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

2008－DSM－48 (7)
2008－QAI－26 (7)
2008／3／6

マジックプロトコル利用によるプライバシーに配慮したShibboleth属性交換の拡張

高木 俊宏[†] 古村 隆明[††]
宮崎 修一[††] 岡部 寿男[††]

本研究では、Webサービスにおいて連携アイデンティティを実現する枠組みであるShibbolethにおける属性交換について、従来の手法では属性あるいは認可のための条件を直接やりとりするために、プライバシーが必要以上に漏れる危険があることを指摘した．その上で、これを解決するためにマジックプロトコルを利用した属性交換手法を設計した．また本手法は通信によって交換される情報が暗号化されているにも関わらず、不正が行われた場合に、のちの検証で不正者が特定できるように設計した．

Privacy Oriented Attribute Exchange in Shibboleth Using Magic Protocols

Toshihiro Takagi,[†] Takaagi Komura,[††] Shuichi Miyazaki[††]
and Yasuo Okabe[††]

In frameworks for Shibboleth, there are cases where users must present detailed privacy information which Service Provider (SP) does not actually require to authorize them. We propose an extension of the attribute exchange protocol between an Identity Provider (IdP) and an SP in Shibboleth. While in the conventional framework of Shibboleth attributes are exchanged in immediate values, in our new extension an SP and an IdP exchange attributes according to the protocol for Millionaire's Problem and the protocols for Oblivious Transfer (these protocols are known as "Magic Protocols"). This extension enables the SP to know whether user's attributes meet the requirement for authorization, without the SP and the IdP revealing their confidential information. We also show how we can detect cheating in execution of this protocol, e.g. the IdP tells an another value instead of a true value to the SP in malice.

1. はじめに

連携アイデンティティ(Federated Identity)は，多様化するウェブアプリケーションサービスにおいて，異なるドメインに属するユーザのアイデンティティを連携させることによって，あるサービスがユーザの認証・認可のために要求する個人情報を，適切に，安全に，単純に提供することを目的とする．ウェブサービスでは，連携アイデンティティにおける属性情報やアクセス制御情報を伝達するプロトコルとしてSAML (Security Assertion Markup Language) * が規定されている．連携アイデンティティやシングルサインオン (Single Sign On, SSO) を実現するために，SAMLを拡張実装，あるいは反映したものとして，Shibboleth** やLiberty*** などがある．

個人情報保護やセキュリティの観点から，サービスプロバイダ (Service Provider, SP) とアイデンティティプロバイダ (Identity Provider, IdP) の間で交換される情報は，必要最低限であることが望ましい．従来のShibbolethの下ではユーザの属性とその値を直接IdPからSPへ伝え，SPがそれを元にユーザを認可する．この属性交換方式には，必要以上に詳細な属性情報がSPへ提供される危険性がある．これに対し藤原ら[1]は，ユーザの属性が満足すべき条件（以下，認可条件と呼ぶ）をSPからIdPへ提示し，IdPからSPへ認可条件の判定結果を"true/false/unanswerable"として伝えることにより，必要以上に詳細な属性情報がSPへ提供されるのを防ぐ拡張を提案している．しかしこの拡張では，SPの認可の判断基準がIdPに提供されてしまうという課題が残る．

本稿では，藤原らの拡張における認可条件の判定において「金持ちの財産比べプロトコル (Millionaire Problem)」[2]および「紛失通信 (Oblibious Transfer)」[3]と呼

† 京都大学情報学研究科
Graduate School of Informatics, Kyoto University
†† 京都大学学術情報メディアセンター
Academic Center for Computing and Media Studies, Kyoto University

* http://www.oasis-open.org/committees/te_home.php?wg_abbrev=security

** http://shibboleth.internet2.edu/

*** Liberty Alliance Project: http://www.projectliberty.org/

ばれるマジックプロトコルを用いることで，SP の認可の判断基準を秘密にしたままで認可条件を判定する手法を提案する．

以下では，まず 2 節で，金持ちの財産比べプロトコルおよび紛失通信を紹介する．3 節で，Shibboleth において認証・認可がどのように行われるかについて説明し，Shibboleth の属性交換が抱えるプライバシーに関する問題を説明する．4 節で，本研究が提案するマジックプロトコルを用いた属性交換の手法を述べ，最後に，5 節で，結論と今後の課題について述べる．

2. マジックプロトコル

2.1 暗号とマジックプロトコル

ネットワークを介して，暗号を利用してある特定の条件を満たすような情報交換を実現する方法を，マジックプロトコルと呼ぶ[4]．マジックプロトコルは，公開鍵暗号による認証や署名といった理論を基礎としている．

2.2 金持ちの財産比べプロトコル

金持ちの財産比べプロトコルは，Alice と Bob の 2 人がお互いの資産額を秘密にしたままで，どちらが金持ちかを判定する通信プロトコルである．

2.2.1 プロトコル

今，Alice が i 円，Bob が j 円それぞれ所持しているとする．ただし，$1 \leq i, j \leq N$ であり，i, j, N はそれぞれ整数とする．

(1) Bob は，公開鍵-秘密鍵のペア (E_B, D_B) を生成し，公開鍵 E_B を Alice に送る．

(2) Alice はランダムに整数 x を選ぶ．

(3) Alice は，$y = E_B(x) + i$ を計算し，y を Bob に送る．ただし $E_B(x)$ は x を公開鍵 E_B で暗号化したものである．

(4) Bob は以下を計算する．

$$u_k = D_B(y-k) \qquad (k = 1, \ldots, N)$$

ここで，$D_B(x)$ は x を秘密鍵 D_B で復号したものを表す．

(5) Bob は，$u_1, \ldots u_N$ に対し，

$$v_k = \begin{cases} E_B(u_k + 1) & (k = 1, \ldots, j) \\ E_B(u_k + 1) + 1 & (k = j+1, \ldots, N) \end{cases}$$

を計算し，$v_1, \ldots v_N$ を Alice に送る．

(6) Alice は，$v_1, \ldots v_N$ の中に，$E_B(x+1)$, $E_B(x+1)+1$ のうちどちらが含まれているかを調べる．

$$E_B(x+1) \text{ が含まれている} \Rightarrow i \leq j$$
$$E_B(x+1)+1 \text{ が含まれている} \Rightarrow i > j$$

2.2.2 計算量と通信量

この財産比べプロトコルは，4. および 5. でそれぞれ N 回の計算が実行されるので，計算量は $O(N)$ である．一方，5. において Bob は N 個の数 $v_1 \cdots v_N$ を Alice に送る必要がある．1 つのデータのビット長が M ビットであるとき，このプロトコルに必要とされる通信量は，$O(MN)$ である．例えば，$N = 100,000,000$, $M = 64$ のとき，4. において Bob から Alice に送信されるデータ容量は 800Mbyte となる．すなわち，この財産比べプロトコルを実装するにあたっては，N の値が現実的に実行可能な範囲であるかを十分に考慮しなければならない．

2.3 紛失通信

紛失通信は，送信者である Bob が N 個の情報 $m_1, \ldots, m_N$ を保持しており，そのうちいずれか 1 つのみが受信者の Alice に伝わるが，Alice がどれを受け取ったかを送信者の Bob は知ることができないというプロトコルである．

2.3.1 プロトコル

Bob は N 個のメッセージ $m_1, \ldots, m_N$ を保持しており，このうち Alice はメッセージ m_i を入手したいとする．

(1) Bob は，公開鍵-秘密鍵のペア (E_B, D_B) を生成し，乱数 $r_1, \ldots, r_N$ を選ぶ．公開鍵 E_B と乱数 $r_1, \ldots, r_N$ を Alice に送る．

(2) Alice は，乱数 x を選び，$y = E_B(x) + r_i$ を計算する．ただし $E_B(x)$ は，x を公開鍵 E_B で暗号化したものである．y を Bob に送る．

(3) Bob は受け取った y に対して，

$$u_k = D_B(y - r_k) \quad (k = 1, \ldots, N)$$

を計算する．ただし $D_B(x)$ は，x を秘密鍵 D_B で復号したものを表す．Bob は，$m_1 + u_1, \ldots, m_N + u_N$ を Alice に送る．

(4) Alice は，$u_i = D_B(y - r_i) = D_B(E_B(x) + r_i - r_i) = x$ であることを知っているので，$m_i + u_i - x$ により，メッセージ m_i を入手する．

2.3.2 計算量と通信量

このプロトコルは 1. および 3. より $O(N)$ の計算量がかかる．一方，1. および 3. で Bob は Alice に N 個のデータを送るため，1 つのデータのビット長が M ビットであるとき，このプロトコルに必要とされる通信量は，$O(MN)$ である．財産比べプロトコルと同様に，このプロトコルを実装するにあたっては，N の値が現実的に実行可能な範囲であるかを十分に考慮しなければならない．

3. Shibboleth における属性交換とその問題

3.1 Shibboleth による属性交換

Shibboleth は，Internet2/MACE*におけるプロジェクトのひとつであり，アクセス制御下のウェブ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。実装の最新版は 1.3 で，SAML1.1 がベースである．そして現在 SAML2.0 をベースとした Shibboleth2.0 が開発中である．以下，特に断らない限り，Shibboleth とは Shibboleth1.3 に関連する枠組み及び仕様のことを指すものとする

Shibboleth で交換されるメッセージは SAML で規定されているものと同じである．Shibboleth のアーキテクチャ

* MACE: http://middleware.internet2.edu/MACE/

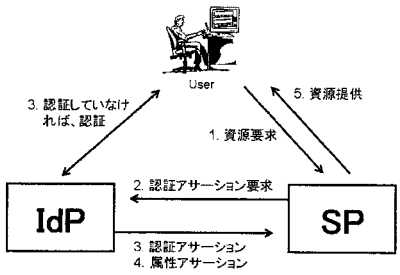


図 1 Shibboleth のアーキテクチャ

は，ユーザ，IdP，SP，および WAYF サービス (Where Are YouFrom Service) という 4 種類のエンティティから構成される．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．図 1 に，Shibboleth における認証と認可の流れを説明する．

(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 はユーザが誰であるか，また，資源へアクセスする権限を持っているかを知らないので，IdP へユーザの認証及び属性を要求する．

(3) ユーザは IdP に対して認証を行う．

(4) IdP は SP から要求された属性を提供しても良いかどうかを属性開放ポリシー (Attribute Release Policy, ARP) 等を利用して確かめ，提供しても良いならば属性とその属性値を SP に渡し，提供できないならばエラーメッセージを SP に渡す．また，同時に認証情報も SP へ渡す．

(5) SP は IdP から受け取ったメッセージから認証情報，属性を得て，属性受理ポリシー (Attribute Acceptance Policy, AAP) に従い認可決定を行う．

ユーザの匿名性を保つために，IdP は SP に対して仮名 ID をユーザの識別子として提供する．この仮名 ID は，ユーザの実際のアイデンティティに関する情報を含まない．

Shibboleth のモデルにおいて，IdP と SP の間で交換される属性とは次のように定義される．

- 属性とは，属性項目と属性値の組である．
- 属性値は有限長のビット列で，その長さは固定長または予め与えられた上限値以下である．

ユーザは IdP に対して，予め指定されている有限個の属性項目について，自身の属性値を登録する．SP は，これらの属性のうち認可に必要となる属性項目を指定し，対応する属性値を IdP から受け取って，ユーザの認可決定を行う．受け取った属性情報を利用して SP がどのように認可決定を行うかについては，制限はない．

以下，用語の定義として，SP が認可のプロセスに基づいてユーザに資源へのアクセス権限を与える決定を下すことを承認とよび，反対にアクセス権限を与えない決定を下すことを否認とよぶことにする．

3.2 属性交換における問題

3.2.1 従来の属性交換

Shibboleth の枠組みの下 IdP と SP 間で交換される属性情報はユーザを特定するための手掛かりを含む．そのためにこの属性情報が，利用後は破棄されるという取り決めがある場合にもかかわらず，蓄積されると属性の保持者を特定される可能性がある．ユーザの匿名性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，“ユーザ U 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 “U は 25 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当たる．

しかし，現状の Shibboleth(SAML) における属性交換プロトコルの下では属性項目とその属性値を直接交換する以外の方法では属性を交換できない．

3.2.2 藤原らによる属性交換の拡張プロトコル

上述の問題に対して藤原らは，従来の属性交換の枠組みにおいて，以下のような属性情報交換を可能にする拡張を提案した（以下，これを藤原らの拡張プロトコルと呼ぶことにする）．

> 認可を要するウェブリソースに対してある 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（以下，これを認可条件と呼ぶ）を IdP に渡す．条件の記述言語は XACML をベースに定義する．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．

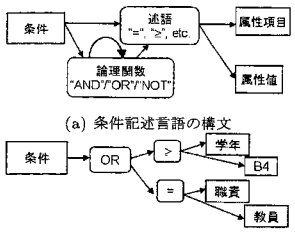
この拡張プロトコルが適用できるのは，その認可条件が，新たに定義した条件記述言語によって記述可能である場合に限られる．

藤原らが定義した条件記述言語は，図 2-(a) のような形式で説明される．この条件記述言語によると，藤原らの拡張プロトコルが扱う認可条件は，ある属性項目と属性値が満たすべき関係を「述語」で繋ぎ（述部），それに 0 個以上の論理関数を適用させたものとなる．述語には，等不等号が用いられる．例えば “「学年」$\geq$「B4」” または「職責」=「教員」” で表される認可条件は藤原らの拡張プロトコルを適用することができる (図 2-(b))．

このような必要以上の属性情報の交換は，Liberty においても問題視されており，藤原らのプロトコルに似た拡張が検討されている．

3.2.3 藤原らの拡張プロトコルにおける問題

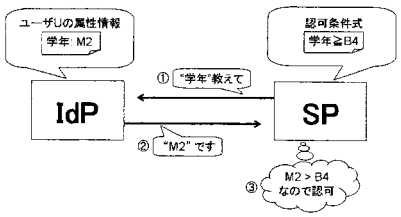
ユーザ U が持つ属性 a を a_U，その値を $V(a_U)$ とする．属性 a_U の属性値が v_a に等しいときを $V(a_U) = v_a$，等しくないときを $V(a_U) \neq v_a$ とする．また，属性 a_U



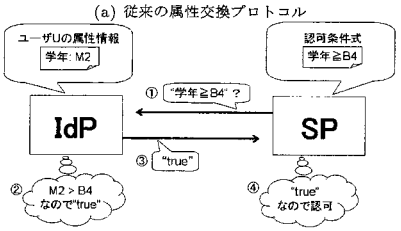
(a) 条件記述言語の構文

(b) 条件例：「学年」≥「B4」または「職責」＝「教員」の場合

図 2 藤原らの拡張プロトコルにおける条件記述言語



(a) 従来の属性交換プロトコル



(b) 藤原らによる拡張プロトコル

図 3 属性交換方法の違い

の属性値と v_a との大小関係を不等式によって表すことができるものとする．また，$\wedge$，$\vee$ をそれぞれ “and”，“or” の意味で用いる．

藤原らの拡張プロトコルにおいて扱われる SP の認可条件は，例えばユーザ U のある属性 a_U，b_U に対して，$\{V(a_U) = v_a\} \wedge \{V(b_U) \geq v_b\}$ といった形で定式化できる．このように認可条件を式で表したものを，認可条件式と呼ぶことにする．前述の認可条件式において v_a，v_b は，ユーザが承認を得るために各属性が満足すべき値と解釈できる．以下，この値を認可条件のしきい値 (あるいは単にしきい値) と呼ぶ．例えば，認可条件が図 2-(b) の例で表されるとき，認可条件式は { 学年 $\geq$B4} $\wedge$ { 職責 = 教員 } と表すことができ，このとき認可条件のしきい値は “B4” および “教員” である．

従来の Shibboleth における属性交換プロトコルでは，SP がこの認可条件式の評価を行う仕様であるため，IdP は $V(a_U)$，$V(b_U)$ を SP へ渡さなければならなかった (図 3-(a))．一方，藤原らの拡張プロトコルでは，SP は認可条件式を IdP へ渡し，IdP がその評価を行う (図 3-(b))．すなわち，2 つの方式の最大の違いは，認可条件式をどちらのエンティティが評価するかというところにある．

さて，改めて藤原らの拡張プロトコルを振り返ると，認可条件のしきい値が SP から IdP へ提供されているのが分かる．しきい値は，SP が自らの管理する資源に対して設定するものであり，SP の秘密情報であるといえる．例えば，ある企業の就職採用課が提供するウェブページに対するアクセス認可条件が，「次年度卒業予定者でかつ “論理回路” の成績が “優”」であったとする．そして，アクセスしてきた学生の所属する各大学 (IdP) に対して認可条件式を渡す．企業が就職採用において何を重視しているかという情報は企業にとっては秘密情報であり，この情報が IdP を通じて拡散することは望ましくない．

このように，認可にまつわる属性交換においては，IdP から SP へ渡される属性情報は必要最小限に抑え，また SP の持つ認可条件のしきい値も必要以上に IdP へ渡さないようにすることが望ましい．そのためには，認可条件式の評価をエンティティに委ねず，かつ評価の結果だけを SP が受け取るように Shibboleth を拡張しなければならない．

4. マジックプロトコルを用いた属性交換の拡張

3 節での議論を踏まえ，本節では，藤原らの拡張プロトコルによる属性交換の枠組みにおいて，マジックプロトコルを用いた通信を行うことによって，属性情報や認可条件のしきい値を直接交換することなく認可条件を評価することが可能であることを示す．

4.1 想定するモデル

3.2.2 節で説明したように，藤原らの拡張プロトコルで扱われる認可条件は，1 個以上の述語（述部）に AND/OR/NOT を作用させた形で表される．本稿では，この述語を次の 3 つのケースに細分化する．

同値判定型 属性値がしきい値に一致するかどうかで表されるもの

大小判定型 属性値がしきい値より大きいか（小さいか）で表されるもの

所属判定型 扱われる属性が，複数の属性値から成る有限集合で表され，その属性値集合の中に，しきい値に一致する値が含まれるかどうかで表されるもの．

同値判定型は，例えば「会員番号は 0123 であるか」のように，ユーザの属性値がしきい値に一致（不一致）するかを認可条件とするものである．これは，認可条件の最も基本的な型である．大小判定型は，「ユーザは 20 歳以上か？」のように，ユーザの属性の値がしきい値より大きいか小さいかを認可の条件とするものである．そのため，大小比較型で扱う属性は数値のように大小の概念が定義されるものである．簡単のため，特にここでは属性が整数で表されるものに限定する．認可に必要となる属性は，普通，年齢や年収，何らかのスコアなど我々が社会的に保有する情報であるため，整数に限定しても大きな問題はないと考えられる．所属判定型は，値を複数もつような属性があるとき，この中にしきい値と等しい

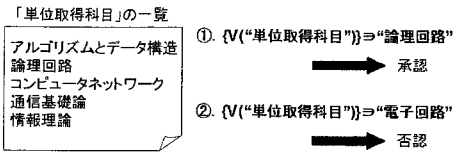


図 4　所属判定型の例

i	科目名	m_i
0	アルゴリズムとデータ構造	○
1	論理回路	○
2	コンピュータネットワーク	○
3	通信基礎論	○
4	情報理論	○
5	計算機科学概論	×
6	数理工学概論	×
7	グラフ理論	×
8	計算機アーキテクチャ	×
9	電子回路	×

図 5　「単位取得科目」のデータ構造
i はインデックス番号，m_i はメッセージを表す．
○は単位を取得していること，×は取得していないことを表す

値が含まれるか含まれないかで認可を判定するケースをいう．例えば，「単位取得科目のうち，"論理回路" が単位取得されているか」といった認可条件が該当する（図4）．図 4 の例の場合，認可条件式が図の①で表されるときはユーザは承認されるが，図の②で表されるときは否認される．以下では，値を複数もつような属性をリスト型属性とよぶことにする．以上 3 つのケースのうち同値判定型の認可条件式は，認可条件のしきい値を IdP に伝えなければ認可判定を行うことは不可能である．本稿では，大小判定型と所属判定型について，3.2 節で説明した属性交換における問題を解決する属性交換プロトコルを考察する．

4.2　提案プロトコルの設計

まず，大小判定型について考える．大小判定型では，IdP のもつ属性と SP のもつ認可条件のしきい値をお互いに秘密にしたまま，それらの大小関係を SP が知ることができればよい．この要求は，金持ちの財産比べプロトコルの機能によって実現できる．

次に，所属判定型について考える．所属判定型では，リスト型属性の値の中に認可条件のしきい値が含まれるかどうかを SP が知ることができればよい．本稿では紛失通信を適用することでこれを実現する．紛失通信を用いた認可判定の方法を，図 4 の例をもとに 2.3.1 節に沿って説明する．

まず，リスト型属性のデータ構造を定義する．「単位取得科目」がとりうる全要素（学生が取得可能な科目）は，図 5 の 10 科目であったとする (すなわち $N = 10$)．これらの要素それぞれにインデックスを与える．IdP には，図 5 のように各要素に対してユーザがその要素（科目）の単位を取得しているかどうかという情報をメッセージとしてもたせる．ここでは，単位を取得している場合には "○"，取得していない場合には "×" とする．この属

表 1　認可条件 $A \wedge B$，$A \vee B$ のとき SP が得る情報

	A の真偽	B の真偽	認可決定に必要な情報	A,B を独立に評価した場合に SP が知る情報
$A \wedge B$	真	真	A，B ともに真	A，B ともに真
	真	偽	A，B のどちらかが偽	A は真，B は偽
	偽	真	A，B のどちらかが偽	A は偽，B は真
	偽	偽	A，B のどちらかが偽	A，B ともに偽
$A \vee B$	真	真	A，B のどちらかが真	A，B ともに真
	真	偽	A，B のどちらかが真	A は真，B は偽
	偽	真	A，B のどちらかが真	A は偽，B は真
	偽	偽	A，B ともに偽	A，B ともに偽

性をもとに，2.3.1 節のプロトコルに従って，SP と IdP は紛失通信を行う．紛失通信の結果 SP は，図 4-①の認可条件の場合は m_1 ="○" を，図 4-②の場合は m_9 ="×" をそれぞれ得る．これらの情報をもとに，SP は認可決定を行う．

4.3　認可条件式を AND/OR で接続したプロトコルの設計

4.2 節では，藤原らの条件記述言語における「述部」に対してマジックプロトコルを利用する拡張を説明した．本節では，それら述部を AND や OR で接続した認可条件（$A \wedge B$ や $A \vee B$ など）におけるプロトコル設計について考える．

最も単純な方法は，A と B をそれぞれ独立に評価し，最後に SP がそれらの結果から $A \wedge B$ や $A \vee B$ を評価するという方法である．ところが，この方法では認可に必要以上の情報を SP に提供してしまう可能性がある．例えば，藤原らの条件記述言語で記述されるある認可条件式 $A \vee B$ に対して，SP が認可決定に必要な情報は，A と B のうちどちらかが成立している，ということのみであり，実際にどちらが成立しているか（あるいは両方が成立しているか）については知る必要がない．こういった情報は，認可に必要以上の情報であり，A と B を独立に評価する方法では，この情報提供を防ぐことはできない（表 1）．

そこで，同値判定，大小判定，所属判定の結果を IdP が知るようにそれぞれのプロトコルを変更し，それらの結果を用いて IdP が AND/OR/NOT の計算を行うようにプロトコルを設計する．

大小判定すなわち財産比べプロトコルについては，IdP と SP の役割を入れ替えるだけでよい．所属判定すなわち紛失通信については，次のようにプロトコルを改める．

(1)　通信を行う前に，IdP は，当該リスト型属性のメッセージである 2 値の記号，"○" と "×" を，確率 1/2 で "0" と "1" にマッピングする（SP には，真を意味する記号が "○" であるか "×" であるかは分からない）．

(2)　IdP と SP は，紛失通信に基づき通信を開始し，SP は所望の要素に対するメッセージ m_i を受け取る．

(3)　SP は，受け取ったメッセージ m_i と，m_i が満たすべき値 v（"0" か "1" か）を IdP へ送る．

(4)　IdP は，SP から受け取った 2 つの情報から，$m_i = v$ であるかどうかを評価する．

この操作によって，SPは，当該リスト型属性の中にしきい値に一致する値が含まれるかどうかをIdPから伝えられることになる．

4.4 不正の種類と検証

属性交換が正しく行われるためには，IdPあるいはSPが何らかの不正を行った場合にその不正を正しく検証する仕組みがなくてはならない．とりわけ，本稿が提案する属性交換プロトコルにおいては，IdPのもつユーザの属性とSPのもつしきい値が互いに秘密に保たれたまま(暗号化されて)認可条件の評価を行うため，不正の責任の所在が自明ではない．この節では，提案プロトコルにおいて考えられる不正を整理し，それらの不正をどのような仕組みで検証するかについて説明する．

考えられる不正は，次のものが挙げられる．

1. **SPによるしきい値の改ざん** SPが間違ったしきい値を用いる．
2. **IdPによる属性値の改ざん** IdPが間違った属性値を用いる．

各エンティティの否認（自分の通信を否認する），なりすまし（別のエンティティの名を名乗る），通信の盗聴といった不正については，Shibbolethによるメッセージ交換はSSL/TLS☆やXML署名☆☆・暗号☆☆☆によって保護され，交換されたメッセージはログとして保持する決まりであるため，防止・検証することができる．

不正検証の仕組みを作る上で，我々はまず以下の取り決めを定めた．

> 不正を検証する場合は，SPが証明者となって，自らの認可の手続きの正当性を証明する．SPが手続きの正当性を証明できなかった場合は，SPに責任があったとみなす．

この取り決めは，SPにすべての通信記録の責任を持たせることで，自身の正当性を主張する権利を与えている．これは，次のような理由からである．例えば，自分は確実に承認されると思っていたユーザがもし否認されてしまったとき，そのユーザはまず第一に当該SPに対して抗議するはずである．そのときSPはユーザに対して，自分は正しいデータと正しい手順に従って認可決定を行ったと証明しなければならない．このように，一般に，認証・認可で何らかの不正があったと疑われるとき，まず最初に検証の対象となるのはサービス提供者であるSPだからである．言い換えれば，通信記録を開示してもなおSPが自身の正当性を証明できなければ，SPは強制的に不正者とみなされる．

この取り決めの下，各マジックプロトコルにおける不正を検証するためのスキーマを，財産比べプロトコルの場合を例に説明する（紛失通信も同様の手順）．SPおよびIdPは，2.2.1節のプロトコルを開始するにあたって次の操作を行う．

- SPは認可のしきい値i，2.2.1節の手順2.のランダムな整数xおよび手順3.のyをそれぞれハッシュ化したもの$h(i)$，$h(x)$，$h(y)$をIdPへ渡す．IdPは送られてきたデータに対して署名を施し，SPへ返す．

この操作は，認可決定において財産比べプロトコルが行われたときに，確かにi，x，yが用いられたことを証明するためのものである．IdPの署名を施すことによって，検証のときSPが改ざんした証拠を提出することを防いでいる．一方，IdPはハッシュ値$h(i)$，$h(x)$，$h(y)$からは元のデータi，x，yを知ることはできない．不正検証は，次の流れに従って行う．

(1) SPは検証者（第3機関）に対してi，x，yと，IdPの署名が施されたそれらのハッシュ値$h(i)$，$h(x)$，$h(y)$を渡す．検証者はIdPの署名が有効なのを確認し，また，元データのハッシュ値が，受け取ったハッシュ値と一致することを確認する．

(2) SPは保持していた通信で交換された情報（これらは全てにSPおよびIdPの署名が施されている）を検証者に渡す．検証者は，SPから受け取った情報に基づき財産比べプロトコルを再現する．

(3) 検証者は，財産比べプロトコルの再現結果が，当初SPが主張した結果（承認/否認）と一致すれば，IdP側が何らかの不正を行ったとみなす．一致しなかった場合は，SPが不正を行ったとみなす．

5. おわりに

本稿では，Shibbolethにおける属性交換において，これまで見過ごされてきたプライバシー漏えいの問題を明確にした．その上で，マジックプロトコルを用いた新しい認可判定のプロトコルを提案した．このプロトコルによって，ユーザの属性情報および認可条件のしきい値をIdPとSPがお互い秘密にしたままで，認可判定を行うことができることを示した．またこのプロトコルは，情報を秘密にしたまま行われるにもかかわらず，不正を正しく検証することができることを示した．今後の課題として，$A+B>r$のように複数の属性を扱う式に対するプロトコル設計を行う必要がある．

謝辞 本研究にあたって，Shibboleth設定のご教授と提案方式へのご助言を頂いたNate Klingenstein氏に深く感謝申し上げます．

参考文献

1) Fujiwara, S., Komura, T. and Okabe, Y.: A Privacy Oriented Extension of Attribute Exchange in Shibboleth, *SAINT2007 Workshop on Middleware Architecture in the Internet*, (2007).
2) Yao, A.: Protocols for Secure Computation, *Proc. of FOCS*, pp.160–164 (1982).
3) Rabin, M.O.: How to exchange secrets by oblivious transfer, *Technical report* (1981).
4) 太田和夫, 渡辺治, 黒沢馨: 情報セキュリティの科学 —マジック・プロトコルへの招待, 講談社.

☆ RFC2246: http://www.ietf.org/rfc/rfc2246.txt
☆☆ XML Signature WG, W3C: http://www.w3c.org/Signature/
☆☆☆ XML Encryption WG, W3C: http://www.w3.org/Encryption/2001/