2B-6

W3C XKMS による証明書更新および失効機能の開発（その２）

武田 哲　　阿部 玲子　　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)ベースの公開鍵情報の管理方法についての仕様 XKMS (XML Key Management Specification)を策定しており、2004 年 4 月に Version 2.0 の勧告候補を公開した[1]。XML を活用したアプリケーション上で今後より一層電子認証や電子署名が行われると予想され、X.509 証明書の管理のため XKMS の普及が期待される。

我々は、昨年 XKMS の鍵登録機能を実装した証明書発行サーバを開発し、XML メッセージによる証明書の新規発行を実現した[2][3]。今回さらに XKMS の再発行および失効の追加開発を実施し、証明書の更新発行および失効を実現した。本稿では、証明書発行サーバに XKMS の再発行および失効を適用するための検討内容や実装結果について述べる。

2.鍵登録機能の実装結果

これまで開発した鍵登録機能における実装結果は以下であった。

- メッセージの手順は、同期型
- 鍵ペア生成は、クライアント
- 1 メッセージで登録可能な公開鍵の数は、1 つ（1 メッセージにつき 1 証明書を発行）
- アクセス承認コードの共有方法は、XKMS 鍵登録前に行うユーザ情報登録時。ユーザ情報の登録結果とともにサーバから応答
- アクセス承認コードとともに、関連情報となるリクエスト ID を、cookie 情報として応答

はじめの 3 点については、実装の実現を最優先するためにメッセージの種類や内容を抑えた結果であった。

最後の 2 点については、XKMS の仕様範囲外となるアクセス承認コードに関する実装結果である。オンラインでやりとりし、関連情報のリクエスト ID をクライアントに意識させること無く、また鍵登録後に行われる再発行や失効時までリクエスト ID を保存することを考慮した結果であった。

3.適用検討

再発行および失効の XKMS メッセージについて、XML スキーマから使用する要素を検討した。

3.1.再発行機能

鍵登録機能について再検討した結果、リクエスト ID の処理を見直した。これは cookie を使用する場合クライアントがブラウザに制限されること、cookie が削除された場合に再発行や失効時 cookie の回復処理が必要になり、却って処理が複雑になることなど考えられたためである。W3C の XKMS メイリングリストなどから、要求メッセージ中の KeyName 要素が ID 情報の格納場所として使用されていることを知り、互換性も考え KeyName 要素をリクエスト ID の格納場所にすることとした。このため、鍵登録前に行われるユーザ登録時のリクエスト ID の応答方法も、登録結果やアクセス承認コードとともに応答するよう、既存処理を変更することとした。

さらに XML スキーマをもとに検討した結果、鍵登録メッセージにあった PrivateKey 要素が、再発行メッセージには無いことが分かった。このことから、鍵登録にサーバで鍵ペアを生成する場合、再発行時の鍵の変更はできない。このため XKMS における再発行機能は、すでに証明書発行された鍵に対して証明書を発行し直すものと考え、以下を実装範囲とした。

- 再発行時の元情報は発行済の証明書
- 元情報の証明書の公開鍵に対し、要求時からの有効期間となる証明書を発行（同一の鍵に対する証明書の更新）
- クライアントの認証方法は、アクセス承認コードを使用した検証と、証明書と対になる秘密鍵による署名の検証

"Implementation of W3C XML Key Management Specification, Part2"
Satoshi TAKEDA, Reiko ABE,
Yasuhide KITAYAMA, Hideyuki SUNADA
Information Technology R&D Center,
Mitsubishi Electric Corporation

3.2.失効機能

XML スキーマをもとに検討した結果、クライアントの認証方法として、アクセス承認コードを使用した検証と、鍵登録時クライアントが指定可能な失効パスフレーズを使用した検証から選択可能なことが分かった。失効パスフレーズを使用する場合、クライアントが失効パスフレーズを指定できるため、サーバから受け取るアクセス承認コードより忘れ難いと考えられ、利点と考えられる。しかし、クライアント側でアクセス承認コードの管理が必要なことに変わりは無く、失効パスフレーズと双方管理する方が却って負担になると予想される。このため以下を実装範囲とした。

- 失効時の元情報は発行済の証明書
- 元情報の証明書を失効
- クライアントの認証方法は、アクセス承認コードを使用した検証

合わせて、鍵登録前に行われるユーザ登録処理を変更し、クライアントからアクセス承認コードを指定可能にした。

4.実装結果

XKMS の各要求メッセージを解析した後のサーバ処理を検討し、実装した。

4.1.再発行機能

要求メッセージ中の証明書を使用することから、証明書に対して以下の確認処理を行った。

- 証明書が自ら発行したものかどうか確認
- 証明書の失効状態の確認。すでに失効済の場合はその時点でエラー応答

上記確認後、証明書から公開鍵を取得した後は、既存の鍵登録処理と同様とすることができた。この結果、再発行処理の流れは**図 1** となった。

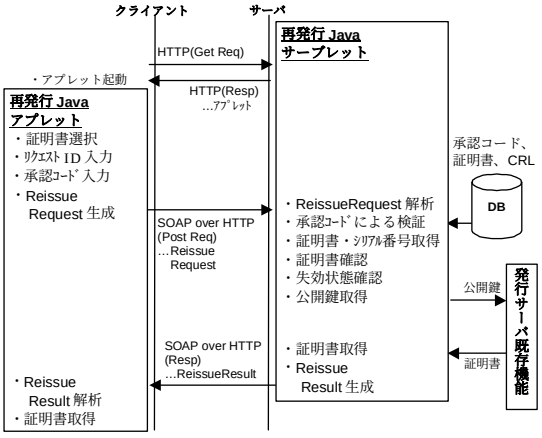

図 1　再発行機能処理の流れ

4.2.失効機能

失効時も要求メッセージ中の証明書を使用するため、要求メッセージに対する処理は以下を除き、再発行同様にできた。

- 証明書の失効状態の確認後、すでに失効済の場合はその時点で正常応答
- 証明書発行サーバの既存機能とのインタフェースは証明書のシリアル番号

この結果、失効処理の流れは**図 2** となった。

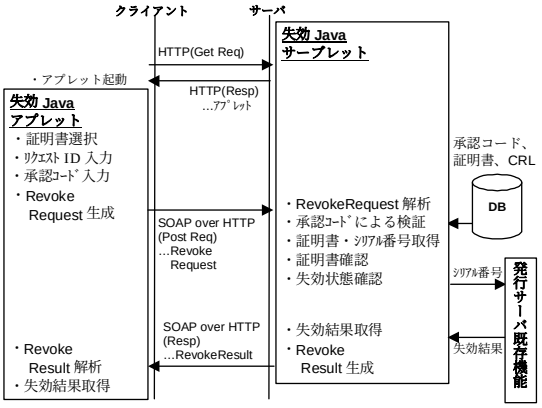

図 2　失効機能処理の流れ

5.むすび

XKMS では各メッセージの形式が似ており、鍵登録→再発行→失効と実装順に、多くの処理を流用できることが分かった。

利用形態については、メッセージを使用したクライアント認証が行われることから、証明書の持ち主自身がクライアントとなるシステムへの適用が考えられる。この場合、再発行については、持ち主が証明書の発行媒体を再利用可能になるため、サーバを運用する認証局の媒体管理の手間や負担が減り利点と考えられる。失効については、証明書の持ち主の利便性向上は考えられるが、持ち主自身による失効を許可するかどうか認証局の運用方針も考慮する必要がある。

今後検証メッセージにも対応し、さらに用途を拡げたい。

参考文献

[1]W3C, "XML Key Management Specification (XKMS) Version 2.0 W3C Candidate Recommendation 5 April 2004", 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
[3]武田他, "W3C XKMS による鍵登録～証明書発行機能の開発（その２）", 情報処理学会第 66 回全国大会 5J-6, 2004