3N-03

XML 文書に対する署名の検討

杉江修*** 中村俊介* 鈴木春洋*** 安田孝美** 横井 茂樹*

*名古屋大学大学院 人間情報学研究科

**名古屋大学 情報文化学部

***（株）シーティーアイ SI 事業部

1. はじめに

近年，インターネットやイントラネットにおいてますます大量の情報がやりとりされるようになり，電子文書のフォーマットの違いが問題視されるようになってきている。その中で，XML(Extensible Markup Language)の標準化がW3C によってすすめられ，昨年から今年にかけては企業の業務システムへの導入も加速し，基盤技術として大変な注目を集めている。

しかし，XML における，S/MIME や PGP に相当するような署名の枠組みは現在，W3C と IETF の合同ワーキンググループで話し合いがおこなわれている段階[1]で，実際に動作する参考実装も極めて少ない。

われわれは，XML 署名を使ったシステムの検討を行うために，ワーキングドラフトに沿った署名ツールの実装を行った。

2. 標準化動向と研究の位置付け

W3C による XML Signature のワーキングドラフトは頻繁に改訂されており，精力的な改訂作業が行われている。

また，文書の同一性を保証し，署名のために欠かせない正規化(canonicalization, C14N)の手法についても検討されており，正規化については参考実装が，Ed Simon[2]，James Clark[3]，丸山[4]によっておこなわれている。

ただし，XML 署名自体の実装は丸山[5]によって DSA 署名の実装がおこなわれているだけである。

われわれは広く世の中に普及している RSA 署名を採用して，XML 署名の実装を試みた。

Signing to XML document.

Osamu SUGIE***, Shunsuke NAKAMURA*, Shunyo SUZUKI***, Takami YASUDA**, Shigeki YOKOI*

*Graduate School of Human Informatics Nagoya University

**School of Informatics and Science Nagoya University

***CTI Co., Ltd.

2. 問題点とその解決策

XML は急速に広まっているが，中身は平文であるために常に改ざんの不安がつきまとう。したがって著作権保護や文書の正当性保証など，セキュリティ面で検討すべき課題は多い。

当然，S/MIME や SSL，IPsec 等により，下位層で暗号化等を行う解決策もあるが，構造化されている XML 文書の利点が損なわれてしまう。

XML 文書を XML 文書のままで署名を付加することによって，これらの問題を解決できると考えている。

3. システム概要

われわれのシステムは以下の特徴を持つ。

- PKCS#12 から鍵情報を取得
  広く普及している PKCS#12 形式のファイルから自分の秘密鍵と電子証明書を取得する

- RSA 署名を利用
  暗号ライブラリ AiCrypto[6]を利用して RSAwithSHA1 を使用する。

- 電子証明書を XML 文書に添付
  公開鍵情報を添付しているので，署名済み XML 文書を受け取った側で鍵の入手方法を考慮する必要がない

また，正規化については，前述の James Clark の expat を正規化オプションつきで利用している。

4. 処理過程

本システムでは，以下のような処理を行って署名をしている（図１）。

署名をしたい XML 文書を与えると，文字コードを UTF-8 に変換し，expat により正規化を行う。正規化された XML 文書の本文から SHA1 を算出し，バイト列を Base64 で変換して，XML Signature で規定されている SignedInfo ノードに

埋め込む。

次に，このSignedInfoノード全体を正規化して，SignedInfoのSHA1を算出し，PKCS#12から読み込んだRSA秘密鍵によって暗号化して，バイト列をBase64で変換してSignatureValueとして書き出す。

また，PKCS#12から読み出した証明書情報をサブジェクトやシリアル番号，証明書としてKeyInfoに書き出している。

現在のところ，以下の仮定をした実装をしている。汎用的な拡張は今後の課題である。

- 署名処理系への入力はUTF-8，正規化済み
- 鍵情報入力はPKCS#12
- 署名方式はRSAwithSHA1

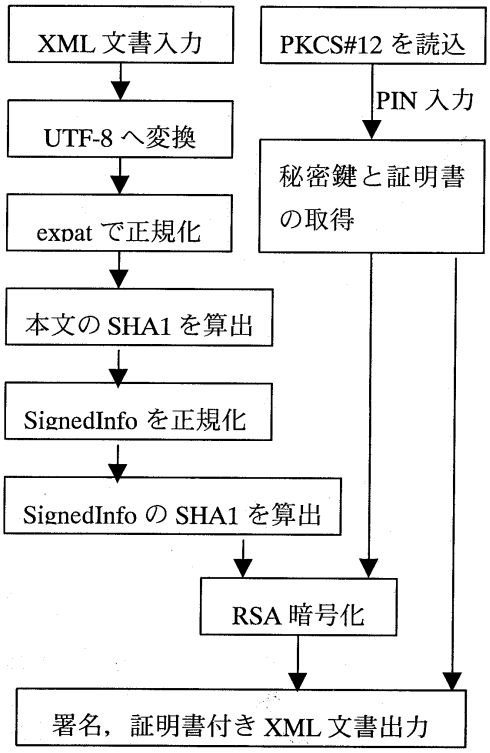

図1．処理の流れ

5 実装の諸問題

ワーキングドラフトに沿って実装を試みたが，以下の点が問題となった。

- KeyValueの扱い
  RSAKeyValueの規定で実際の鍵パラメータをBase64で記述することになっているが，KeyInfoの中でKeyValueしか定義されていない。

- X509CertificateとKeyValueの冗長性
  X509Certificateがあれば，鍵パラメータは全て取得できるはずだが，KeyValueの必要性が不明。

今回の実装では，X509Certificateから鍵情報が取得できるので，KeyValueは付加しないようにしている。

6 おわりに

比較的入手しやすいRSA鍵を利用した電子証明書を基盤にして，XML署名の実装を行った。今後は以下のような課題に取り組みたいと考えている。

- 一般ユーザがXML署名できる簡便な環境を用意して，署名の運用に関する検討

- XML署名ツール自体の規格準拠性を高める

参考文献

[1] http://www.w3.org/Signature/
[2] http://www.w3.org/Signature/Code/Canonicalizer.java
[3] ftp://ftp.jclark.com/pub/test/expat.zip
[4] ftp://ftp.pothole.com/pub/xmldsig/c14n.tar.gz
[5] http://www.alphaworks.ibm.com/tech/xmlsecuritysuite
[6] 若山 公威，奥野 琢人，岩田 彰，村瀬 晋二，鈴木 春洋："暗号ライブラリと認証局パッケージの開発"，第59回情報処理学会全国大会，1999.