XML Consortium

～ 第6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

スキーマ変換方式による妥当性検証の実現

2007年5月21日

XMLコンソーシアム セキュリティ部会
中山　弘二郎（株式会社 日立製作所）

© XML Consortium 2007

目次

- 背景と目的
- 妥当性検証とデータバインディング
- スキーマ変換方式の詳細
- まとめ

© XML Consortium

目次

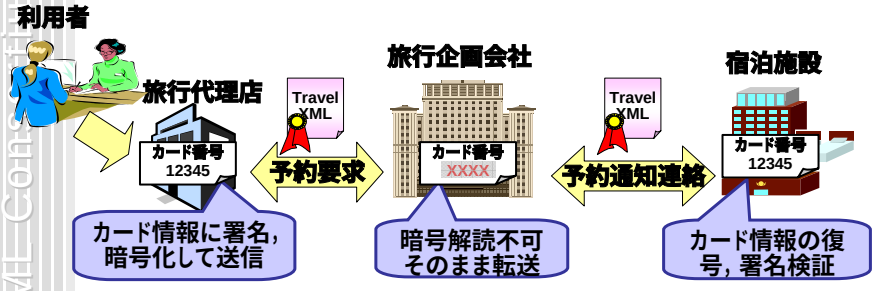
- 背景と目的
- 妥当性検証とデータバインディング
- スキーマ変換方式の詳細
- まとめ

XML Consortium

© XML Consortium

Webサービスのセキュリティ

- 想定するシーン
  - 中継者を介したマルチホップのWebサービス
  - End-to-Endのセキュリティを確保



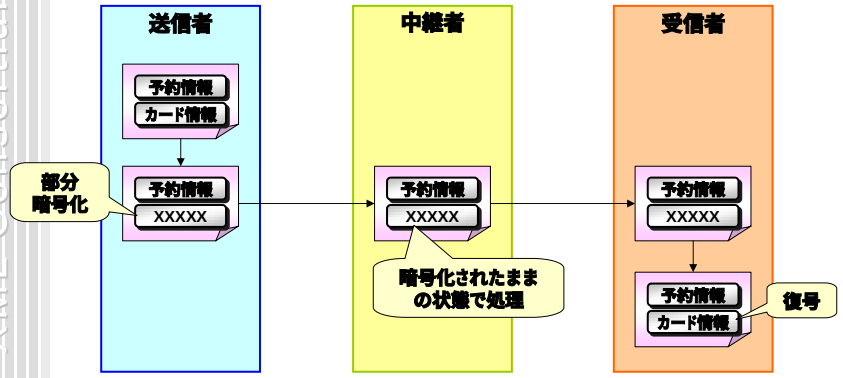
※ XMLコンソーシアム「TravelXML利用Webサービス実証実験プロジェクト公開資料」より
http://www.xmlconsortium.org/koukai/travelxml-p/

XML Consortium

© XML Consortium

暗号化処理の流れ

- XML暗号によるメッセージの暗号化
  - 中継者に開示したくないデータを部分暗号化
  - 中継者に対する秘匿性を確保

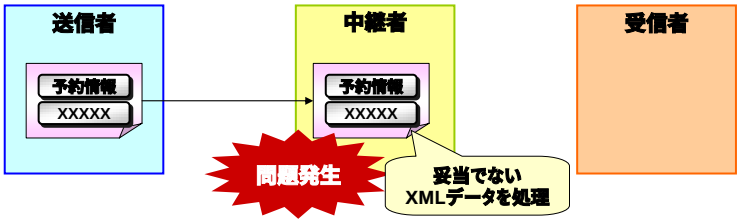


© XML Consortium

課題と目的

- 課題
  - XMLデータの構造はスキーマにより事前に定義されている
  - XMLデータを部分暗号化することで、スキーマに対する妥当性が失われる
    - 中継者はスキーマに対して妥当でないXMLを処理する必要がある
  - XMLプロセッサはXMLデータが妥当であることを期待していることが多い

中継者の処理において問題が発生



- 目的
  - Webサービスの中継者における適切な暗号化データの処理方式を検討、開発、提案する

© XML Consortium

目次

- 背景と目的
- 妥当性検証とデータバインディング
- スキーマ変換方式の詳細
- まとめ

XML Consortium

© XML Consortium

中継者における問題点

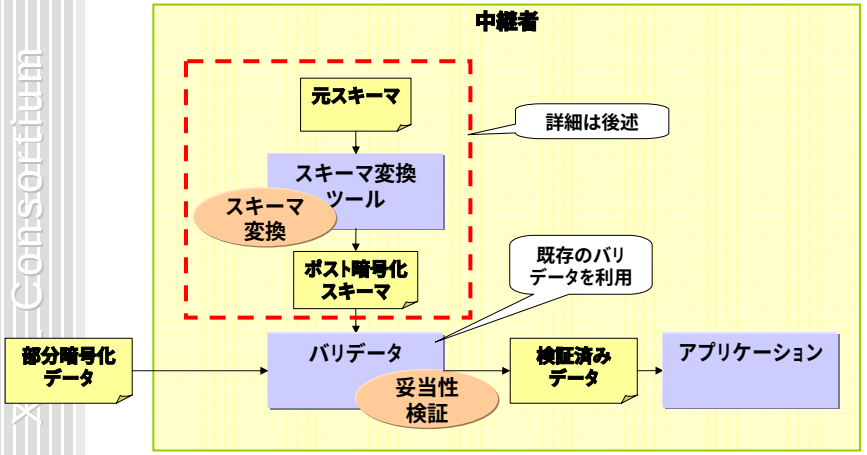
- 妥当性検証
  - 妥当性検証とは，
    - XMLデータがスキーマ定義に従って正しく構成されていることを事前に確認する処理のこと
  - XMLデータが妥当でないと…
    - 妥当性検証時にエラーが発生
- データバインディング
  - データバインディングを使うことで，
    - スキーマ定義からXMLデータに簡単にアクセスするためのAPIを自動生成が可能
  - XMLデータが妥当でないと…
    - XMLデータからオブジェクトへのマッピング（アンマーシャリング）の際にエラーが発生

XML Consortium

© XML Consortium

妥当性検証方法

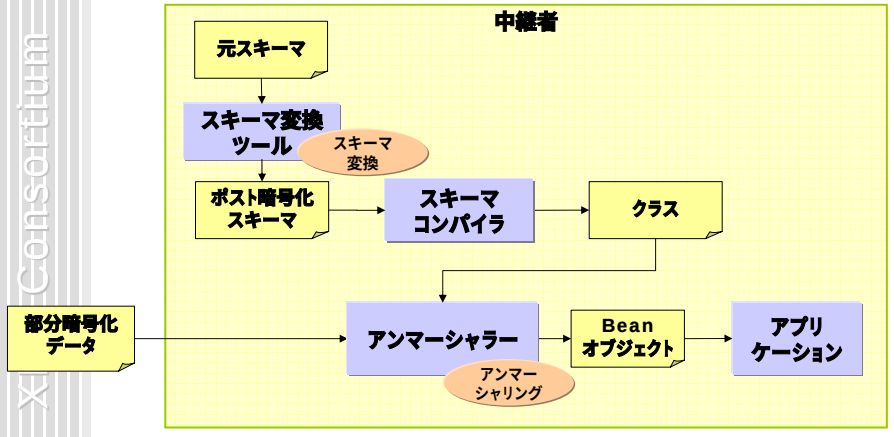
- 暗号に対応したスキーマ（ポスト暗号化スキーマ）に変換
- 既存のバリデータを利用して妥当性検証を実施



© XML Consortium

データバインディング方法（1）

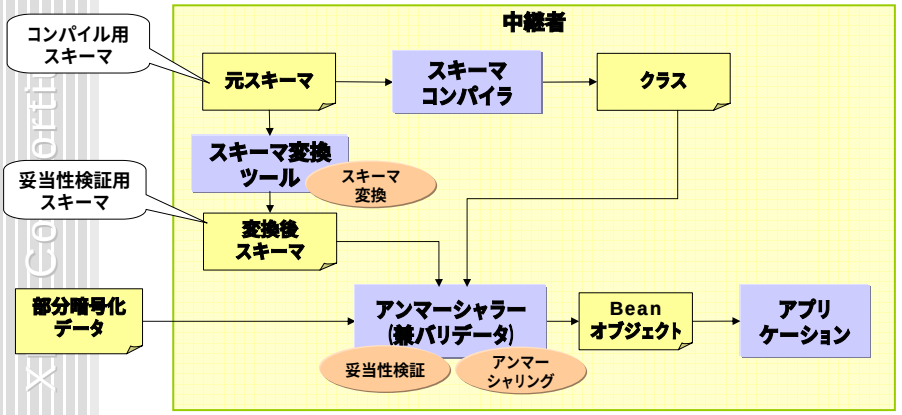
- ポスト暗号化スキーマを使ってクラスを生成



© XML Consortium

データバインディング方法 (2)

- Flexible Unmarshalling (JAXB 2.0)
  - 妥当でないXMLデータのアンマーシャリングが可能
- 元スキーマを使ってクラスを生成
- (必要であれば) ポスト暗号化スキーマを使って妥当性検証も可能



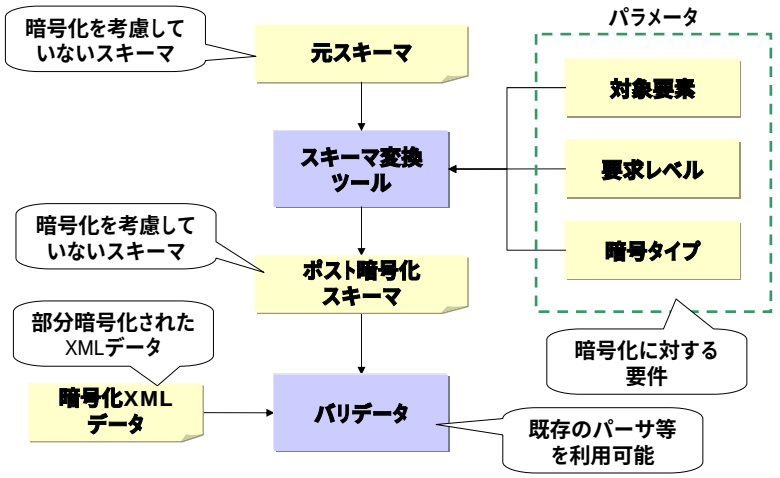
© XML Consortium

目次

© XML Consortium

スキーマ変換の概要

- スキーマ変換により、暗号化に対応したスキーマ（ポスト暗号化スキーマ）を生成する



© XML Consortium

パラメータ

- ポスト暗号化スキーマを生成するために必要な情報をパラメータとして与える

パラメータ	意味	値
対象要素	暗号化の対象となる要素	XPathで記述 （例）/Order/CardInfo
要求レベル	暗号化の要求レベル	・必須（暗号化が必要） ・任意（暗号化してもよい）
暗号タイプ	XML暗号の仕様で規定された暗号化のタイプ	・エレメント暗号 ・コンテント暗号

© XML Consortium

パラメータとポリシー

- Webサービスのセキュリティ要件はポリシーとして公開される
- スキーマ変換のパラメータはポリシーから導き出すことができる

■ WS-Policy / WS-SecurityPolicyによるポリシーの例

暗号タイプが「コンテント暗号」であることを表す

要求レベルが「任意」であることを表す

```
<sp:ContentEncryptedElements    wsp:Optional="true">
  <sp:XPath>S:body/Order/CardInfo</sp:XPath>
</sp:ContentEncryptedElements>
```

対象要素のXPathを表す

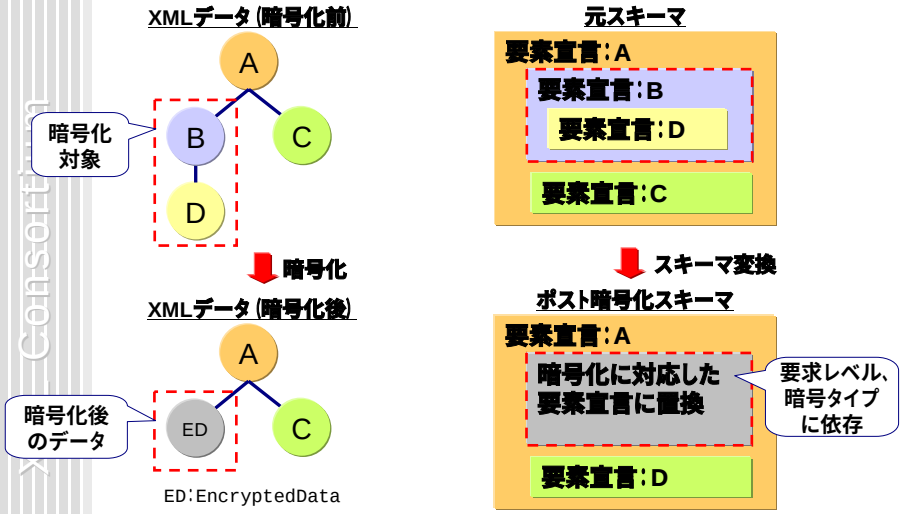
→三者間でのポリシー交換について、次のセッションで説明

※上記のポリシーの例はドラフト版の仕様に基づいたものです。最終的なOASIS標準仕様とは表記方法が異なる可能性があります。

© XML Consortium

スキーマ変換の例

- 対象要素が /A/B の場合



© XML Consortium

要素宣言の置換 (1)

要求レベル ：暗号化が必須
暗号タイプ ：エレメント暗号

✕ XMLデータ (暗号前)

```
<A>
  <B>.....</B>
  <C>.....</C>
</A>
```

暗号化必須

○ XMLデータ (暗号前)

```
<A>
  <enc:EncryptedData> .....</enc:EncryptedData>
  <C>.....</C>
</A>
```

対象要素<B>の代わりに<enc:EncryptedData>が出現

元スキーマ

```
<complexType>
 <sequence>
  <element name="B"...
  <element name="C"...
 </sequence>
</complexType>
```

スキーマ変換

ポスト暗号化スキーマ

```
<complexType>
 <sequence>
  <element ref="enc:EncryptedData" />
  <element name="C"...
 </sequence>
</complexType>
```

EDへの参照に置換

© XML Consortium

要素宣言の置換 (2)

要求レベル ：暗号化が任意
暗号タイプ ：エレメント暗号

○ XMLデータ (暗号前)

```
<A>
  <B>.....</B>
  <C>.....</C>
</A>
```

暗号化任意

○ XMLデータ (暗号前)

```
<A>
  <enc:EncryptedData> .....</enc:EncryptedData>
  <C>.....</C>
</A>
```

暗号化前<B>と暗号化後<enc:EncryptedData>のどちらかが出現

元スキーマ

```
<complexType>
 <sequence>
  <element name="B"...
  <element name="C"...
 </sequence>
</complexType>
```

スキーマ変換

ポスト暗号化スキーマ

```
<complexType>
 <sequence>
  <choice>
   <element ref="B" />
   <element ref="enc:EncryptedData"/>
  </choice>
  <element name="C"...
 </sequence>
</complexType>
```

<choice>により両方を許容

暗号前

暗号後

© XML Consortium

要素宣言の置換 (3)

要求レベル　:暗号化が必須
暗号タイプ　:コンテント暗号

✕ XMLデータ (暗号前)

```
<A>
  <B>
    <C>...</C>
    <D>...</D>
  </B>
  <E>...</E>
</A>
```

暗号化対象

暗号化必須

○ XMLデータ (暗号前)

要素の内容だけを暗号化 (<B>タグは残す)

```
<A>
  <B>
    <enc:EncryptedData>...</enc:EncryptedData>
  </B>
  <E>...</E>
</A>
```

対象要素<b>の子要素として<enc:EncryptedData>が出現

元スキーマ

```
<element name="B">
  <complexType>
    <sequence>
      <element name="C"...
      <element name="D"...
    </sequence>
  </complexType>
</element>
```

スキーマ変換

ポスト暗号化スキーマ

```
<element name="B">
  <complexType>
    <sequence>
      <element ref="enc:EncryptedData" />
    </sequence>
  </complexType>
</element>
```

対象要素の子要素を変更

© XML Consortium

XML Consortium

要素宣言の置換 (4)

要求レベル　:暗号化が必須
暗号タイプ　:コンテント暗号

○ XMLデータ (暗号前)

```
<A>
  <B>
    <C>...</C>
    <D>...</D>
  </B>
  <E>...</E>
</A>
```

暗号化任意

○ XMLデータ (暗号前)

```
<A>
  <B>
    <enc:EncryptedData>...</enc:EncryptedData>
  </B>
  <E>...</E>
</A>
```

対象要素<b>の子要素として
暗号化前の要素か<enc:EncryptedData>が出現

元スキーマ

```
<element name="B">
  <complexType>
    <sequence>
      <element name="C"...
      <element name="D"...
    </sequence>
  </complexType>
</element>
```

スキーマ変換

ポスト暗号化スキーマ

```
<element name="B"
  <complexType>
    <choice>
      <sequence>
        <element name="C"...
        <element name="D"...
      </sequence>
      <sequence>
        <element ref="enc:EncryptedData" />
      </sequence>
    </choice>
  </complexType>
</element>
```

暗号前

暗号後

© XML Consortium

XML Consortium

スキーマ変換における問題

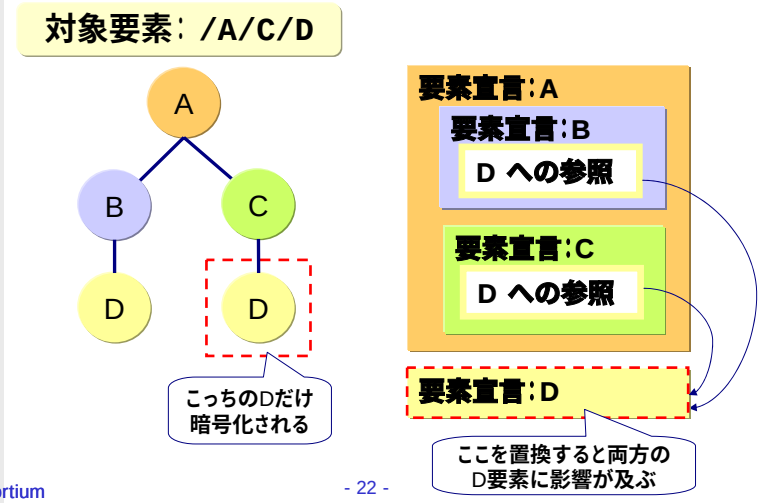
- 置換の影響範囲の問題
- UAP違反の問題

XML Consortium

© XML Consortium

置換の影響範囲の問題

- スキーマ内の参照関係により、想定外の箇所に置換処理の影響が及ぶことがある



XML Consortium

© XML Consortium

UPA違反とは

- UPA (Unique Particle Attribution)
  - XML Schemaの仕様によって規定された、曖昧なスキーマを排除するための制限
  - 妥当性検証時に、対応する要素宣言が先読みすることなく一意に特定できなくてはならない
- UPA違反の例

XMLデータ

```
<A>
  <B>xxx</B>
  <C>yyy</C>
</A>
```

<b>に対応する要素宣言が一意に決まらない

スキーマ（正しくないスキーマ）

```
<element name="A">
  <complexType>
    <sequence>
      <element ref="B" minOccurs="0"/>
      <choice>
        <element ref="B" />
        <element ref="C" />
      </choice>
    </sequence>
  </complexType>
</element>
```

出現回数が 0 or 1回

オーバーラップ

© XML Consortium

UPA違反の問題

- 暗号対象要素が複数ある場合、ポスト暗号化スキーマがUPA違反となることがある

オリジナルスキーマ（UPA違反なし）

```
<element name="A">
  <complexType>
    <sequence>
      <element ref="B" minOccurs="0" />
      <element ref="C" />
    </sequence>
  </complexType>
</element>
```

出現回数が 0 or 1回

■パラメータ
対象要素/A/B、暗号化必須、エレメント暗号
対象要素/A/C、暗号化任意、エレメント暗号

スキーマ変換

ポスト暗号化スキーマ（UPA違あり）

```
<element name="A">
  <complexType>
    <sequence>
      <element ref="enc:EncryptedData" minOccurs="0"/>
      <choice>
        <element ref="enc:EncryptedData" />
        <element ref="C" />
      </choice>
    </sequence>
  </complexType>
</element>
```

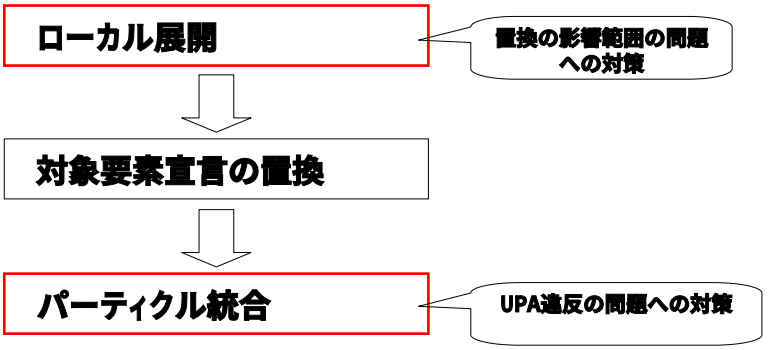
オーバーラップ

UPA違反発生

© XML Consortium

スキーマ変換の処理の流れ

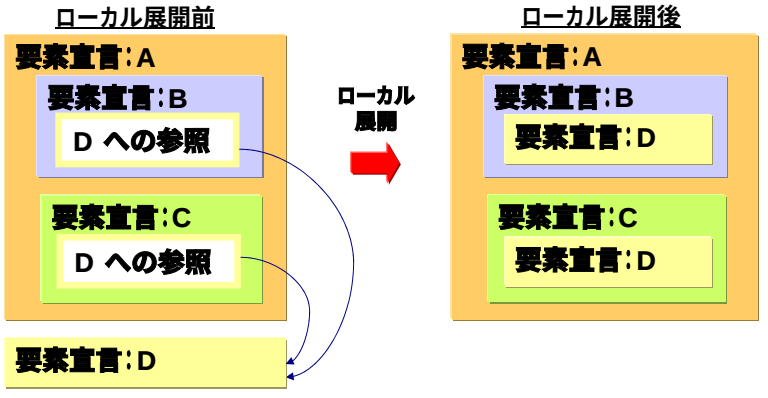
- ローカル展開、パーティクル統合の処理を追加



© XML Consortium

ローカル展開 (1)

- 参照先の要素宣言を、参照元 (ローカル) に展開する

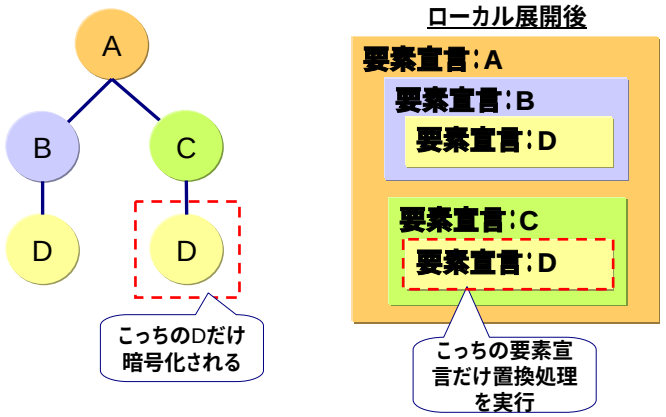


© XML Consortium

ローカル展開 (2)

- ローカル展開後のスキーマに対してスキーマ変換を実行する

対象要素パラメータ：/A/B/D



© XML Consortium

パーティクル統合

- オーバーラップするパーティクル（スキーマを構成する要素）を1つに統合することでUPA違反を回避する

パーティクル統合前（UPA違反あり）

```
<sequence>
  <element ref="enc:EncryptedData" minOccurs="0"/>
  <choice>
    <element ref="enc:EncryptedData" />
    <element ref="C" />
  </choice>
</sequence>
```

element パーティクル

choice パーティクル

パーティクル統合

パーティクル統合後（UPA違反なし）

```
<sequence>
  <choice minOccurs="1" maxOccurs="2">
    <element ref="enc:EncryptedData" />
    <element ref="C" />
  </choice>
</sequence>
```

1つのchoice パーティクルに統合

元スキーマの持つ制約条件が失われることがあるので注意が必要

© XML Consortium

目次

- 背景と目的
- 妥当性検証とデータバインディング
- スキーマ変換方式の詳細
- まとめ

XML Consortium

© XML Consortium

まとめ

- Webサービスの中継者における，適切な暗号化データの処理方式について検討
- 暗号化データの妥当性検証を実現する方法として、スキーマ変換方法の詳細を検討
  - 影響範囲の問題をローカル展開で対策
  - UPA違反の問題をパーティクル統合で対策

XML Consortium

© XML Consortium