W3C XKMS による鍵登録～証明書発行機能の開発（その２）

武田 哲　　坂上 勉　　北山 泰英　　砂田 英之
三菱電機株式会社 情報技術総合研究所

1.はじめに

標準化団体 W3C では、XML(Extensible Markup Language)における公開鍵情報の管理方法について仕様策定しており、2003 年 4 月に XKMS (XML Key Management Specification) Version 2.0 のワーキングドラフトを公開した[1]。XML は Web サービスなどの普及とともに今後より一層の利用拡大が見込まれ、その場合 XML 上で公開鍵暗号方式による認証や署名が行われ、X.509 証明書が利用される機会も増えると考えられる。

我々は、これまで X.509 証明書を発行する証明書発行サーバや XML 署名／暗号 Java ライブラリといったソフトウェアをそれぞれ開発してきた。今回これらサーバとライブラリを組み合わせ、XKMS の鍵登録機能を実現する証明書発行サーバを開発した。本稿では、証明書発行サーバに XKMS を適用するための検討内容や実装方法について述べる。

2.XKMS 適用検討

XKMS はメッセージの種類として大きく 2 つに分けられ、登録関連の X-KRSS と問合せ関連の X-KISS からなり[2]、その中で証明書発行サーバに関係するのは X-KRSS である。さらに X-KRSS には、公開鍵の登録、再発行、失効、回復の 4 種類のメッセージがある。これら 4 種類の中で、証明書を利用するためにはじめに必要となるのは公開鍵の登録であるため、証明書発行サーバには、まず公開鍵の登録機能を実装することとした。

また XKMS にはメッセージの通信方法として 3 種類あり、1 度の通信で処理が終了する同期型、通信を 2 回連続させる 2 段階型、複数回の通信が連続しない非同期型がある。2 段階型では、2 回の通信は同じ種類のメッセージで行われ、メッセージの内容が異なる。非同期型では、通信の度にメッセージの種類が異なる。今回は実装の実現を最優先するため、処理するメッセージの種類や内容を抑え、同期型のメッセージ手順を実装することとした。これらの検討結果を踏まえ、今回の実装範囲は以下の通りとした。

- メッセージの種類は、鍵登録 (RegisterRequest, RegisterResult)
- メッセージの手順は、同期型
- 1 メッセージで登録可能な公開鍵の数は、1 つ

3.XKMS 範囲外の仕様検討

さらに XKMS で使用されるものの仕様の範囲外となっている、アクセス承認コードのリクエスト側とレスポンス側との間の共有方法について検討を行った。アクセス承認コードは、鍵登録の要求メッセージである RegisterRequest 送信前に、予めリクエスト側とレスポンス側双方で共有する必要がある。リクエスト側は RegisterRequest 作成時に KeyBindingAuthentication 要素に入れる HMAC(Hashed Messsage Authentication Code)の作成に、またレスポンス側は HMAC の検証に、それぞれ自ら持つアクセス承認コードを使用し、リクエスト側を事前に承認済であることを確認する。

XKMS では通信プロトコルとして SOAP over HTTP が推奨されており、レスポンス側の証明書発行サーバは Java サーブレットとして実装し、リクエスト側は Web ブラウザを考えた。そこでアクセス承認コードの共有方法として、**図 1** の通り鍵登録処理の前に、まずユーザ情報の登録処理を行い、その応答として、アクセス承認コードとそのリンク情報であるリクエスト ID を、サーブレットからブラウザへ応答するようにした。その後、鍵登録時にブラウザからリクエスト ID を送信するようにし、サーブレットはリクエスト ID を条件に DB か

"Implementation of W3C XML Key Management Specification"
Satoshi TAKEDA, Tsutomu SAKAGAMI,
Yasuhide KITAYAMA, Hideyuki Sunada
Information Technology R&D Center,
Mitsubishi Electric Corporation

らアクセス承認コードを取得する手順を考えた。このとき、リクエスト ID の通信上の受け渡し方法として、HTML に含める方法、URL に含める方法、cookie に含める方法が考えられた。XKMS では鍵登録以外にも、再発行、失効、鍵回復用にそれぞれのアクセス承認コードが必要であり、それらへの関連付けが必要となる。このためリクエスト側に関連情報であるリクエスト ID を保持させるため、保存可能な cookie に格納する方法とした。cookie については、中身を見られるなど安全面で心配されるが、リクエスト ID 自体は秘密情報ではなく、cookie に格納して問題ないと考えた。

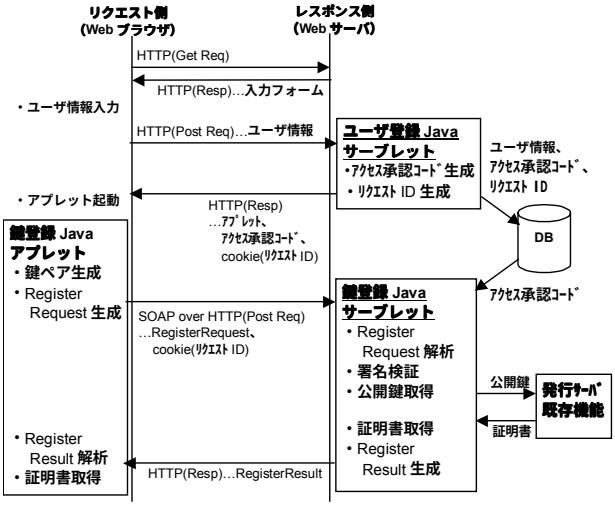

図 1　XKMS 鍵登録機能の構成と処理の流れ

4.証明書発行サーバの連携検討

証明書発行サーバは、これまで鍵登録機能として PKCS #10 形式に対応しており、可能な限り処理を共通化できないか、具体的には鍵情報と署名情報について検討した。PKCS#10 では鍵情報は ASN.1 データ形式の SubjectPublicKeyInfo であり、署名情報は SubjectPublicKeyInfo を含む CertificationRequestInfo の署名である。一方 RegisterRequest では鍵情報は KeyInfo 要素であり、署名情報は KeyInfo 要素を含む PrototypeKeyBinding 要素の署名が ProofOfPossession 要素に入る。このため、鍵情報は KeyInfo 要素から SubjectPublicKeyInfo に変換し、署名の検証を行った後、処理を共通化可能であることが分かった。

5.鍵登録機能の実装

検討結果を元に、同期型の鍵登録機能を実現する、以下のサーブレットを実装した（**図 1**）。

・ **ユーザ登録 Java サーブレット**

鍵登録を行うユーザの氏名などの情報を登録し、アクセス承認コードとリクエスト ID、さらに鍵登録～証明書取得を行う鍵登録 Java アプレットを Web ブラウザに応答する。

・ **鍵登録 Java サーブレット**

鍵登録 Java アプレットから RegisterRequest の SOAP メッセージを受け取り、証明書発行サーバの既存処理と連携して証明書を取得し、RegisterResult を作成して SOAP メッセージを応答する。RegisterRequest の解析や署名検証、RegisterResult の組み立ては、既存の XML 署名／暗号ライブラリを機能拡張し、XKMS 要素を処理する XKMS メッセージ組み立て／解析 Java ライブラリを使用する。

鍵登録 Java サーブレットは、**図 2** の通り SOAP 処理、XKMS 処理、鍵登録処理、証明書発行サーバ連携と分かれており、今後 X-KRSS の他のメッセージである再発行、失効、鍵回復や、他のメッセージ手順である 2 段階型、非同期型に対応可能な構成としている。

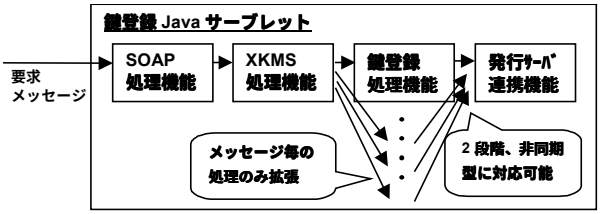

図 2　鍵登録 Java サーブレットの構成

6.おわりに

W3C XKMS に定義された鍵登録処理を実装し、鍵登録～証明書発行機能を実現した。適用例として、証明書の即時発行を行うオンライン認証局が考えられる。今後の課題として、まず性能評価がある。時間当たりの証明書発行数について調査したい。さらに他の X-KRSS、X-KISS メッセージや手順に対応し、様々な用途に利用できるようにしたい。

参考文献

[1]W3C, "XML Key Management Specification (XKMS) Version 2.0 W3C Working Draft 18 April 2003", http://www.w3.org/TR/xkms2/
[2]北山他, "W3C XKMS による鍵登録～証明書発行機能の開発（その１）", 情報処理学会第 66 回全国大会 5J-5, 2004