XML Consortium

～ 第6回 XMLコンソーシアムWeek ～

sPlatプロジェクト活動報告

3者間のポリシー伝達

2007年5月21日
アドソル日進株式会社
荒本道隆

sPlatの検討内容

- マルチホップのWebサービス呼び出しにおける、特に中継者で発生する課題を検討
  - WS-Securityの最大のメリットは、End-to-Endのセキュリティ
  - WS-Security関連のミドルウエアは、署名・暗号化したり、署名検証・復号する機能は提供してくれるけど、中継する機能がない
- 暗号化XMLデータの扱い方について検討
- **ポリシーを伝達する場合の課題について検討**

ポリシー伝達方式の分類

- 事前に交換 ←2004年のWebサービス実証部会で実験済み
  - 事前に何らかの方法でポリシーが伝達できていればいい
  - 事前コンパイルを伴う方式（スキーマ変更方式など）が使える
    - 相手ごとに異なるコードを準備する必要がある
    - 変更時は、アプリの再起動が必要
  - ポリシーの異なる相手が複数ある場合、それぞれと交換が必要
- 動的に交換 ←今回の検討テーマ
  - 動的なポリシーの伝達方法が必要
  - SOAPメッセージに対して、ポリシーをどう適用するのか？
  - ポリシーの異なる相手が複数あっても、途中でポリシーが変更になっても、呼び出す毎にポリシーを取得する事で対応できる

WSITの紹介

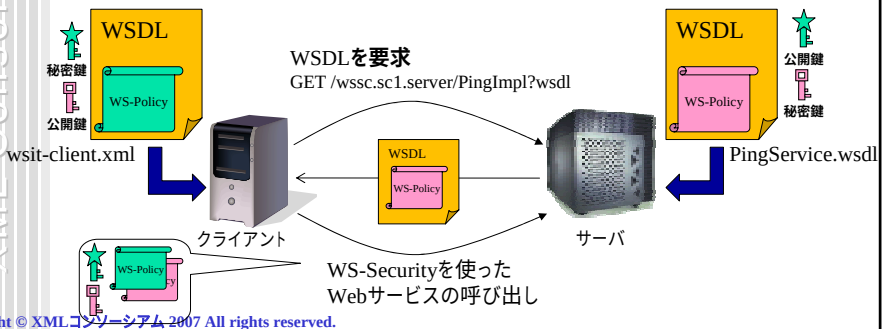
- WSIT(Web Services Interoperability Technologies)
  - WS技術の相互接続性を実現するための活動
    - MicrosoftのWCF（Windows Communication Foundation）との相互運用の確立

  - http://java.sun.com/webservices/interop/
  - 開発者向けサイト https://wsit.dev.java.net/
    - テストケースを動かしてみたい人は、Day(2006/12/12)の資料を参照
      - Milestone3,4は、増えたjarをCLASSPATHに追加すれば、同じ手順で動く

WSITの動的なポリシー交換について

- クライアントから、サーバに対してWSDLを要求
- サーバは、ポリシーの入ったWSDLを加工して、返す
  - KeyStore に関する情報を削除するなど
- クライアントは、自分のポリシーとサーバのポリシーを使って、リクエストを発行

テストケース wssc/sc1 の場合



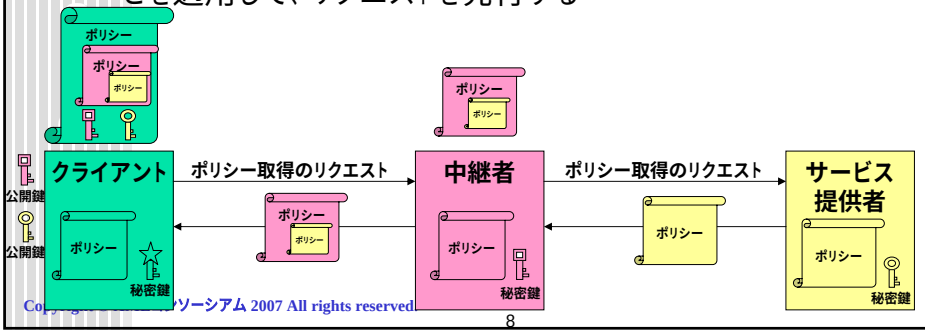
そこで

- このような2点間のポリシー交換の方式を
  3者間での通信に適用した場合について検討しました

ポリシーだけを動的に伝達する場合

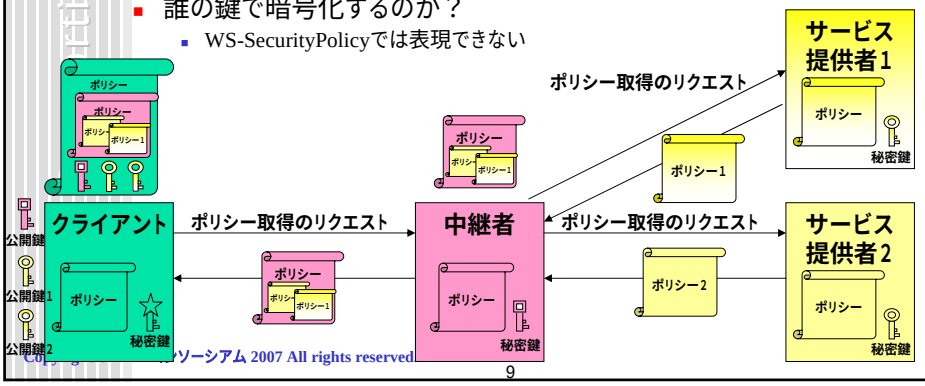
ポリシーの伝達－１

- 中継者は、自身の物とサービス提供者のポリシーをマージ
  - クライアントは、事前に全サーバの公開鍵を持っている
    - どのサービスを使うかは、事前に知っている必要がある
  - ポリシーには署名がしてある
    - そのポリシーが改竄されていないか確認できる
- クライアントは、自身のポリシーと中継者から受け取ったものとを適用して、リクエストを発行する



ポリシーの伝達ー2

- 中継者の課題
  - 複数のポリシーをどう渡すか？
    - マージして渡す or 2つをそのまま渡す
    - 署名検証ができるようにマージする必要がある
- クライアントの課題
  - 誰の鍵で暗号化するのか？
    - WS-SecurityPolicyでは表現できない



ポリシーの例

- サービス側のポリシーの例
  - 署名：BodyとWS-Addressingの各要素
  - 暗号化：Body

```
<wsp:Policy wsu:Id="SecureConversation_MutualCertificate10SignEncrypt_IPingService_Ping_Input_p
 <wsp:ExactlyOne>
  <wsp:All>
   <sp:SignedParts xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
    <sp:Body />
    <sp:Header Name="To" Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
    <sp:Header Name="From" Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
    <sp:Header Name="FaultTo" Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
    <sp:Header Name="ReplyTo" Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
    <sp:Header Name="MessageID" Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
    <sp:Header Name="RelatesTo" Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
    <sp:Header Name="Action" Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
   </sp:SignedParts>
   <sp:EncryptedParts xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
    <sp:Body />
   </sp:EncryptedParts>
  </wsp:All>
 </wsp:ExactlyOne>
</wsp:Policy>
```

ポリシーの例

- マージし易い表記（Xpath指定）で指定する

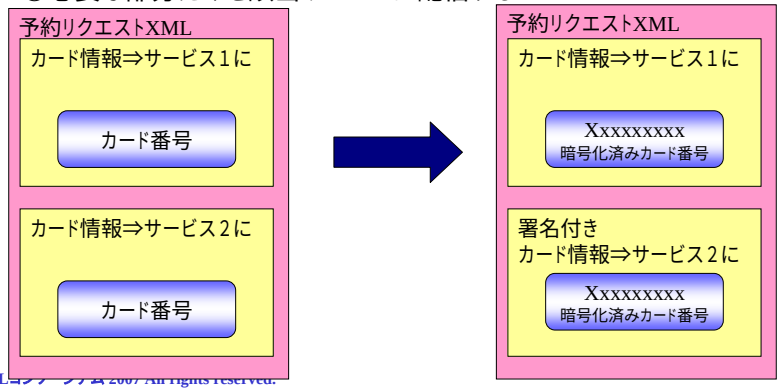
```
<sp:EncryptedParts xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
  <sp:Body />
</sp:EncryptedParts>
```

```
<sp:EncryptedElements xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
  <sp:XPath>/*[local-name() = 'Envelope']/*[local-name() = 'Body'] </sp:XPath>
</sp:EncryptedElements>
```

注意：WSIT Milestone4時点ではサポートされていない
https://wsit-docs.dev.java.net/releases/m4/WSITTutorial.pdf
Encrypted XPath elements may not work but may be fixed in a future milestone.

ポリシーをマージする場合の課題

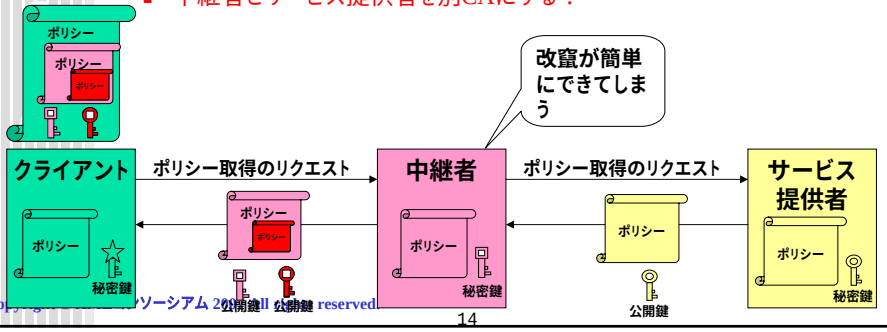
- どのデータが、どのサービスに到達するのかをポリシーで定義できない
  - 例：複数のカード会社のカード番号が、それぞれどこに到達するか
  - ×同じメッセージを全サービスに配信する
  - ○必要な部分だけを該当サービスに配信する



ポリシーと公開鍵を動的に伝達する場合

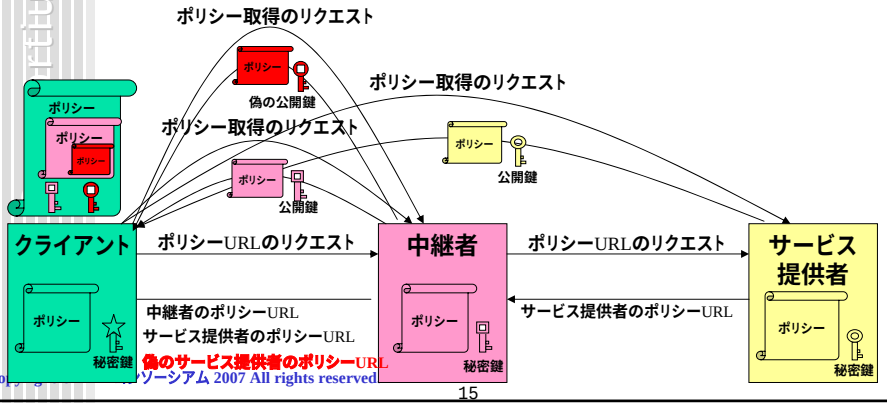
ポリシーと公開鍵の伝達－１

- 中継者がポリシーと公開鍵をマージし、クライアントに渡す
- 中継者が、ポリシーを不正操作したり、公開鍵をすり替えることが出来てしまう
  - 中継者がサービス提供者のURLを通知しても、URLが偽物かも
  - クライアントの公開鍵を使っても、中継者がすり替え可能
  - CAを使っても、中継者もCAの証明書を持っている
    - 中継者とサービス提供者を別CAにする？



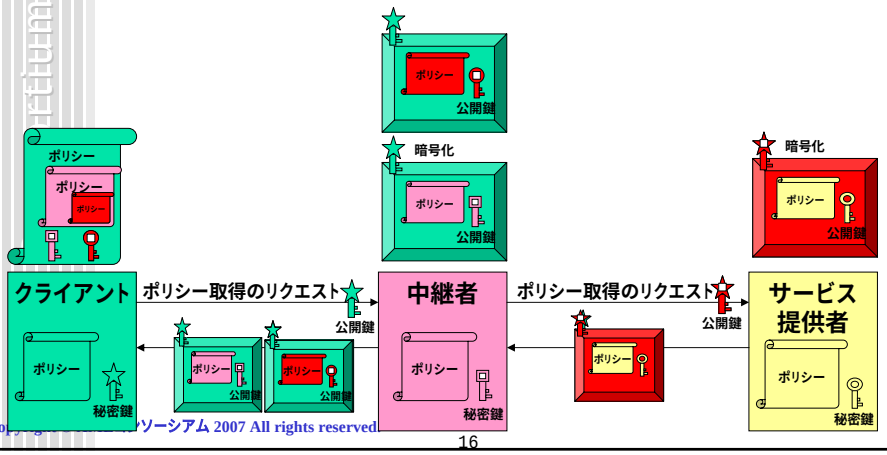
ポリシーと公開鍵の伝達－2

- 中継者がサービス提供者のURLを通知
  - 偽のURLを通知することもできる



ポリシーと公開鍵の伝達－3

- クライアントの公開鍵を送り、暗号化して返す
  - 暗号化すると、中継者でマージできない
  - 中継者が不正操作を行う余地はある

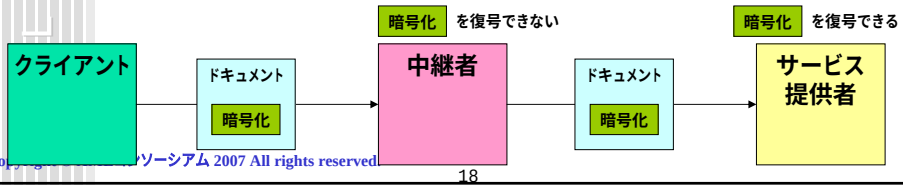


中継者が信頼できなければ…

- クライアントが、事前に公開鍵を持っておく必要がある
  - 中継者が自由にサービス提供者を増やせない
  - クライアントは、公開鍵の更新の手間がかかる
  - サービス自体も、直接、クライアントからサービス提供者を呼んだ方がマシ？
  - 何のための、End-to-Endのセキュリティなんだろう？

End-to-Endのセキュリティ

- ポリシーと公開鍵を伝達する方式
  - 中継者をある程度は信頼しなければならない
    - 中継者が悪意を持って攻撃はしないはず
  - 署名や公開鍵によって、否認は実現できる
    - 被害があった後では遅い場合も
  - 中継者のメリット
    - 自由にサービス提供者を増やす事ができる
    - **「見たくても見れない」ではなく、「見ることが出来ない事が証明される」**
      - 個人情報取扱業者の義務が発生するかどうか？
      - 時間をかければ暗号化は解けるので、「保存しておかない」と約束

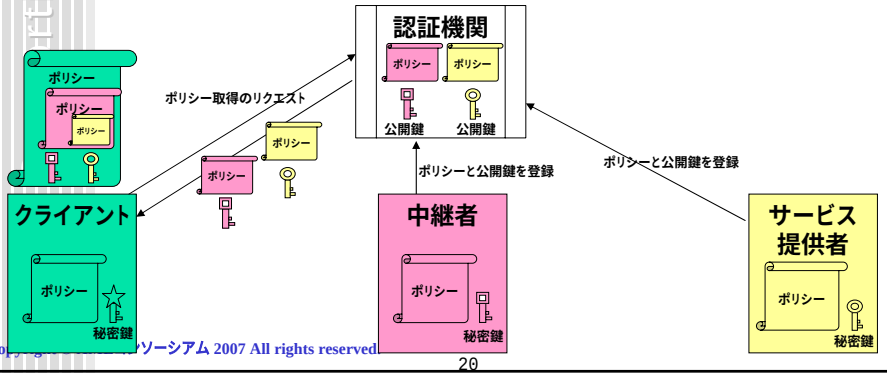


中継者を信頼するのが前提だと

- それで「真のEnd-to-Endのセキュリティ」と言えるのだろうか…
  - **契約によって保護されているだけでは、悪意を持った者に破られる**
- 安心して使えるセキュリティとは
  - **安全であることをシステムが保証する**
  - 例：httpsによるセキュリティ
    - Internetは通信内容を覗かれる可能性がある⇒RSA暗号
    - 相手の確認⇒URLのドメイン名を確認、DNS、CAの証明書
    - 複数が安全性を担保することで、信頼性を高める
  - 「中継者」１人に頼った方式では、セキュリティを高めるのに限界がある

ポリシーと公開鍵を安心して交換するには

- ベリサインのような、独立した認証機関を利用
  - 課題
    - どのポリシーを取得するのかを選択する方法
    - 中継者は、サービス提供者を増やすたびに審査が必要
  - SAML（**S**ecurity **A**ssertion **M**arkup **L**anguage）をうまく使えないか？



まとめ

- End-to-Endのセキュリティで注意すること
  - ポリシーを動的に交換することのメリットとデメリット
  - ポリシーと公開鍵を動的に交換することのメリットとデメリット
  - メッセージ構造やポリシーだけでなく、ビジネス上の契約形態にも注意

- **誰を一番信頼するのか？**
  - サービス提供者を信頼するパターン
    - サービス提供者＝カード会社、などの場合
    - クライアントは、サービス提供者と契約を結ぶ
    - クライアントは、中継者に余計な情報を**『絶対に』**見せたくない
  - 中継者を信頼するパターン
    - 中継者＝旅行企画会社、などの場合
    - クライアントは、中継者と契約を結ぶ
    - 中継者が、サービス提供者を自由に増減できる事がメリットに
  - 認証機関を信頼するパターン
    - クライアントは認証機関と契約を結ぶ
    - 認証機関が、中継者とサービス提供者の関係を審査する必要がある