティティの動きを簡単に述べる．

4.3.1 ユーザが資源にアクセスするとき

具体的には以下の手順で SP は hashedID を次のようにして得る (図 7).

1. IdP と AP は事前に鍵 (I) を交換しておく
2. IdP は SP に transientID(Ⅱ) と I によって暗号化した TargetedID(もしくは TransientID)(Ⅲ) を SP に送信する
3. SP は Ⅱ を使い Ⅲ を AP に送信する
4. AP は Ⅲ を I で復号し，それを元に hashedID(Ⅳ) を作成し SP に送信する
5. SP は Ⅳ を使って利用ログを取る

また，この時重要なのが **Ⅱ によるログを残さないこと** である．つまり SP は受け取った Ⅱ を即座に破棄する設定かもしくは受け取った ID をログに保存しないよう

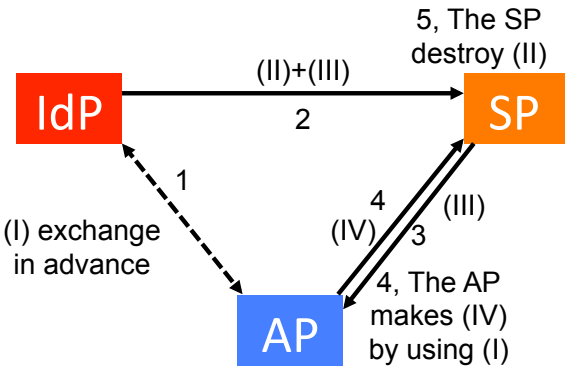

図 7: SWITCH VO when user access resources

明示的に設定しなくてはならない．この設定は SP のログに II が残ることで IdP が問い合わせ時間などの外部的要因から ID を推定することを防ぐ効果を想定している．

4.3.2 SP がユーザを特定するとき

ユーザが不正をするなどのインシデントにより，SP がユーザを特定したいときも存在する．予めこの場合はインシデント時のポリシーをユーザ間で構築しておき，ポリシーに則り，フェデレーション内での協力により不正を行ったユーザを特定出来るよう取り決めておく必要がある．この場合次のようにして hashedID から個人を特定できる (図 8).

1. SP は特定したいユーザの hashedID を IdP に対して問い合わせる
2. IdP は受け取った hashedID を AP に渡す
3. AP は受け取った hashedID から IdP が発行した TargetedID もしくは TransientID に逆変換を行い IdP に渡す
4. IdP はポリシーに則ったペナルティをユーザに課す

4.4 hashedID の逆変換

AP を使って ID を変換する時，AP は同時に変換後の hashedID に対し元々の ID を求める方法を持たなくてはならない．方法としては，

1. hashedID と ID に関する変換テーブルを持つ
2. 変換に可逆変換可能な関数を使う

1 の場合, 変換は AP が任意の関数で行うことができるため，実装が簡易になることが予想される．しかし，変換テーブルが流出した場合に逆流の問題が起こりうる．

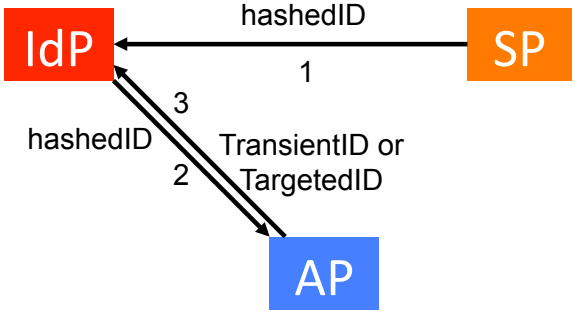

図 8: SWITCH VO when an incident occurs

また変換テーブルを破棄した後に不正が発覚した場合には IdP から全ての ID を提示してもらわねばならない．一方，2 の場合，変換に関して可逆変換ができるために個別にログを管理する必要がない．しかし，この場合もこの関数を秘密にせねばならず，関数の管理が難しい．本稿では実装の簡易性から 1 の手法をとるが，2 の方法についても十分に検討したい．

5 まとめ

フェデレーション内で情報の逆流が起こる際の危険性について指摘し，その問題を新たに AP と hashedID を用いることで解決する手法を提案した．今後は実際に AP と hashedID を実装し，条件や AP に対する信用性など評価の柱を用意して考察を進めていく．

参考文献

[1] OpenID Foundation. OpenID. `http://openid.net/`, Jun,2007.

[2] OAuth Core Workgroup. OAuth 1.0a. *OAuth Core 1.0 Revision A*, 24 June,2009.

[3] Shibboleth-A project of the Internet2 Middleware Initiative. `http://shibboleth.internet2.edu/`.

[4] MACE-Middleware Architecture committee for Education. `http://middleware.internet2.edu/MACE/`, 4 2011.

[5] Shoichiro Fujiwara, Takaaki Komura, and Yasuo Okabe. A Privacy Oriented Extension of Attribute Exchange in Shibboleth. *SAINT2007 Workshop on Middleware Architecture in the Internet*, Jan. 2007.

[6] Toshihiro Takagi, Takaaki Komura, Shuichi Miyazaki, and Yasuo Okabe. Privacy oriented attribute exchange

in shibboleth using magic protocols. *SAINT2008 Workshop on Middleware Architecture in the Internet*, pp. 293–296,Turku, FINLAND, 28 July - 1 Aug. 2008.

[7] Keith Hazelton(The editor of the MACE-Dir working group). eduPerson Object Class Specification (200806). *Internet2 Middleware Architecture Committee for Education, Directory Working Group (MACE-Dir)*, 30 June 2008.