3.どうやって使うのか？
～XML暗号化検証報告～

XMLコンソーシアムWeek　2009年05月12日

セキュリティ部会
大沼啓希（日本アイ･ビー･エム株式会社）

© XML Consortium 2009

ご説明内容

- MOF2008合同デモシステム向けセキュリティ報告書
- MOF2008合同デモシステム向けセキュリティ報告書　目次
- オープンソースのツールによる暗号化実践
- (2009/01/09での) 今後の計画
- 2月から4月までの活動
- 実際の活動
- XML暗号化検証報告
- XML暗号化検証報告まとめ

XML Consortium

MOF2008合同デモシステム向けセキュリティ報告書
http://www.xmlconsortium.org/

XML Consortium Top - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 検索 お気に入り

アドレス(D) http://www.xmlconsortium.org/ 移動

XML コンソーシアム

入会申込

HOME | 会員の窓 | セミナー情報 | プレスリリース | XML 関連情報 | ご紹介 | Introduce

現在の会員数…154会員
- 会員一覧・アライアンスパートナー一覧
- 会員製品一覧

入会のご案内
- パンフレット(PDF 25KB)

部会活動
- セキュリティ部会
  活動要綱
  開催案内:第5回 11/19(水)
  公開ブログ
- Webサービス実証部会
  活動要綱
  開催案内:第6回11/19(水)
  公開ブログ
- Web2.0部会
  活動要綱
  開催案内:第6回 12/10(水)
  公開ブログ
- SOA部会
  活動要綱
  開催案内:第5回 11/20(木)
  公開ブログ
- ビジネス・イノベーション研究部会
  活動要綱
  開催案内:第5回 11/20(木)
  公開ブログ
- クロスメディア・パブリッシング部会

トピックス、主催イベント、関連イベント

■トピックス、主催イベント

・1月9日(金)
第10回XMLコンソーシアムDay

・12月18日(木)
XMLコンソーシアムセミナー
「エンタープライズマッシュアップ、次世代クライアント製品徹底検証」(Part2)

・12月17日(水)
XML設計技術勉強会(第七回)<会員限定>

・12月2日(火)
「XMLコンソーシアム エバンジェリストブログ」を開始

・11月28日(金)
XMLコンソーシアムセミナー
「エンタープライズマッシュアップ、次世代クライアント製品徹底検証」(Part1)

・11月20日(木)
XML設計技術勉強会(第六回)<会員限定>

・11月10日(月)
メルマガ発行 → <会員専用>、<一般公開版>
◆メルマガ購読方法:<会員会社の方>、<会員会社以外の方>

・10月31日(金)
XMLコンソーシアムセミナー
～XMLマスタープロフェッショナル(アプリケーション開発)直前対策セミナー～

Webサービス関連
- 技術解説書
- 開発ガイド
- 観光情報Webサービス実証プロジェクト資料
- TravelXML利用Webサービス実証実験プロジェクト成果資料
- 道路交通情報Webサービスを使った複合Webサービス実証実験成果資料

公開資料
- XMLDBに関する質問回答集 New
- MOF2008合同デモシステム向けセキュリティ報告書 New
- XML利用実態俯瞰図 製造情報連携フォーラム
- SCF2007 デモシステム向けセキュリティ検討報告書
- 「エンタープライズ・システムのためのWeb 2.0」提言書
- Web Services Security 日本語訳
- TravelXML 仕様勧告案
- ContentsBusinessXML 仕様勧告案
- XMLQueryUseCases(W3Cワーキングドラフト日本語訳版)
- XML PathLanguage (XPath) 2.0 (W3Cワーキングドラフト日本語訳版)
- ContactXML version 1.1a

インターネット

© XML Consortium 2009

MOF2008合同デモシステム向けセキュリティ報告書

http://www.xmlconsortium.org/public_doc/mof2008_security/mof2008.html

MOF2008合同デモシステム向けセキュリティ報告書 - XMLコンソーシアム - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 検索 お気に入り

アドレス(D) http://www.xmlconsortium.org/public_doc/mof2008_security/mof2008.html 移動

XML コンソーシアム 入会申込

HOME

MOF2008合同デモシステム向けセキュリティ報告書

本報告書は、製造業の技術標準化団体の連携によるフォーラムイベントであるMOF2008（IA懇談会主催）で行われた連携デモシステムに対して、XMLコンソーシアムセキュリティ部会が、セキュリティ技術の適用を検討し、報告したものです。

報告書 PDF

概要

2007年の製造情報連携フォーラムSCF2007デモシステム向けセキュリティ検討(PDF)に引き続き、MOF2008 (Manufactureing Open Forum 2008)の実証デモシステムにおけるセキュリティを対象にしています。
2007年の検討では、システム間連携を行うデモシステム全体に対して、必要となるセキュリティ要件を網羅的に検討しましたが、今回は実装面も含めた、より具体的な情報提供を目的に検討を行っています。

検討メンバー：

- 大沼啓希（日本アイ・ビー・エム株式会社）
- 中山弘二郎（株式会社日立製作所）
- 舟木康浩（日本セーフネット株式会社）
- 松永豊（東京エレクトロン デバイス株式会社）
- 宮地直人（有限会社ラング・エッジ）

利用条件

本書は、本書に記載した要件・技術・方式に関する内容が変更されないこと、および出典を明示いただくことを前提に、無償でその全部または一部を複製、翻案、翻訳、転記、引用、公衆送信等して利用できます。なお、全体を複製、翻案、翻訳された場合は、本書にある著作権表示および利用条件を明示してください。

ページが表示されました インターネット

MOF2008合同デモシステム向けセキュリティ報告書
目次

1. 概要
1.1. 対象システム
1.2. 検討の内容
2. XMLメッセージの保護に必要なセキュリティ技術
2.1. 概要
2.2. 暗号化 (XML暗号)
2.3. 電子署名 (XML署名)
2.4. SOAPへの適用 (WS-Security)
2.5. 長期署名 (XAdES)
3. XMLメッセージの保護の実践
3.1. オープンソースのツールによる暗号化と電子署名
3.2. ハードウェアを使った暗号化と電子署名
3.3. 長期署名によるXMLデータの保管
4. まとめ - 各手法の比較
4.1. ソフトウェアとハードウェアの実装比較
4.2. ハードウェアセキュリティモジュール(HSM)について
4.3. XMLセキュリティのツール

私の担当分

© XML Consortium 2009

オープンソースのツールによる暗号 実践

- XML暗号、XML署名は、商用製品やオープンソースなど様々なソフトウェアにおいて実装されている。代表的なオープンソース実装である**Apache XML Securityを用いてXML暗号を実践する。**
- Apache XML Securityは、Apache Software FoundationのApache XMLプロジェクトによって開発されているXML暗号・XML署名の実装である。Apache XML Securityは、当プロジェクトのWebサイト(http://xml.apache.org/security/)によりJavaおよびC++の実装が公開されているが、**Java版の実装について実践する。**

中山弘二郎（株式会社日立製作所）さんがまとめ役

XML Consortium

© XML Consortium 2009

オープンソースのツールによる暗号実践

- XML暗号･XML署名を実践するためのJavaの標準APIは、それぞれJSR 106 (XML Digital Encryption APIs) とJSR 105 (XML Digital Signature APIs) で規定されている。JSR105についてはすでに最終版の仕様が公開されておりJava SE 6にもその実装が取り込まれているが、JSR106はまだ仕様策定途中である。そのため、これらのJavaの標準APIではなく、Apache XML Securityの独自APIを用いたコード例について説明する。

XML Consortium

(2009/01/09での)今後の計画

- Security　ツール調査、ツール検証
  - ツール調査
    - アプリケーションサーバー、ライブラリ、ゲートウェイ/ESB、テストツール
    - この分野は2005年5月のXMLコンソーシアムWeekにおける発表内容の更新
    - 以下の機能を持つツールの調査

      XML暗号･復号機能
      XML電子署名付与機能
      XML電子署名検証機能
      WS-Security(Web Services Security)機能
      ･･････
  - ツール検証
    - MOF2008合同デモシステム向けセキュリティ報告書での「XMLメッセージの保護の実践」の継続検証
    - プラットフォーム (Windows,　Linux,　･･)
    - Java6対応状況
    - Webサービス実証部会との協業

XML Consortium

2月から4月までの活動 (2月)

- 第7回ミーティング議事録
  - 2009年2月20日(金) 14:00～17:00
    参加者 大沼、松永(記)
    2.ツール検証の打ち合わせ (Webサービス実証部会と合同)
    ・Weekまでにはプラットフォーム+ツールの組み合わせ一種類は検証して報告したい。
    ・3月末までに、ユースケース検討を終える。
    暗号化の要件、対象と方式の洗い出し
    暗号化のスクリプト(フローチャート)作成とその検証方法
    標準仕様の確認
    ・その後、プラットフォームとツールを決めて検証作業
    Webサービス実証部会に実施をお願いする。
    ・使用するサンプルはContactXMLのサンプルがいいのではないか。

XML Consortium

2月から4月までの活動（3月・4月）

- 第8回ミーティング議事録なし
  - **2009年3月18日(水)** 遅れて参加
    参加者 宮地、大沼、松永
    2.ツール検証の打ち合わせ **(Web**サービス実証部会と合同**)**
    ・検証叩き台
- 第9回ミーティング議事録なし
  - **2009年4月16日(木)　14:00～**
    参加者 宮地、大沼、松永
    ツール検証の作業が進んできました。
    検証状況の確認と進め方の相談を行います。

XML Consortium

XML暗号対象

ネストされた構造
もしくはRoot全体

```
<?xml version="1.0" encoding="UTF-8"?>
<ContactXML version="1.1" creator="http://www.pfu.co.jp/meishi-app/xml/contact/1.0" xmlns="http://www.xmlns.org/2002/ContactXML">
 <ContactXMLItem lastModifiedDate="2008-08-24T14:10:59+9:00">
  <PersonName>
   <PersonNameItem xml:lang="ja-JP">
    <FullName pronunciation="ﾏﾂﾔﾏ ﾉﾘｶｽﾞ">松山 憲和</FullName>
    <LastName pronunciation="ﾏﾂﾔﾏ">松山</LastName>
    <FirstName pronunciation="ﾉﾘｶｽﾞ">憲和</FirstName>
   </PersonNameItem>
  </PersonName>
  <Occupation>
   <OccupationItem xml:lang="ja-JP">
    <OrganizationName pronunciation="ﾋﾟｰｴﾌｭｰｿﾌﾄｳｪｱ">ＰＦＵソフトウェア株式会社</OrganizationName>
    <JobTitle>プロジェクト部長</JobTitle>
    <Department>第三ソフトウェア開発部 </Department>
   </OccupationItem>
  </Occupation>
  <Extension>
   <ExtensionItem xml:lang="ja-JP" extensionType="Extended" name="PFU-Department1">プロダクト統括部プロダクト開発部</ExtensionItem>
  </Extension>
  <Address>
   <AddressItem locationType="Office">
    <AddressCode codeDomain="ZIP7">929-1192</AddressCode>
    <FullAddress xml:lang="ja-JP">石川県かほく市宇野気ヌ９８－２</FullAddress>
    <AddressLine xml:lang="ja-JP" addressLineType="Prefecture">石川県</AddressLine>
    <AddressLine xml:lang="ja-JP" addressLineType="City">かほく市宇野気</AddressLine>
    <AddressLine xml:lang="ja-JP" addressLineType="Number">ヌ９８－２</AddressLine>
   </AddressItem>
  </Address>
  <Phone>
   <PhoneItem usage="Official" phoneDevice="Phone">(076)283-5800</PhoneItem>
   <PhoneItem usage="Official" phoneDevice="Fax">(076)283-5905</PhoneItem>
  </Phone>
  <Email>
   <EmailItem usage="Official" emailDevice="Unknown">matsuyama.nori@pfu.fujitsu.com</EmailItem>
  </Email>
 </ContactXMLItem>
</ContactXML>
```

複数の構造

単一の要素

単一の構造

複数の要素

テキストのみ

XML Consortium

叩き台に追加した要件

2009/3/18

1. セキュリティー部会から「XML暗号対象」の叩き台を提出
2. Webサービス部会から、以下検証要件で合意
   - AES+3DESで暗号化する
   - KEK (Key Encryption Key)は外部ファイルで交換
   - <KeyInfo>は<EncryptedData>に入れる
   - XML暗号化の実装

     ApacheXMLSecurity (Java)
     (http://santuario.apache.org/Java/index.html)

     JSR-106 (IBMSDK60に内蔵)

     MicroSoft .Net Framework3.5(C#)

XML Consortium

実際の活動

- **2009/04/06 11:40**
  - 荒本@アドソル日進です。
    Apache実装を使用した**Java**版の検証用ツールが、とりあえず動くようになりました。
    検証の仕様を書いてある**PowerPoint**の2ページ目の図の「**XML**暗号対象」の、上から順にそれぞれ暗号化して、
    e1.xml：**Root**全体
    e2.xml：**PersonName**＋**Occupation**（複数の構造）
    e3.xml：**ExtensionItem**（単一の要素）
    e4.xml：**AddressItem**（単一の構造）
    e5.xml：**PhoneItem**（複数の要素）
    e6.xml：**EmailItem**（テキストのみ）
    というファイル名で保存し、それをそれぞれ復号して **d1.xml** などのファイルに保存するようにしました。
- **2009/04/06 12:14**
  - おつかれさまです。上村です。
    また素早いですね。
    IBM 実装の方ですが、一応、暗号化と復号はできたのですが、複数要素はまだ試していません（作っていない**...**）し、いろいろ整理します。
    何か **Apache** の **API** とはノリが違いますね**...**頭が追いついてません。
    最終的に荒本さんにいただいたツールと仕様を合わせますので、もう少し時間をください。

XML Consortium

実際の活動

- **2009/04/17 15:45**
  - 荒本@アドソル日進です。
    XML暗号化･復号について、**.net**でのコーディングについては、ここが参考になります。
    http://dotnetslackers.com/articles/xml/XMLEncryption.aspx
    このサンプルは**AES+RSA**ですが、**RSACryptoServiceProvider → TripleDESCryptoServiceProvider** とすれば、**AES+3DES**でできます。
    注目の**KeyInfo**は、中に入っています。(上記**URL**中のサンプル参照)
    4/16 (木) の部会で「こんな**API**があったら使い易いのに」と言っていたのは、このぐらいの粒度でしょうか？
    対象要素とアルゴリズムを指定して、「暗号化しろ」とか「復号しろ」というメソッドを叩くだけなので、
    暗号化は6行、復号は4行で書けちゃってます。
    確かに、ほとんどの用途はこれで済んでしまうかも。
    当然、細かい指示をしたければできますし。
    3DESの鍵を文字列で渡せるようにすれば、**Java**版との相互テストができるだろうから、
    そこまで完成したら、**SNS**にアップする予定です。
- **2009/04/17 17:03**
  - 上村@キヤノンソフト情報システムです。
    ***** 荒本さんのおかげで、せっかく実装が3つそろうことですし、**JSR 106** の方でも鍵を中に入れた形で出力するよう変更することにします。
    ソースコードをご覧になった方は分かると思うのですが、ルート要素の暗号化の場合は、ほとんど私が直接 **DOM** を操作して鍵を含めているので、最悪でも同じことをすれば良い訳ですし (本当はもっと正しいやり方があるはずですが**...**) 。
    ***** それから復号鍵の見つけ方ですが、やはりサンプルにあったガチの固定のままではさすがによろしくない、と思います。そこで、**javax.xml.crypto.KeySelector** を使った、とりあえず真っ当だと思われる方法を実装してみたいと思います。これから作業するつもりです。そのため、これから出力結果は変わってしまいますが、とりあえず現時点での出力結果を以下の **URL** で参照できるようにしておきました。ページの一番下の方にある、 **e1.xml ～ e6.xml** と **kek** が現時点での出力結果です。

XML Consortium

実際の活動

- **2009/04/18 20:11**
  - 荒本@・・・・・
  - ただし、**.net**版で暗号化したものは、**Apache**版ではそのままでは復号できませんでした。
    復号しようとすると「**Illegal key size**」というエラーが出てしまいます。
    KeyInfoに入っている、**AES**で使う鍵サイズを比較すると、以下のように全然違います。
- **2009/04/18 21:59**
  - 上村@散髪後です。
    ・・・
  - ちょっと試してみましたので、取り急ぎご報告を。
    .net 版を復号しようとすると荒本さんが言われていたのと同じ例外になります。
- **2009/04/18 23:11**
  - 上村@ネット放浪中です。
    IIBMJCEFIPS を探しています。どのようにして入手するものなのでしょうか？
- ・・・・・・

XML Consortium

XML暗号化検証報告

- ApacheXMLSecurity(SunJDK), JSR-106(IBMSDK), .net のXML暗号の実装を使ってみて（荒本@アドソル日進です。）
  - 各実装のサンプルプログラムは、XML暗号の仕様の一部のみを実現しており、あくまでもサンプルである
    特に復号については、どんな暗号化データにも対応できているわけではない
    復号するパターン（KeyInfoの位置など）に対応した実装をそれぞれ行う必要がある
    XML暗号化したデータを交換する場合には、暗号化後のパターンを詳細に決めておいた方がよい ←※重要
  - 各実装は、APIの粒度がバラバラ
    ApacheXMLSecurity, JSR-106, .net でAPIの使い方が全然違う
  - 複数の要素を復号する場合には、注意が必要
    復号するたびに<EncryptedData>が減るので、ループで処理する場合には要注意
  - ApacheXMLSecurityやJSR-106はデフォルトでAES128を使うが、.netはデフォルトでAES256を使う
    SunJDKやIBMSDKでAES192やAES256を使うには、JCE（Java Cryptography Extension）を入れる必要がある
    使えるアルゴリズムや鍵強度がバラバラの場合、最大公約数のものを使わなければならない ←アタリマエだけど重要
  - JSR-106(IBMSDK)は、パディングの実装が違う
    ApacheXMLSecurityや.netで暗号化したものは、AESの処理単位ピッタリのサイズのみ復号できた →『[wg-sec/evaluation.ml 00047]』参照
  - 3つの実装間で、実際に暗号化したXMLデータを交換して復号したことは、是非、発表資料に入れて欲しいです。
    別々の実装をまたぐ上で注意が必要だったのは、KEKの共有です。
    AES強度やパディングの件は、相互データ交換をする上で非常に重要なので、
    私（荒本）や上村さんのメールを参考に、それぞれ別ページを割いて説明した方が良いと思います。
    パディングについては、一般的なパディングの話と、ここについて説明してはどうでしょう？
    http://www.w3.org/TR/xmlenc-core/#sec-Alg-Block「Padding」の部分

XML Consortium

XML暗号化検証報告

- おつかれさまです。上村です。
ちょっと出遅れてしまったようですね。とりあえず、荒本さんからいただいた部分に、私からも少し付け加えさせていただきます。細かいところですので、発表では遠慮なく端折っちゃってください。
> 特に復号については、どんな暗号化データにも対応できているわけではない
はい、JSR 106 では特にこの影響が大きいと思います。
実際の鍵がどこにあるか？は以下の URL にある KeySelector の実装を用意する必要があり、パターンを限定できないと相当な手間だと思います。頑張れば、結構対応できるとは思いますが（リファレンスリストなど、約束はありますので）。
http://www.ibm.com/developerworks/java/jdk/security/60/secguides/xmlsecDocs/api/javax/xml/crypto/KeySelector.html
実際、Apache とは違い、EncryptedData 要素に一緒に鍵が含まれているパターンすら、自分で実装しないと鍵を特定できません。ただし JSR 106 の実装によっては、最初から便利なパターンを一緒に作っておいてくれることは期待してもいいのかもしれません。（今回は、ちょっと探してみて見たが見つからなかった、標準規格の作法を守ってみる？という理由で、とりあえず作ってみましたが…）
> ・各実装は、APIの粒度がバラバラ
この辺りは、呼び出し側とライブラリの中で、どちらも同じ DOM を触るとなると出てきてしまいそうな問題ですよね。
暗号化に関わらず、API の決め方が難しそうなところではあります。
> ・JSR-106(IBMSDK)は、パディングの実装が違う
パディングの仕様的には、PKCS5 は ISO10126 に完全に含まれてしまうものなので、結構微妙ですね。
「ISO10126 であればOK」という約束のところへ、「PKCS5 のものを送る」のはセーフですが、
「ISO10126 であればOK」という約束のはずなのに、「PKCS5 ではないから NG」として弾くのはちょっとやりすぎだ、という話だと思うので。

XML Consortium

XML暗号化検証報告

- おつかれさまです。上村です。
それから、別のメールにあったところにも少し言及しますが。
>単一の暗号化キーで、複数の要素若しくは構造を暗号化する時は、必ずファイルによるKEK共有をしなければならないと言う意味でしょうか？
これは、文書の中に暗号鍵がすべて使える形で含まれていたら暗号化していないのと同じなので、暗号鍵を暗号化してから文書に含めるわけですが、その後者の暗号化に使う鍵は、当然文書に入れてはいけません、ということだと思います。
これは、文書内で使っている鍵が1個かどうかとは関係なく、発生する問題ですね。
>それとも、「別々の実装をまたぐ上」では、別のKEK共有の課題があるのでしょうか？
文書の外に置いた鍵は、別の方法で相手に届ける必要があるのですが、この鍵のデータはどういうフォーマットなら相手は理解してくれるのか？という課題だと思います。
この部分は、アプリケーションレベルのプロトコルのレベルの問題になり、XML 暗号化の規格とは違う話になります。
例えば、同じ Apache のライブラリを使用した製品やツールがいくつかあるとして、それぞれに簡単な暗号化･復号ツール（コンソールプログラムもあればGUI のものあるかも）が付属しているとします。
そのとき、この外の鍵を読み込ませるのに、「DER形式のファイル名を引数に指定…」とか「テキストエリアに PEM 形式の文字列をコピペしろ（コメントの部分も含めてな！）」とか、いろいろ考えられる訳ですから…
>必ずPaddingをしなければならないのでしょうか？
>かなり、面倒な処理と思うのですが？（Sampleでは、そのようなことはしていなかったような気がします）
これは XML の暗号化とは関係なく、ブロック暗号で一般的な事情ですね。ブロック暗号ならアルゴリズムには関係なく、いつも発生します。XML 暗号の規格では、とりあずデータの暗号化に AES の鍵長違いの3種類が挙げられているので、AES の話みたいになっていますが…（鍵のラップはまた別のようですし）
ブロック暗号の出力は元データの長さに関係なくブロック長の倍数になりますから、
最後のブロックの空きのところ（ブロック長ずつ切り分けて暗号化していき、元データの最後の一切れを入れて余った空間）を、あるルールを持ってキチンと埋めないと、元データの長さが分からなくなってしまいます。
ストリーム暗号ではこの問題はありませんが、なぜか XML 暗号の仕様の中では、使えるアルゴリズムが特に指定されてはいない様です。（なぜでしょう？）
使っちゃいけない訳ではなさそうですが、URI はきちんと決めてもらわないと、相互に運用できないですよね…

以上です。参考になれば幸いです。

XML Consortium

XML暗号化検証報告まとめ

復号 ＼ 暗号化	Apache	JSR-106	.net
Apache	◎	◎	○*1
JSR-106	○ *3	◎	○ *2 *3
.net	○ *4	◎	◎

◎：サンプルプログラムのままで復号できた
○：対応策を入れることで復号できた
*1：JCE (Java Cryptography Extension) を入れてAES256に対応
*2：ポリシーファイルを入れ替えて、AES256に対応
*3：暗号化時に ISO10126Padding ではなく PKCS5Padding を指定
*4：暗号化時に /EncryptedData/KeyInfo/EncryptedKey/KeyInfo/KeyName を追加

XML Consortium

Webサービス実証部会の皆様

- 有難うございました
- 詳細な報告書は、・・・

XML Consortium