

OASIS Web Services Security の実装と評価

砂田 英之 北山 泰英 茂木 強

三菱電機株式会社 情報技術総合研究所

1. はじめに

OASIS Web Services Security⁽¹⁾ (以下、WS-Security と略す) は、SOAP⁽²⁾⁽³⁾ (Simple Object Access Protocol) メッセージ仕様を拡張して、セキュリティトークンによるユーザ認証、SOAP メッセージへの署名付与および暗号化のサポートにより、SOAP によるセキュアなメッセージ交換を実現するものである。

我々は、2003 年 8 月 27 日公開の、WS-Security 仕様 (Web Services Security: SOAP Message Security Working Draft17) に基づき、SOAP メッセージへの署名付与機能の試作開発を行った。

本稿では、試作開発を通して、仕様そのものと実装方式の評価を述べる。

2. OASIS Web Services Security 概要

(1) 特長

WS-Security では、SOAP メッセージを拡張し、任意のセキュリティ情報を、SOAP メッセージヘッダに格納することにより、SOAP メッセージの機密性・秘匿性を保証する (図 1)。

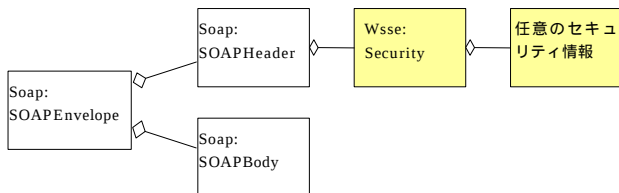


図 1 SOAP メッセージの拡張

(2) 機能

WS-Security にて規定されている機能は、以下の 3 種類である。

- ・セキュリティトークン：ユーザ認証
受信側が認証や認可を行えるように、認証・認可情報を SOAP メッセージヘッダにセキュリティトークンとして格納する。
- ・SOAP 署名：SOAP メッセージへの署名付与
メッセージの完全性を保証できるように、SOAP メッセージに対する署名情報を SOAP メッセージヘッダに格納する。署名方法については、W3C の XML 署名仕様⁽⁴⁾を用いる。
- ・SOAP 暗号：SOAP メッセージの暗号化
メッセージの秘匿性を保証できるように、SOAP メッセージに対する暗号情報を SOAP メッセージ内に格納する。暗号方法については、W3C の XML 暗号仕様⁽⁵⁾を用いる。

(3) セキュリティモデル

WS-Security のセキュリティモデルでは、以下の 3 者のプレーヤが想定されている。

- ・サービスリクエスタ：Web サービスの要求側
- ・サービスプロバイダ：Web サービスの提供側
- ・セキュリティトークンサービス：
認証書やチケットを提供する第三者機関。

これらのプレーヤが、図 2 に示す処理フローにて、セキュアなメッセージ交換を実現する。

リクエスタはセキュリティトークンサービスに対して認証要求を行い、トークンを取得し、サービスのリクエストメッセージにセキュリティトークンを付与して送信する。

プロバイダは、リクエストメッセージを受信し、付与されたセキュリティトークンを取得し、セキュリティトークンサービスに対して照合要求を行う。セキュリティトークンが信頼されると、リクエストを処理し、サービスのレスポンスを返す。

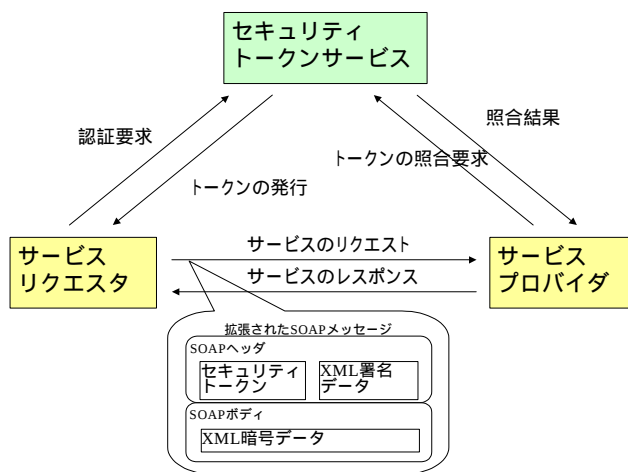


図 2 WS-Security セキュリティモデル

3. SOAP 署名機能の試作開発概要

今回の開発では、WS-Security の SOAP 署名機能について試作開発を行った。以下に、試作の構成と実装した機能を説明する。

(1) 構成

本試作では、上位の Web サービス / SOAP クライアントが意識せずに SOAP メッセージの署名作成 / 検証が行えるように、Handler インタフェースを用いた実装方式を採用した (図 3)。

これにより、SOAP メッセージの送信 / 受信時に Web サービス / SOAP クライアントの要求 / 応答フローをフックして処理することができる。

機能の中心となる、XML 署名に関しては、当社製の XML 署名ライブラリを用いた。

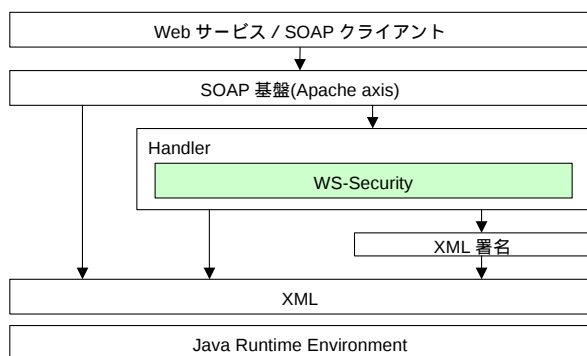


図 3 WS-Security 試作構成

(2)機能

本試作では以下の機能を実装した。

- ・ SOAP メッセージボディへの署名作成 / 検証
SOAP メッセージボディを署名対象とする署名作成・検証を実装した。署名に用いる鍵情報は、試作 (Handler) のデプロイ時のパラメータとして設定可能である。
- ・ 対象メッセージの選択
署名作成 / 検証の対象となる SOAP メッセージの種類 (要求 / 応答) が設定可能である。
- ・ エラー通知
送信側へのエラー通知として、SOAP Fault を用いたエラー通知を実装した。

4 . OASIS Web Services Security 仕様の評価

試作開発にて明らかになった WS-Security 仕様そのものに関する評価を述べる。

(1)エンドツーエンドの暗号化を実現

WS-Security では、Web サービスの中継者を考慮し、メッセージの完全性・秘匿性を保証するための SOAP メッセージの記述方法を規定している。これにより、ポイントツーポイントではなく、エンドツーエンドの暗号化を実現できる。

(2)セキュリティトークンの利用

セキュリティトークンとメッセージを関連づける方法を規定しているため、XML 署名・XML 暗号とトークンを組み合わせて利用することができる。これにより、効率的なメッセージ組み立てが可能である。

(3)相互接続での注意

WS-Security にて規定されるメッセージを構成する要素としては、ANY (任意) として定義されており、実装に依存する部分が多く、処理方式に関しても特に規定がない。

例えば、セキュリティトークンの情報では、任意のトークンを添付することが可能であるため、特定のトークンを利用しない場合には、完全に実装依存となる。汎用性のある一方で、相互接続を行う際には、注意が必要である。

現在、相互接続の確認用として、OASIS Web

Services Security TC にて、3 パターンのインターオペラビリティのシナリオが定義されているだけで、今後、相互接続性を確保するためには、より一層の確認パターンの充実が必要である。

(4)鍵交換 / 認証・認可方法

鍵交換や認証・認可の方法に関しては対象外となっているため、Web サービスシステムを設計する上では、個別に考慮する必要がある。

5 . OASIS Web Services Security の実装評価

今回実装した試作開発そのものの評価を述べる。

(1)Handler インタフェースによる実装

本試作のように、Handler インタフェースを用いて実装することは、Web サービス / SOAP クライアントを変更することなく署名機能を用いることができるため、既存の Web サービスに固定的な処理を実現する場合に最適である。一方、処理に応じて任意の要素に対して署名・暗号を行う場合や、用いる鍵情報を変更する場合には、Handler インタフェースによる実装では設定情報が複雑になるため、Web サービス / SOAP クライアントから直接呼び出せる API にて、詳細な設定を可能とするライブラリにて実装する必要がある。

(2)柔軟性・拡張性の実現

WS-Security 仕様は実装依存の部分が多い。試作開発を通じ、Web システムのセキュリティポリシーに応じた処理の変更や、任意の要素にも対応できるように、柔軟性・拡張性を持った設計をすることが必要だと認識した。

具体的には、セキュリティトークンなど、システムのポリシーに応じて処理を変更する場合や、任意の要素を用いる場合などを考慮して、要素に対応する処理を行う実装クラスをプロバイダ方式などのプラグブルな方式にて実装するなど柔軟性・拡張性を考慮した設計が必要であった。

5 おわりに

本稿では、OASIS Web Services Security の概要紹介および弊社で行った SOAP 署名機能の試作開発について記述した。また、試作の開発を通じて、仕様および実装方式に関する評価を報告した。今後、Web サービスの普及とともに、Web Services Security 技術の幅広いシステムへの適用に向けた開発を行っていく所存である。

参考文献

- (1) OASIS “Web Services Security SOAP Message Security”
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
- (2) W3C “SOAP Version1.2 Part0: Primer”
<http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>
- (3) W3C “SOAP Version1.2 Part1: Messaging Framework”
<http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>
- (4) XML Signature Syntax and Processing
<http://www.w3.org/TR/xmlsig-core/>
- (5) XML Encryption Syntax and Processing
<http://www.w3.org/TR/xmlenc-core/>