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

提案方式の概要

2006年5月24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）

目次

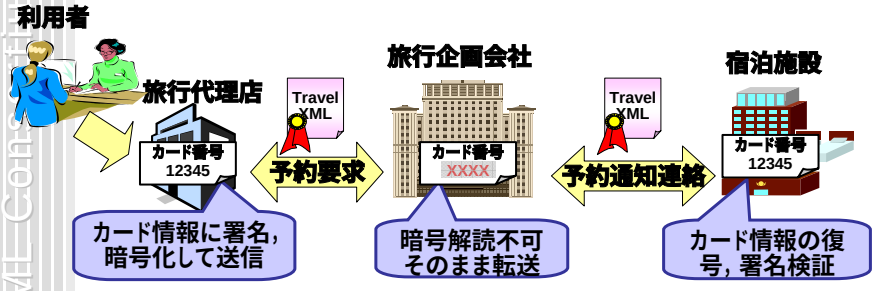
XML Consortium

目次

- 背景 – Webサービスのセキュリティ
- 課題とsPlatプロジェクトの目的
- 提案方式の概要
- まとめ

Webサービスのセキュリティ

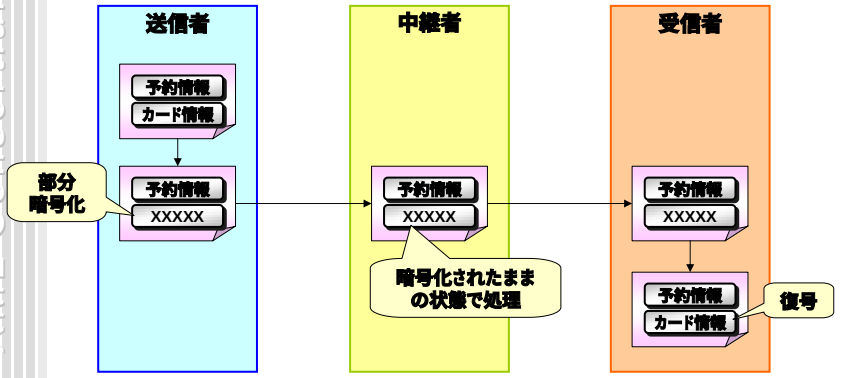
- 想定する利用シーン
  - 中継者を介したマルチホップのWebサービス
  - End-to-Endのセキュリティを確保



※ 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/

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



スキーマ定義の例

- XMLのデータ構造はスキーマ言語で定義される
  - XML Schema，DTD，RELAX NGなど

XML Schemaによる
スキーマ定義

```
<element name="CreditCardInformation">
  <complexType>
    <sequence>
      <element ref="CreditCardAuthority" minOccurs="0" />
      <element ref="CreditCardNumber" minOccurs="0" />
      <element ref="ExpireDate" minOccurs="0" />
      <element ref="CardHolderName" minOccurs="0" />
    </sequence>
  </complexType>
</element>
```

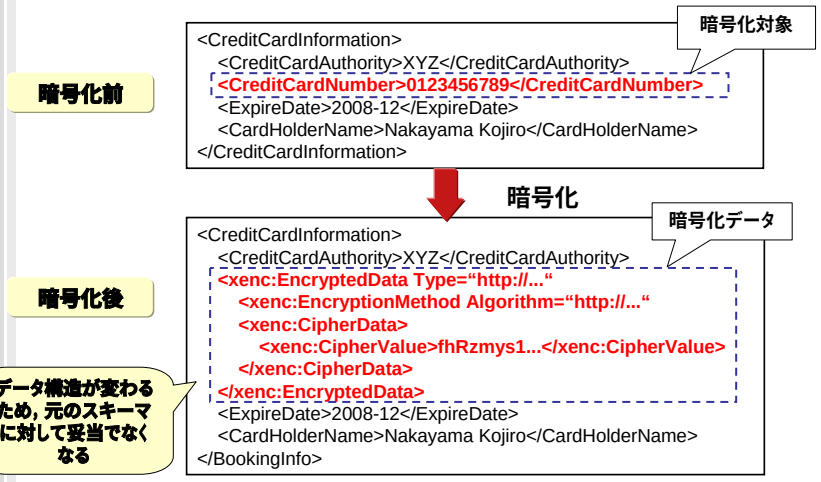
妥当

XML Schemaに対して
妥当なXMLデータ

```
<CreditCardInformation>
   <CreditCardAuthority>XYZ</CreditCardAuthority>
   <CreditCardNumber>0123456789</CreditCardNumber>
   <ExpireDate>2008-12</ExpireDate>
   <CardHolderName>Nakayama Kojiro</CardHolderName>
</CreditCardInformation>
```

XML暗号の例

- XML暗号を用いて，メッセージの一部分だけを暗号化することが可能

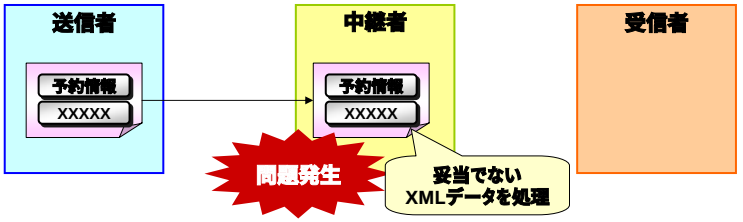


目次

課題

- XML暗号をWebサービスに適用する際の課題
  - 中継者では，スキーマに対して妥当でないXMLデータを扱う必要がある
  - 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



- 中継者の処理における問題点
  - 妥当性検証ができない
  - データバインディングの処理に失敗する

中継者における問題点

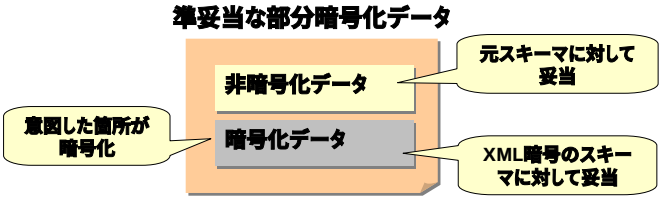
- 妥当性検証
  - 妥当性検証とは，
    - XMLデータがスキーマ定義に従って正しく構成されていることを事前に確認する処理のこと
  - XMLデータが妥当でないと…
    - 妥当性検証時にエラーが発生
- データバインディング
  - データバインディングを使うことで，
    - スキーマ定義からXMLデータに簡単にアクセスするためのAPIを自動生成が可能
  - XMLデータが妥当でないと…
    - XMLデータからオブジェクトへのマッピング（アンマーシャリング）の際にエラーが発生

sPlatプロジェクトの目的

- sPlatプロジェクトの目的
  - 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
- 検討項目
  - 妥当性検証方式
    - 本日の発表で報告
  - データバインディング方式
    - 今後の検討課題

暗号化データの妥当性

- 次の条件を満たす場合，部分暗号化データは準妥当であると呼ぶ
  - 非暗号化部分が，元スキーマに対して妥当
    - 自身の処理で仕様するデータが妥当であることの確認
  - 暗号化部分が，XML暗号のスキーマに対して妥当
    - XML暗号に従い正しく暗号化されていることの確認
  - 意図した箇所が暗号化されている，意図しない箇所は暗号化されていない
    - 適切な箇所が暗号化されていることの確認



妥当性検証方式の要件

- 今回の検討では，以下の要件を満たす妥当性検証を提供することを目標とした

要件①	Webサービスの中継者において適用可能であること
要件②	準妥当性の検証が可能であること
要件③	既存のXMLプロセッサを利用して実現可能であること
要件④	既存の（暗号化を考慮していない）スキーマを用いたシステムでも適用可能であること

妥当性検証方式

- 問題点の整理
  - スキーマ（元スキーマ）に対して
    妥当でないXMLデータ（部分暗号化データ）を，
    XMLプロセッサで処理することで問題が発生する
- 解決方法として以下の3つが考えられる
  - スキーマに工夫をする
  - XMLデータに工夫をする
  - XMLプロセッサに工夫をする
- 本プロジェクトの要件との適合性
  - 要件③：既存のXMLプロセッサを利用可能
    - XMLプロセッサの変更はしない
    - スキーマもしくはXMLデータに工夫をする

妥当性検証方式

- 考えられる妥当性検証の方針※

方式①	元スキーマ作成時に暗号化を考慮する （スキーマに工夫をする）
方式②	元スキーマとは別に暗号化に対応したスキーマを作成する （スキーマに工夫をする）
方式③	復号してから妥当性検証を実施する （XMLデータに工夫をする）

※ W3C “XML Encryption Requirements” http://www.w3.org/TR/xml-encryption-req より

- 本プロジェクトとの要件との適合性
  - 要件①：中継者において適用可能
    - 中継者は暗号化データを復号することができない
      - ➡ 方式③の適用は不可
  - 要件④：既存のスキーマでも利用可能
    - 既存のスキーマでは暗号化が考慮されていないことが多い
      - ➡ 方式①の適用は困難

➡ **方式②「暗号化に対応したスキーマを作成」を採用**

目次

処理の流れ

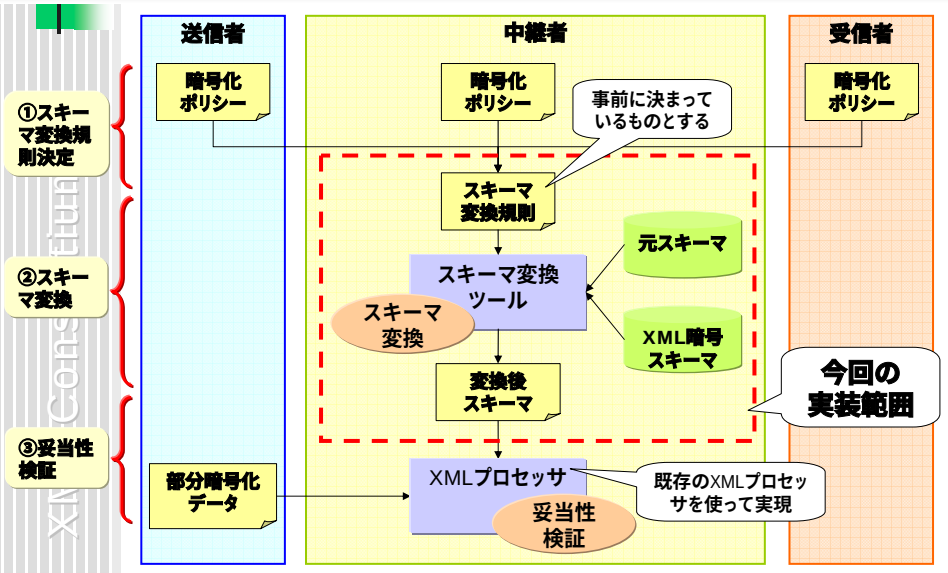
- 提案方式の処理の流れ

① スキーマ変換規則の決定（開発時 or 実行時）
- 暗号箇所や暗号タイプなどから，スキーマの変換規則を決定する
- 様々な方式が考えられる
  - 事前にシステム全体で合意
  - 実行時にセキュリティポリシーの伝達/ネゴシエーションにより決定
- 処理の詳細は未検討

② スキーマ変換（開発時 or 実行時）
- ①で決定したスキーマ変換規則に従い，元スキーマを暗号化に対応したスキーマ（ポスト暗号化スキーマ）に変換する
- 今回，sPlatプロジェクトにて実装

③ 妥当性検証（実行時）
- ②で変換したスキーマを用いて，妥当性検証を実行する
- 既存のXMLプロセッサを使って実現可能

全体像



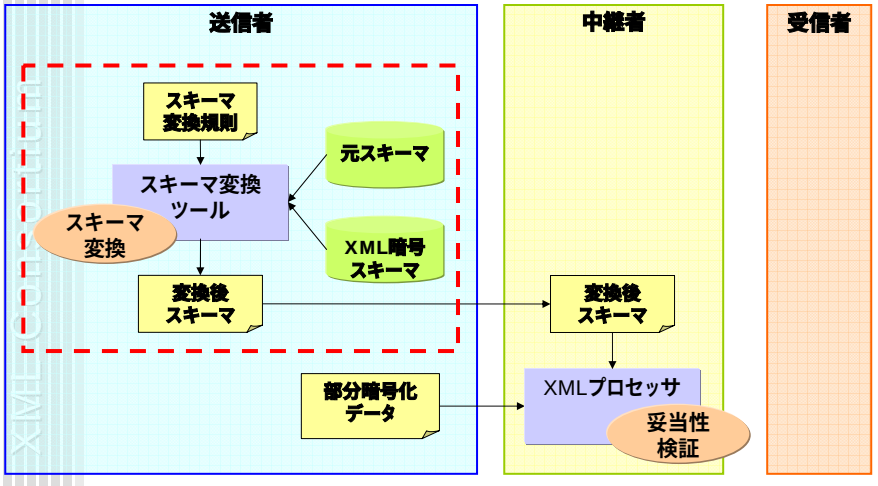
提案方式の変形例

- 想定する利用シーン
  - 暗号化データに関する情報は，一切中継者に開示したくない
    - カード番号を開示しないのは当然だが，暗号化されているデータがカード情報であること自体も開示したくない

    → 暗号化部分のスキーマを中継者に隠蔽する
- 処理の流れ
  - ① 送信者もしくは受信者がスキーマ変換を実施する
  - ② 変換後のスキーマを中継者に送信する
  - ③ 中継者が受信したスキーマを使って妥当性検証を実施
  - ※ スキーマ変換処理自体は前のケースと同じ

提案方式の変形例 (続き)

- 中継者に暗号化部分のスキーマを開示しないケース



スキーマ変換規則

- スキーマ変換規則
  - 許容する暗号化の方式を示したデータ
  - スキーマ変換の前に決定する必要がある
- スキーマ変換規則の内容
  - 暗号対象要素 (e.g. CreditCardNumber要素)
  - 暗号化の要求レベル (MUST or MAY)
  - 暗号タイプ (エレメント暗号 or コンテント暗号)
- 例)
  - CreditCardNumber要素は，エレメント暗号により暗号化されていなければならない
    (CreditCardNumber要素，MUST，エレメント暗号)

スキーマ変換規則 (暗号対象要素)

- 暗号対象の要素を指定する
- スキーマ定義内の変換対象箇所の位置を指定する

```
<element name="CreditCardInformation">
  <complexType>
    <sequence>
      <element ref="CreditCardAuthority" minOccurs="0" />
      <element ref="CreditCardNumber" minOccurs="0" />
      <element ref="ExpireDate" minOccurs="0" />
      <element ref="CardHolderName" minOccurs="0" />
    </sequence>
  </complexType>
</element>
```

ここの部分を指定

- XMLデータ内の暗号対象要素の位置を指定する方法も考えられる
  - 指定された要素からスキーマ定義内の変換対象箇所を特定する必要あり

```
<CreditCardInformation>
   <CreditCardAuthority>XYZ</CreditCardAuthority>
   <CreditCardNumber>0123456789</CreditCardNumber>
   <ExpireDate>2008-12</ExpireDate>
   <CardHolderName>Nakayama Kojiro</CardHolderName>
</BookingInfo>
```

ここの部分を指定

今後の課題

暗号式定義 (要求レベル)

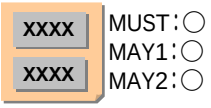
- 暗号化の要求レベルを指定する
- 次の要求レベルを指定可能

MUST	指定した要素が暗号化されていなければならない
MAY	指定した要素が暗号化されていてもよい

- MAYは，暗号対象が複数ある場合の処理に応じて，次の2種類に細分化される

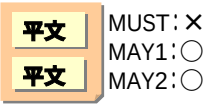
MAY1	暗号化する場合は，全ての暗号化対象要素が暗号化されていなければならない
MAY2	暗号化されている要素と暗号化されていない要素が混在していてもよい

両方暗号化



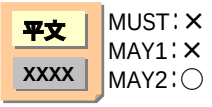
MUST：○
MAY1：○
MAY2：○

両方平文



MUST：×
MAY1：○
MAY2：○

片方だけ暗号化



MUST：×
MAY1：×
MAY2：○

スキーマ変換規則 (暗号タイプ)

- 暗号化のタイプを指定する
- XML暗号で規定されている，次の2種類の暗号タイプを指定可能
  - エレメント暗号
    - 指定したエレメントを丸ごと暗号化
    - 指定したエレメントを<EncryptedData>に置換
  - コンテント暗号
    - 指定したエレメントの要素内容 (コンテント) を暗号化
    - 指定したエレメントの子要素として<EncryptedData>を追加

変換後スキーマの例

- スキーマ変換規則の例
  - 暗号対象　　：CreditCard要素
  - 要求レベル　　：MUST
  - 暗号タイプ　　：エレメント暗号

元スキーマ

```
<element name="CreditCardInformation">
  <complexType>
    <sequence>
      <element ref="CreditCardAuthority" />
      <element ref="CreditCardNumber" />
      <element ref="ExpireDate" />
      <element ref="CardHolderName" />
    </sequence>
  </complexType>
</element>
```

変換

変換後スキーマ

```
<import
namespace="http://www.w3.org/2001/04/xmlenc#"
schemaLocation="xenc-schema.xsd"/>

<element name="CreditCard
  <complexType>
    <sequence>
      <element ref="CreditCardAuthority" />
      <element ref="EncryptedData" />
      <element ref="ExpireDate
      <element ref="CardHolde
    </sequence>
  </complexType>
</element>
```

XML暗号のスキーマをインポート

CreditCardNumberをEncryptedDataに置換

目次

まとめ

- sPlatプロジェクトの目的
  - XML暗号をWebサービスに適用する際の技術課題について検討
  - 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
- 提案方式
  - スキーマ変換規則を決定（詳細未検討）
  - 元スキーマを，暗号化に対応したスキーマ（ポスト暗号化スキーマに）に変換
  - 変換されたスキーマを使って妥当性検証を実施

続いて，実装の詳細についてご説明いたします