2B-5

W3C　XKMS による証明書更新および失効機能の開発（その１）

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

１．はじめに

XKMS(XML Key Management Specification)は、PKI(Public Key Infrastructure)を組み込んだ業務システム開発の容易化やコスト削減等を XML や Web サービスを活用して実現することを目的としたものである。現在 W3C でそのメッセージ仕様を標準化策定中で、最新のドキュメントは 2004 年 4 月 5 日公開の W3C　XKMS 勧告候補である[1]。

今回我々は、本ドキュメントに基づき証明書の更新、失効を可能とする Java クラスライブラリの実装開発を行ったので報告する。

２．W3C　XKMS メッセージ概観

XKMS のメッセージは、X-KRSS(鍵の登録/回復、証明書の更新/失効に関する機能：Register, Recover,Reissue,Revoke)及び X-KISS(証明書の検証/取り出しに関する機能：Locate,Validate)の 2 種類に大別できる(図 1)。

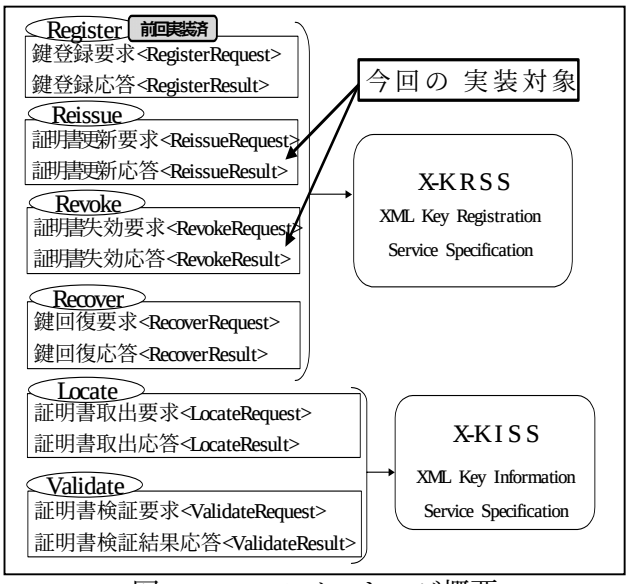

図１　XKMS メッセージ概要

以降、今回のターゲットである証明書更新、失効機能の仕様について規定している X-KRSS の Reissue、Revoke にフォーカスして説明する。

３．XKMS 証明書更新、失効系機能について

ReissueRequest メッセージは、リクエスト側で保持される証明書情報、POP 情報(Proof of Possession：処理要求者が対応する証明書を所有することの証明情報)、事前に交換したアクセス承認コードにて本人確認を可能とする AuthenticationCode などの情報から構成されている。構成は RegisterRequest とほぼ同等であるが、RegisterRequest では鍵情報の格納先として KeyBindingAbstractType から派生した PrototypeKeyBinding タグを用い、ReissueRequest では KeyBinding から派生した ReissueKeyBinding を用いる点で差がある。

また、RevokeRequest メッセージの構成は ReissueRequest とほぼ同等であるが、POP 情報の他に AuthenticationCode 要素あるいは RevocationCode 要素のどちらかを子要素として持つ。どちらも、処理要求者に対して身元確認を行うための情報であるが、RevocationCode はユーザがパスフレーズを定めて用いる仕組みであり、以下のように使用する。まず、鍵登録時にパスフレーズを定め、RegisterRequest の子要素 RevocationCodeIdentifier の値としてパスフレーズに HMAC-SHA1 処理を２回施したものを入れ、本人確認登録情報として登録する。証明書失効時には、RevocationCode の値として、鍵登録時に設定したパスフレーズに HMAC-SHA1 処理を１回施したものを与えて送信する。受け取ったサーバ側では、メッセージから取り出した RevocationCode の値に HMAC-SHA1 処理をもう一回施し、鍵登録時に登録された RevocationCodeIdentifier 情報と比較し一致することを確認する。つまり、鍵登録時に２回の HMAC-SHA1 処理を施した RevocationCodeIdentifier 情報が漏洩したとしても、証明書失効時に用いる１回の HMAC-SHA1 処理を施した RevocationCode 値を推測することは出来ないため、セキュリティ性が保たれる仕組みとなっている。

一方、サーバ応答値である ReissueResult、RevokeResult メッセージは、応答ステータス(処理の成功、不成功)等から構成され、Reissue では更新済 X.509 証明書も添付される。

"Implementation of W3C XML Key Magement Specification, Part 1"
Reiko ABE,Yasuhide KITAYAMA, Hideyuki SUNADA, and Satoshi TAKEDA,
Information Technology R&D Center, Mitsubishi Electric Corporation

４．開発概要

今回は、前回作成した Register 機能に加え、Reissue、Revoke 機能を実現する XKMS メッセージ作成クラスライブラリ、クライアント機能を実現するクライアントクラスライブラリ、サーバ機能を実現するＸＫＭＳサーバを作成した。以下にクライアントクラスライブラリを構成する主要パッケージにフォーカスして」説明する。

① xkms.client パッケージ
XKMS クライアント機能を実現するための、証明書および秘密鍵管理機能、プロバイダ管理機能、メッセージの送受信機能など共通的な項目からなるクラスから構成される。
② xkms.client.register パッケージ
XKMS 鍵登録機能を実現するための、操作画面、register メッセージの作成機能を提供するクラスから構成される。
③ xkms.client.reissue パッケージ
XKMS 証明書更新機能を実現するための、操作画面、reissue メッセージの作成機能を提供するクラスから構成される。
④ xkms.client.revoke パッケージ
XKMS 証明書失効機能を実現するための、操作画面の提供と revoke メッセージの作成機能を提供するクラスから構成される。

①は、Register、Reissue、Revoke の各クライアント共通で使用するクラスで構成されている。機能としては、証明書及び秘密鍵管理機能（証明書保存、X509 取得、秘密鍵取得機能）、クライアントサーバ間のメッセージ送受信機能、プロバイダ管理機能など XKMS クライアント全てに共通する機能であり、未実装である Validate、Locate 機能にも共通する機能となる。

②～④は、各機能に対応した操作画面の提供と各リクエストメッセージ作成機能で構成されている。証明書に関しては JKS および PKCS#11、PKCS#12 をサポートした。PKCS#12 機能は JDK1.5 でも未サポートであるため自製のプロバイダで対応し、PKCS#11 に関してもラッパーを自製しＩＣカードの使用を可能とした。このように複数の証明書タイプをサポートしたため、使用する証明書タイプ選択手段提供のため、二種類の設定を用意した。１つめはサーバ提供の HTML ファイルにパラメータを埋め込む事でユーザ指定不可とする設定、２つめはユーザが証明書タイプ選択可能とする設定である。これにより、証明書タイプを固定して操作の誤りを無くすか、ユーザ選択可能とし柔軟性を持たせるかを選択可能とする作りとした。

５．XKMS の適用効果

本 Java クラスライブラリを使用して構築した XKMS サーバ及び XKMS クライアントを用いる事で、証明書取得（鍵登録）、証明書更新、証明書失効の一連の処理をオンライン上で行う事が可能となる。従来の処理との比較した場合、ＩＣカードなどの媒体送付が不要となる（図 2）。

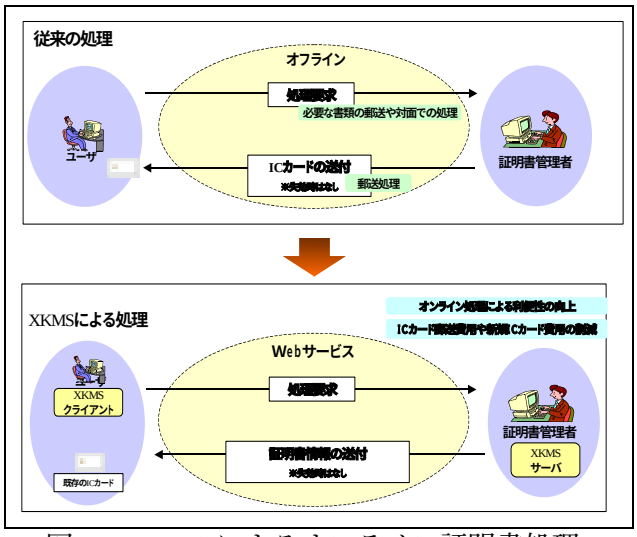

図 2　XKMS によるオンライン証明書処理

６．まとめ（評価）

本開発では W3C　XKMS 仕様で規定される要素のスキーマ定義に対応した処理クラスの実装およびクライアント機能、サーバ機能の開発を行った。電子証明書の新規発行では本人確認を対面で行う場合が多く完全なオンライン処理となっていないのが現状である。しかし、本開発により、更新や失効機能は証明書の配布や本人確認が新規発行時点で確立できていることから本人確認がオンライン上で実現可能であるため、特に証明書の更新や失効において XKMS の適用により利便性が高まるという事がわかった。

また、クライアントのパッケージ化については、共通機能を洗い出し、鍵登録、証明書更新および失効時共通に使用できるクラスとしてまとめてパッケージ化した。その結果、コード量は共通機能約 0.6KL、非共通機能約 2.8KL となった。XKMS クライアントの Java アプレット[2]のサイズは XML 署名機能も含めて 400KB 程度に収まり、共通化可能部分を洗い出すことで、効率的な開発が可能となる事がわかった。

参考文献

[1]W3C “XML Key Management Specification (XKMS2.0) Version 2.0 W3C Candidate Recommendation 5 April 2004” http://www.w3.org/TR/xkms2/
[2]武田他、”W3C XKMS による証明書更新および失効機能の開発(その 2)”、情報処理学会第 67 回全国大会 2B-6,2005