

オープンソース長期署名 FreeXAdES 第1回 ～XAdES-BES編～



宮地 (miyachi@langedge.jp)

2016年5月23日

オープンソースのXAdES

xadesjs - JavaScript実装のXAdES

<https://www.npmjs.com/package/xadesjs>

- 最近公開されたPure JavaScriptな新実装
- webcryptoを使って実装されている
- kjurさんがそのうちきつと試して情報が出てくる…

OpenXAdES - エストニアのオープンソース

<http://www.openxades.org/> だったのだけど…

→ <http://www.id.ee/> の下に飛ばされる…

- 昔からあるけどDigiDocの一部になった？
- DigiDocはC/C++/Java版がありソース入手可
- DigiDocは LGPKI v2.1 で公開されている

JavaでフリーなXAdESライブラリ？

XAdES生成はXML署名を使えば簡単!?

- ✓ Java6以降でXMLSignatureをサポート
- ✓ .NETではSignedXmlをサポート

※ Java/.NETでのXAdES提供の日は近い？

依存が少ないXAdES実装があると便利

- ✓ xadesjsもOpenXAdESも使うのが面倒…
- ✓ Java標準機能だけで実装できないか？

長期署名普及と勉強の為に作りますか…

- ✓ Java標準だけでFreeXAdESを作る
- ✓ JNSA勉強会ネタとして1年間かける



FreeXAdES

入門/勉強用にシンプルなXAdESを実装

※ 高度な機能が必要なら弊社製品版のご検討を…(^^;

Java標準機能で実装（他に依存しない）

➤ 簡単でシンプルに使えること。

XAdESレベル毎に勉強会で説明して行く

➤ 本日が**第1回**です！次回からスキルアップTFで。

MPL v2.0 (Mozilla Public License)で公開

<http://mozilla.org/MPL/2.0/>

➤ ソースを公開し、商用利用も可能です。

MPL v2 ライセンス

ソース公開義務	GPL	MPL	BSD
OSS本体への 修正/追加分	○ 公開必須	○ 公開必須	× 公開不要
OSSを利用した プログラム	○ 公開必須	× 公開不要	× 公開不要

私の理解:間違っていたらご指摘ご指導をm(__)m

- FreeXAdESを使うプログラム/システムはソース公開義務無し
- FreeXAdES自体を修正したら**修正部はソース公開義務あり**

※ 可能ならGitHubに修正分を反映ください！

FreeXAdES 公開 (現在BES-β1版)

公開リポジトリ

<https://github.com/miyachi/FreeXAdES>

開発環境

Eclipse IDE for Java Developers

Version : Mars.2 Release (4.5.2)

Java環境 : **Java8** (JDK/JRE 1.8.0)

その他 : JUnit4を利用

※ Java7以前の環境はそのままでは動作しません。

※ GitHubとEclipseの使い方は説明しません。

Java環境とXML署名

XAdESの実装にXML署名とBase64が必要。

機能	Java 5	Java 6	Java 7	Java 8
javax.xml.crypto.dsig. XMLSignature	×	○	○	○
java.util. Base64	×	× ※1	× ※1	○

※1 非標準の org.apache.commons.codec.binary.**Base64** は利用可能。

XMLコンソーシアム セキュリティ部会
「署名ツール検証報告書 2010年01月27日」

Java6と.NETの
XML署名利用
と相互運用性

http://xmlconsortium.org/public_doc/securitytool/SignToolVerificationReport20100127.pdf

XML署名と長期署名XAdES

注:XML署名済みファイルをXAdES化はできない！

機能	XML署名	XAdESレベル
デジタル署名	○	○ XAdES-B (XAdES-BES)
署名証明書保護	▲ ※1	
署名時刻証明	× TS使えない	○ XAdES-T
検証情報保持	▲ ※2	○ XAdES-LT (XAdES-X Long)
長期保管 (長期署名)	× TS使えない	○ XAdES-LTA (XAdES-A)

※1 KeyInfo を参照 (Reference) 追加すれば可能。

※2 証明書認証パスの証明書群は KeyInfo の下に格納可能。

XAdESのXML構造例

Signature (XmlDsig : ルート)

SignedInfo (XmlDsig : 署名情報)

Reference URI="#Sign-Target" (対象参照)

Reference URI="#XAdES-Sign-Atrb" (XAdES参照)

SignatureValue (XmlDsig : 署名値)

KeyInfo (XmlDsig : 鍵情報)

Object (XmlDsig : XAdESオブジェクト)

QualifyingProperties (XAdES : 属性情報)

SignedProperties Id="XAdES-Sign-Atrb"
(XAdES : 署名属性領域-署名証明書ハッシュ値等)

UnsignedProperties (XAdES : 非署名属性領域)

Object Id="Sign-Target" (XmlDsig : 署名対象)

XML署名要素に
XAdES要素を追加
することでXAdES化

非署名属性領域に
ついては**次回以降!**

XAdES のバージョン

大きく分けて**v1.3.2**と**v1.4.1**の実装が必要

v1.4.1は追加要素のみでベースは**v1.3.2**

➤ $v1.3.2 = v1.3.2$

➤ $v1.4.1 = v1.3.2 + v1.4.1$

※ **v1.3.1**は**v1.3.2**とは別の名前空間で別仕様

※ **v1.4.2**は**v1.4.1**のバグ修正で同じ名前空間

v1.4.1の以下の2要素を追加/変更が重要

✓ TimeStampValidationData 新規追加

✓ ArchiveTimeStamp (v1.4.1) 仕様変更

※ **EN化されたETSI最新も基本v1.4.1である。**

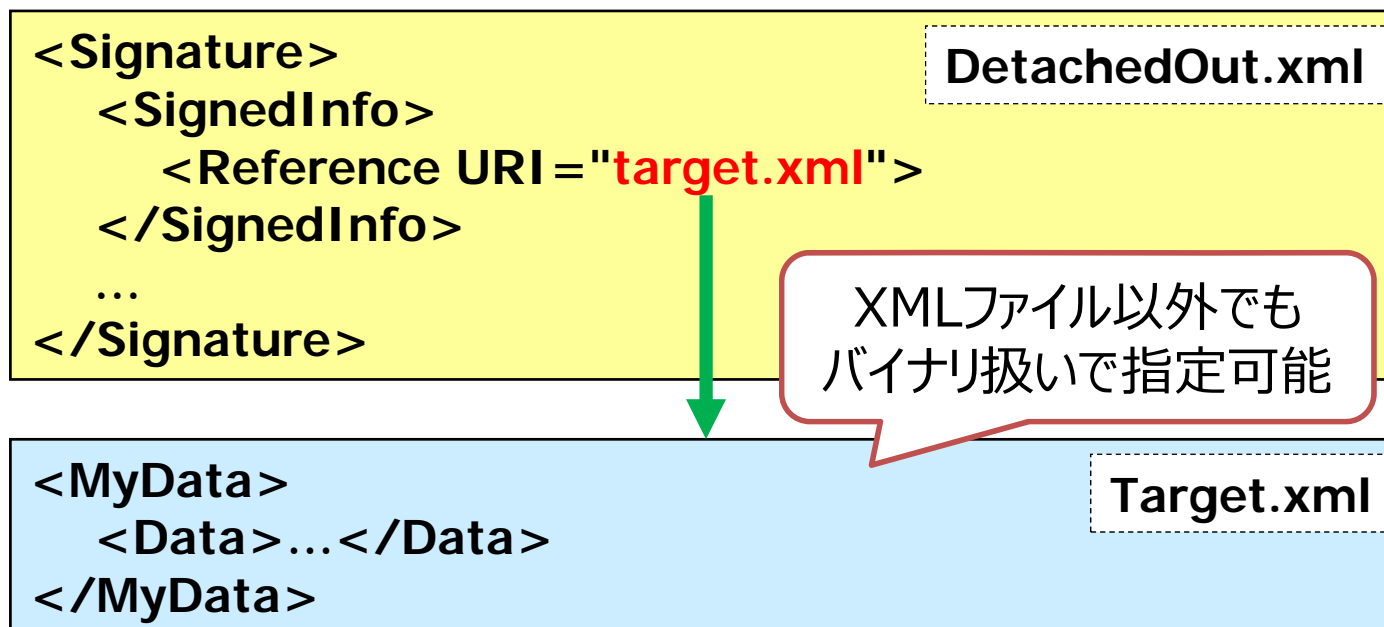
XML署名の署名方式

大きく分け以下4種類で組み合わせ利用も可

署名方式	機能	概要
外部Detached	外部参照	外部ファイルを参照 署名ファイルは別管理 👉 XML以外も参照可能
内部Detached	内部参照	同一XML内を参照 署名の子要素は不可
Enveloping	対象内包	署名対象を署名内に Objectとして含み参照 👉 XML以外も利用可能
Enveloped	埋め込み	一般のXML情報の中に XML署名要素を埋め込む 👉 複数Envelopedは不可

外部Detached (外部参照)

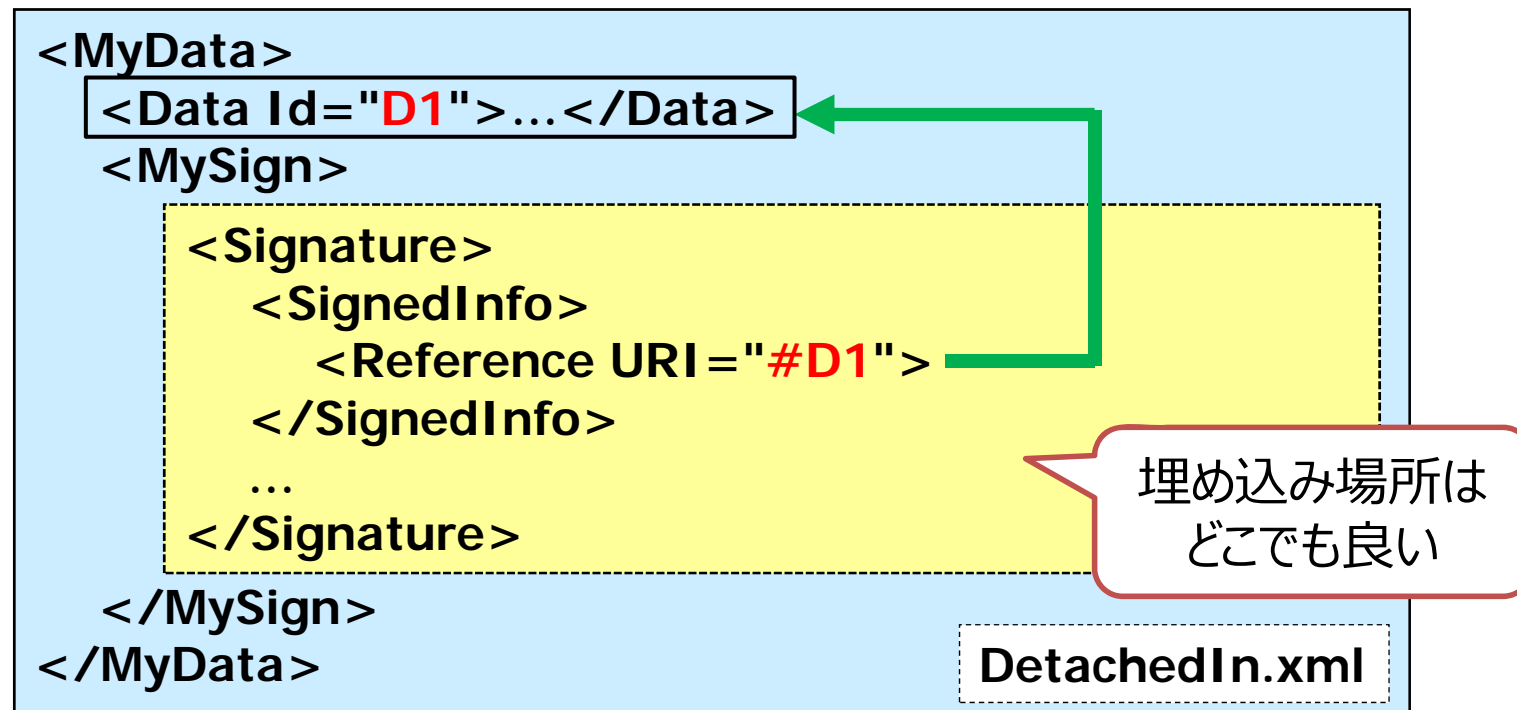
URI指定 (間接可) で外部ファイルを参照



- 署名ファイル自体が小さくシンプルなので良く使われている
- バイナリ扱いによりXML以外に何でも署名対象に可能
- × 署名ファイルと署名対象は別ファイルとして別管理が必要

内部Detached（内部参照）

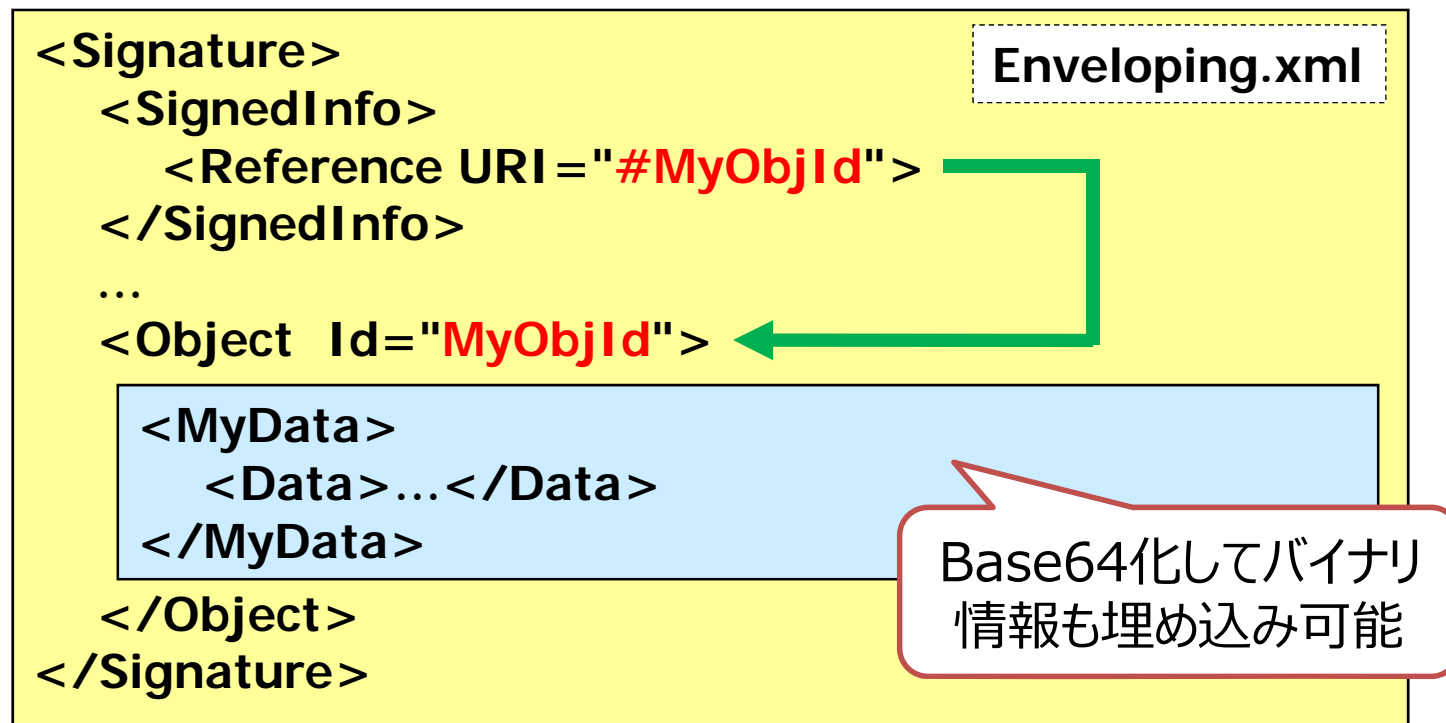
同一XMLファイル内の別要素をId指定で参照



- 署名対象のXML構造を崩さずに署名の埋め込みが可能
- 複数Detachedにより複数の署名対象を指定可能
- × XML要素のみ署名対象として可能

Enveloping (対象内包)

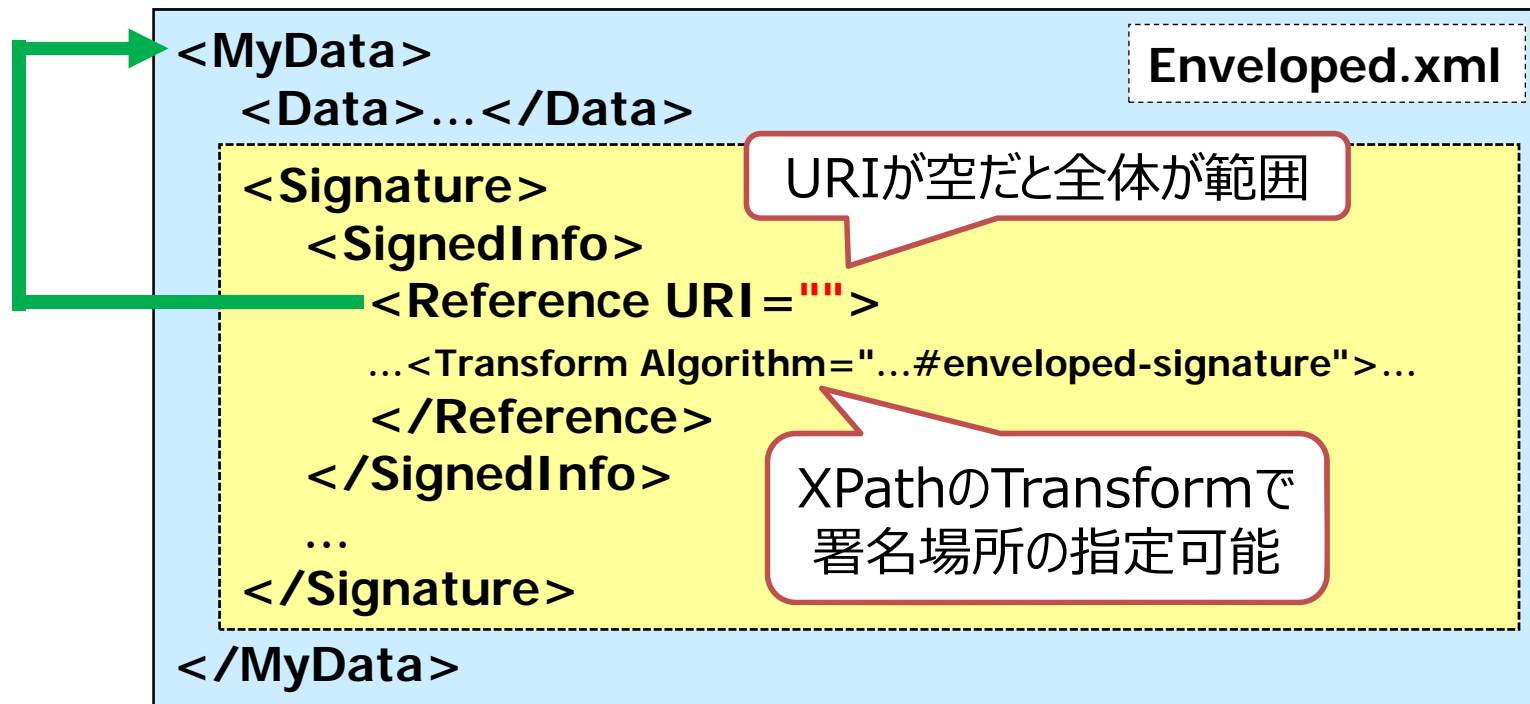
署名対象をObject要素下に含み参照



- 電子封筒として複数の署名対象を1つのファイルにできる
- Base64化によりXML以外に何でも署名対象に可能
- × 署名がメインであり内包されたデータの利用が少し面倒

Enveloped (埋め込み)

署名対象XMLの中にXML署名を埋め込む



- 署名対象のXML構造を崩さずに署名の埋め込みが可能
- 比較的良く使われている（考え方がシンプル）
- × Enveloped可能な署名対象は1つで、複数是不可

XML署名の署名方式まとめ

元のXML構造をそのまま生かして署名したい

- **内部Detached**を選択（Id名は固定にする）
一部のみか複数要素を対象にするなら内部Detachedが良い
- **Enveloped**を選択（基本全体が署名対象）
ほぼ全体を対象にするならEnvelopedが良い

大量の外部ファイルにまとめて署名したい

- **外部Detached**を選択（別管理が必要）

1つの署名ファイルに全てを入れて管理したい

- **Enveloping**を選択（バイナリは大きくなる）

※複数署名方式を組み合わせることも可能!

FreeXAdES 利用サンプル

JUnit4の **IFreeXAdESTest.java** 参照!

testDetachedOut	: 外部Detached試験
testDetachedIn	: 内部Detached試験
testEnvelopingXml	: Enveloping試験(XML)
testEnvelopingBase64	: Enveloping試験(Binary)
testEnveloped	: Enveloped試験
testVerify	: 検証試験(共通 ※)

※ 検証試験は testDetachedIn で失敗する為に簡易実装。

各試験で利用する入力ファイルと生成された
XAdESファイルは test フォルダ下にある。

FreeXAdES 利用手順例 (署名生成)

1. インスタンス生成・初期化設定

- setRootDir / setHashAlg (オプション)

2. オプション : XMLの読み込み

- setXml / loadXml (内部DetachedかEnvelopedのみ)

3. 署名対象の追加 (複数呼び出し可)

- addDetached / addEnveloping / addEnveloped

4. XAdES署名実行 (P12ファイル指定)

- execSign

5. XAdES署名結果の取得/保存

- getXml / saveXml

FreeXAdES オプション

✓ ルートディレクトリ指定

- 外部Detachedのファイル位置ベース指定可

✓ ハッシュアルゴリズム指定

- SHA256(標準)/SHA512/SHA1指定可

✓ C14N正規化アルゴリズム指定

- TRANS_C14N (標準)/TRANS_C14N_EX指定可

✓ XAdESオブジェクトを追加しない

- NO_XADES_OBJ (標準オフ) 指定するとXML署名

✓ SigningTime 出力しない

- NO_SIGN_TIME (標準オフ)

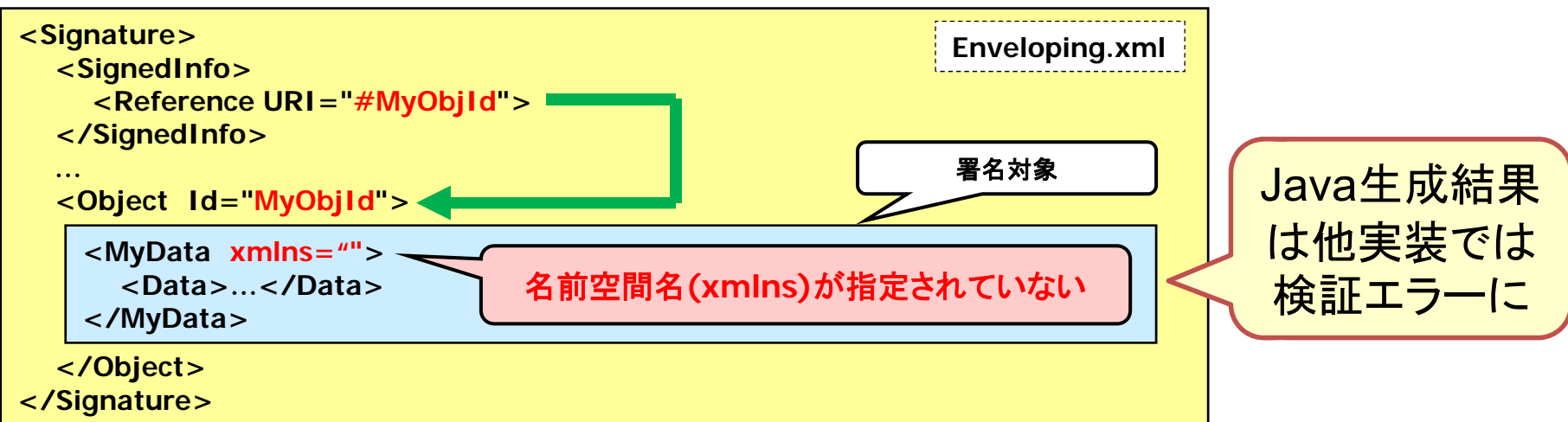
SHA384は
XML Signature
が未サポート

Java8 の XMLSignature 問題1

Enveloping参照先のハッシュ計算が異常！
Java6（2010年）の時代からあった。

「互換性、課題と対策 ～XML署名ツール検証報告～」
XMLコンソーシアムWeek2010 発表資料

http://xmlconsortium.org/seminar09/100310-11+16-18/data/100316/20100316week-wgsec-3_2-signtool.pdf



名前空間無し Enveloping

● 署名対象のオリジナルXML

```
<Object Id="MyObjId" xmlns="http://www.w3.org/2000/09/xmldsig#">  
<MyData xmlns=""><Data Id="D1">book</Data</MyData>  
</Object>
```

● 正しいハッシュ値の正規化後のXML（注：改行が追加されています）

```
<Object xmlns="http://www.w3.org/2000/09/xmldsig#" Id="MyObjId">  
<MyData xmlns=""><Data Id="D1">book</Data></MyData>  
</Object>
```

● Javaハッシュ値の正規化後のXML（注：改行が追加されています）

```
<Object xmlns="http://www.w3.org/2000/09/xmldsig#" Id="MyObjId">  
<MyData><Data Id="D1">book</Dat></MyData>  
</Object>
```

MyData要素の空名前空間 xmlns="" が省略されている...
※ 名前空間に何か指定すれば問題無くなるので回避可能。

名前空間/Id指定あり Enveloping

- 署名対象のオリジナルXML

```
<Object Id="MyObjId" xmlns="http://www.w3.org/2000/09/xmldsig#">  
<MyData Id="D1" xmlns="http://testns"><Data>book</Data></MyData>  
</Object>
```

- 正しいハッシュ値の正規化後のXML（注：改行が追加されています）

```
<Object xmlns="http://www.w3.org/2000/09/xmldsig#" Id="MyObjId">  
<MyData xmlns="http://testns" Id="D1"><Data>book</Data></MyData>  
</Object>
```

- Javaハッシュ値の正規化後のXML（注：改行が追加されています）

```
<Object xmlns="http://www.w3.org/2000/09/xmldsig#" Id="MyObjId">  
<MyData Id="D1" xmlns="http://testns"><Data>book</Dat></MyData>  
</Object>
```

MyData要素の名前空間属性とId属性の順番が逆だぞ...

Java8 の XMLSignature 問題2

内部Detachedのハッシュ計算に失敗する！

Java6では**正常に動作**していた。

Java6と同じソースでJava8ではエラーに。

Java8署名時も**検証時もエラー**になってしまう。

- **FreeXAdES署名時は自分でハッシュ値計算**
- **FreeXAdES検証時はまだ未実装**

※ 何かAPIを追加する必要があるのかも…

Javaのバグとして報告しなきゃ…orz

FreeXAdES BES-β1版の課題

1. 内部Detachedの署名値検証に失敗する
 - 署名は対応済みなのでToDo項目です。
2. 署名用の秘密鍵/証明書がPKCS#12のみ
 - P8の署名ツール検証報告書を見よ！
3. XAdES要素（署名属性） 一部のみ実装
 - SigningCertificate/SigningTime のみ。
4. 署名したインスタンスでそのまま検証できない
 - 面倒なので検証できないように変更予定。
5. 速度やメモリ利用量の調整等はしていない
 - まあこれはおいおい確認して対応で…

以上で第1回は終了！

次回

**第2回～XAdES-T編～
「タイムスタンプを使おう！」**

