XML Consortium

～ 第5回 XMLコンソーシアムWeek ～

sPlatプロジェクト
暗号化XMLデータ利用技術についての課題と対策

XML Encryption概要

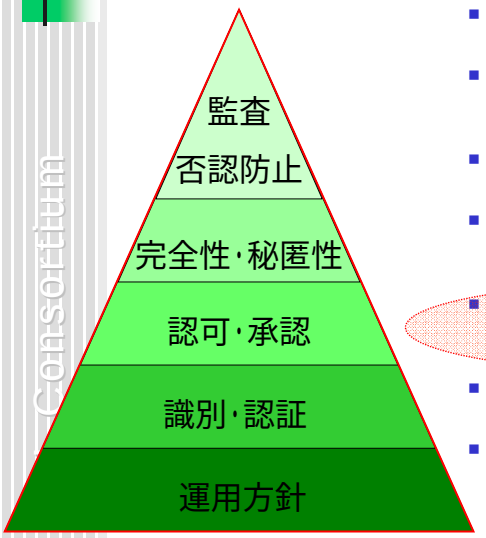
2006年5月24日

XMLコンソーシアム セキュリティ部会
岡村 和英（株式会社ネット・タイム）

アジェンダ

- 暗号化とは
- XML文書の暗号化
  - XML Encryption
- 暗号化の問題点

Webサービスに求められるセキュリティ要件



- 識別 (Identification)
  - サービスの相手を識別する。
- 認証 (Authentication)
  - 識別した相手が、その本人であることを証明する。
- 認可 (承認) (Authorization)
  - 操作の権限を判断する。
- 完全性 (Integrity)
  - 送信されたメッセージと受信したメッセージが同一であることを証明する。
- 秘匿性 (Confidentiality)
  - 送信されたメッセージが盗み読みされないことを保証する。
- 監査 (Accounting)
  - サービスの利用状況を事後確認する。
- 否認防止 (Non-repudiation)
  - 受信されたメッセージが、送信されたメッセージであることを証明する。

2つの暗号化方式

- 共通鍵暗号方式 (対象鍵暗号方式)
  - 暗号と復号で同じ鍵を用いる。
  - 事前に秘密手段によって、鍵を共有しておくことが必要。
  - 演算処理が単純。
  - DES、AES(Rijindael) など
- 公開鍵暗号方式 (非対称鍵暗号方式)
  - 暗号と復号で異なる鍵を用いる。
    - 「公開鍵」と「秘密鍵」のペア
  - 片方の鍵を公開することができ、秘密手段による共有が不要。
  - 演算処理が複雑。
  - RSA、楕円曲線、ElGamal など

- ハイブリッド暗号 (セッション鍵暗号)
  - 共通鍵暗号方式と公開鍵暗号方式の組み合わせ
    - ①「メッセージ」を自動生成した共通鍵で暗号化する
    - ②「暗号化に用いた共通鍵」を公開鍵で暗号化する
    - ③「暗号化された共通鍵」と「暗号化されたメッセージ」を送付する
  - その場での鍵交換により、メッセージ毎／セッション毎に異なる鍵を利用できる。
  - データ本文は高速な共通鍵方式で処理。
  - SSL、PGPなど

XML文書の暗号化

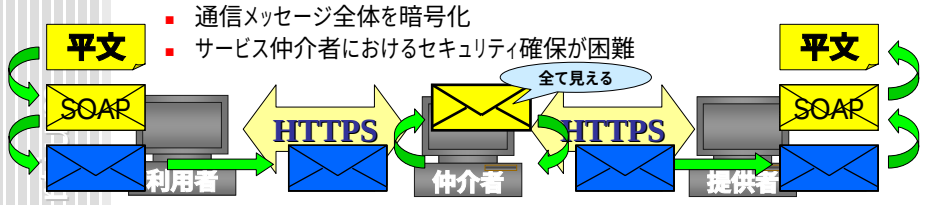
- XML Encryption

  http://www.w3.org/Encryption/2001/

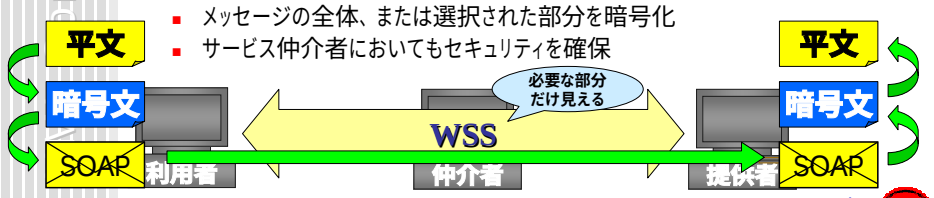
  - 暗号化の対象
    - XML文書全体
    - XML文書の一部分
    - 任意のバイナリデータ
  - 暗号化されたデータのXMLによる表現
    - 復号に必要な情報（アルゴリズム、鍵情報）
    - 暗号化されたデータ本体　または　暗号化されたデータへの参照

SSLとXML暗号の違い

- SSLによるWebサービス
  - トランスポート層による、「Point-to-Point」のセキュリティ
  - 通信メッセージ全体を暗号化
  - サービス仲介者におけるセキュリティ確保が困難



- XML暗号を用いたWebサービス
  - メッセージコンテントによる、「End-to-End」のセキュリティ
  - メッセージの全体、または選択された部分を暗号化
  - サービス仲介者においてもセキュリティを確保



暗号化されたXML文書

```
<enc:EncryptedData xmlns:enc="http://www.w3.org/2001/04/xmlenc#">
  <enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
  <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <enc:EncryptedKey>
      <enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
        <enc:OAEPParams>91Wu3Q==</enc:OAEPParams>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      </enc:EncryptionMethod>
      <ds:KeyInfo>
        <ds:KeyName>okamura@nettime.co.jp</ds:KeyName>
      </ds:KeyInfo>
      <enc:CipherData>
        <enc:CipherValue>xoAg093t1GigaE‥‥dsfa8agi3</enc:CipherData>
      </enc:CipherData>
    </enc:EncryptedKey>
  </ds:KeyInfo>
  <enc:CipherData>
    <enc:CipherValue>j5H0dag4S5sdth8KJaG ‥‥ Gea2rh7r56dkhd=</enc:CipherValue>
  </enc:CipherData>
</enc:EncryptedData>
```

暗号化アルゴリズム

暗号化に用いた共通鍵の情報

鍵交換アルゴリズム

公開鍵の情報

暗号化された共通鍵

暗号化された文書データ

XML文書の部分暗号化

- 任意の部分の暗号化はできない
  - 属性だけの暗号化や、任意のノードセットの暗号化はできない。
- エレメント暗号
  - 指定されたXML要素とその子要素を含めた全体（開始タグから終了タグまで）を暗号化する。
  - 暗号化された要素が<enc:EncryptedData>に置換される。
- コンテント暗号
  - 指定されたXML要素の内容（開始タグの直後から終了タグの直前まで）を暗号化する。
  - 暗号化された要素の内容が子要素<enc:EncryptedData>に置換される。

エレメント暗号

```
<paymentInfo xmlns="http://xmlconsortium.org/sPlat/PaymentSample">
  <name>Kazuhide Okamura</name>
  <amount currency="JPY">10000</amont>
  <creditCard cardType="VISA">
   <cardNumber>4987-0012-3456-7890</cardNumber>
   <expirationDate>2006/05</expirationDate>
  </creditCard>
</paymentInfo>
```

エレメント暗号化

```
<paymentInfo xmlns="http://xmlconsortium.org/sPlat/PaymentSample">
  <name>Kazuhide Okamura</name>
  <amount currency="JPY">10000</amont>
  <enc:EncryptedData xmlns:enc="http://www.w3.org/2001/04/xmlenc#"
            Type="http://www.w3.org/2001/04/xmlenc#Element">
   <enc:CipherData>
    <enc:CipherValue>j5H0da45dt8JaG ‥‥ e2hr5GVd=</enc:CipherValue>
   </enc:CipherData>
  </enc:EncryptedData>
</paymentInfo>
```

<creditCard>要素全体が見えなくなる

コンテント暗号

```
<paymentInfo xmlns="http://xmlconsortium.org/sPlat/PaymentSample">
  <name>Kazuhide Okamura</name>
  <amount currency="JPY">10000</amont>
  <creditCard cardType="VISA">
   <cardNumber>4987-0012-3456-7890</cardNumber>
   <expirationDate>2006/05</expirationDate>
  </creditCard>
</paymentInfo>
```

コンテント暗号化

```
<paymentInfo xmlns="http://xmlconsortium.org/sPlat/PaymentSample">
  <name>Kazuhide Okamura</name>
  <amount currency="JPY">10000</amont>
  <creditCard cardType="VISA">
   <enc:EncryptedData xmlns:enc="http://www.w3.org/2001/04/xmlenc#"
            Type="http://www.w3.org/2001/04/xmlenc#Content">
    <enc:CipherData>
     <enc:CipherValue>sG30N9a8uC3Bt‥‥MkexGa9dgiu2h</enc:CipherValue>
    </enc:CipherData>
   </enc:EncryptedData>
  </creditCard>
</paymentInfo>
```

<creditCard>の要素名と属性はそのまま

暗号化と妥当性検証

- 暗号化後のXML文書の妥当性検証を可能とするべきか？
- XML暗号化後に妥当性を保持し続けることは難しい。
- 妥当性検証を行なうか否かは個々のアプリケーションに任される。
- 妥当性検証を行なうアプリケーションは、以下のいずれかの方法を取ることになる。
  - 暗号化前と暗号化後の両方を許すスキーマを用いる。
    ＝ 暗号化後のＸＭＬ文書は元のスキーマに対して妥当である
  - 暗号化前と暗号化後で別のスキーマを用いる。
    ＝ 暗号化後のＸＭＬ文書は元のスキーマに対して妥当でない
  - 常に復号してから妥当性検証を行なう。
    ＝ 暗号化後のＸＭＬ文書に対する妥当性検証は行なわない