5N－3

# S/MIMEメールにおける証明書廃棄状態検証サーバの開発と評価

若山公威*　福岡雄治*　岩田彰*　村瀬晋二**　鈴木春洋**
*名古屋工業大学　電気情報工学科
**（株）コンピュータ・テクノロジー・インテグレイタ　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上での電子メールやWWWなどのやり取りを安全に行うため、公開鍵暗号方式によるメッセージの暗号化や電子署名が利用されている。

暗号化や電子署名検証のためには、相手の公開鍵証明書（以下、証明書と呼ぶ）を入手し利用する必要があるが、証明書は廃棄されることがあるため最新のCRL(Certificate Revocation List、証明書廃棄リスト)を取得して有効性を確認する必要がある。しかしユーザ側でこの確認が行われない場合は、セキュリティ上の問題が生じる。

PEPOPのような暗号化サーバを用いることによっても、ユーザ側で証明書の有効性を気にする必要はなくなるものの、暗号化サーバまでの経路の安全性が必要となる。

そこで、我々は証明書の有効性確認を検証サーバという代理サーバで行う方法を提案し、S/MIMEメールの検証サーバを開発した。

## 2. CRL取り扱いの問題点と解決法

公開鍵暗号方式において、秘密鍵は本人が保持しているものであるが、公開鍵は暗号通信を行う相手に配布する必要がある。この配布がセキュアでない通信路で行われる場合、通信途中に改竄や成り済ましされる可能性がある。これを防ぐために、CA(Certificate Authority、認証局)と呼ばれる機関によって公開鍵に署名を付けて証明書として発行する。

証明書の発行後、秘密鍵が盗難されたり申請者の異動や退職などの身分の変更が起きた場合、CAは既に発行した証明書を無効にする必要がある。

もし受け取り人の秘密鍵が盗まれたことを知らずに暗号化メッセージを送ると、秘密鍵を盗んだ人物が経路を盗聴することによりメッセージを読むことが可能となる。また、秘密鍵を盗んだ人物が電子署名を送ることによって、秘密鍵の本来の所有者に成り済ますことが可能である。

無効にした証明書を一般ユーザに知らせるため、CAは無効となった証明書の一覧を定期的に発行する。これはCRLと呼ばれる。通常CRLはFTPサーバやディレクトリサーバなどで公開し、ユーザに定期的に取得させる形態を取る。ユーザ側で相手の証明書を用いる前に、CRLを取得し使用する証明書がリストに含まれていないかどうかを確認する必要がある。クライアントソフトによってはCRLの取得と確認を自動で行えないため、ユーザが証明書の発行CA名を確認し、手動でCRLを取得し、証明書を使うたびに1つ1つ有効性を確認する必要がある。

このようにCRLの取り扱いには不便な点があり、現状では証明書の有効性を厳密に確認されていない場合がある。我々はこの点を解消するため、サーバ側で暗号化メッセージと署名付きメッセージに使われている証明書の有効性を確認する方式を提案する。ユーザ側で証明書の有効性を確認する必要がなくなるため、ユーザの負担が少なくなり、かつ安全性を高めることが可能となる。

## 3. S/MIME

### 3.1 暗号化メール

メッセージの暗号化には共通鍵方式の鍵を用い、この共通鍵の暗号化には公開鍵暗号方式を用いる。暗号化されたメッセージと共通鍵は、PKCS#7のEnvelopedDataタイプに含められる。

```
EnvelopedData ::= SEQUENCE {
  version Version,
  recipientInfos RecipientInfos,
  encryptedContentInfo EncryptedContentInfo }
```

共通鍵の暗号化に使用した受信者の証明書情報(発行CA Distinguished Name、証明書シリアル番号など)は、recipientInfosに含まれている。

### 3.2 署名付きメール

署名付きメールには、次の2方式がある。

・ メール本文と署名をPKCS#7のSignedData

---

Implementation and evaluation of certificate validity check server for S/MIME
Kimitake WAKAYAMA*, Yuji FUKUOKA*, Akira IWATA*, Shinji MURASE**, Shun-yo SUZUKI**
*Nagoya Institute of Technology
**Computer Technology Integrator Co., Ltd.

内に入れる。

・ RFC1847 で規定されている multipart/signed MIME タイプにして、署名入りの PKCS#7 SignedData とメール本文を別々にする。このとき SignedData の contentInfo は空にする。

SignedData の構成は次の通りである。signerInfos に署名者の証明書情報が含まれている。

```
SignedData ::= SEQUENCE {
  version Version,
  digestAlgorithms DigestAlgorithmIdentifiers,
  contentInfo ContentInfo,
  certificates
    [0] IMPLICIT
        ExtendedCertificatesAndCertificates
        OPTIONAL,
  crls
    [1] IMPLICIT CertificateRevocationLists
        OPTIONAL,
  signerInfos SignerInfos }
```

## 4. システム概要

検証サーバを用いた場合の、メールの流れを図1に示す。

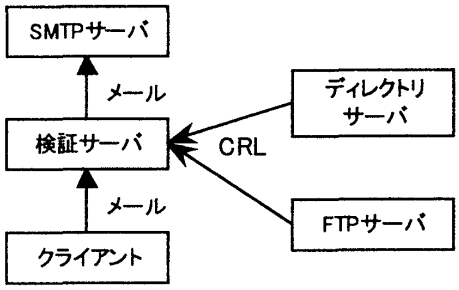


図1 メールの流れ

ユーザは通常の S/MIME クライアントを使用し、メールを SMTP サーバへ送る代わりに検証サーバへと送る。検証サーバではメールが S/MIME メールかどうかチェックし、S/MIME メールならば証明書の有効性を確認する。

検証サーバでは、S/MIME メール内の次の部分から証明書発行 CA の Distinguished Name と証明書シリアル番号を取り出す。

・ 暗号化メールの場合は recipientInfos

・ 署名付きメールの場合は signerInfos

そして、発行 CA に対応したリポジトリから CRL を取得し、証明書のシリアル番号が含まれているかどうか確認する。CRL に載っていないシリアル番号であれば、そのメールに使われた証明書の有効性が確証できるので通常通り SMTP サーバへとメールを送信する。もし CRL に載っていれば、メールを送信者に送り返す。一度取得した CRL は検証サーバで保持しておき、次回の CRL 発行時がきたらリポジトリから取得する。各発行 CA に対する CRL の取得先が証明書に書かれていない場合が多いため、今回は検証サーバであらかじめ設定しておくことにした。

署名付きのメッセージを暗号化したメールについては、復号しなければ署名に用いた証明書の情報が得られないため、暗号化に用いた証明書の有効性のみをチェックすることにした。

## 5. 処理速度

検証サーバでの暗号化メールと署名付きメール(multipart/signed)の処理時間を表1と表2に示す。メールクライアントには Microsoft Outlook Express を使用し、検証サーバは Sun Sparc Station 5 上に設置した。検証サーバ上での処理時間のみを計測し、CRL はあらかじめ検証サーバが保持しているものを使用した。

本サーバでは暗号文の復号は行わず、単に base64 あるいは uuencode のデコードと EnvelopedData、SignedData の解釈を行うのみなので、実用に耐えうる速度と考えられる。

表1 暗号化メール

| メールサイズ (k byte) | 時間(s) |
|---|---|
| 1.5 | 0.05 |
| 8.0 | 0.11 |
| 52 | 0.64 |
| 110 | 1.29 |

表2 署名付きメール

| メールサイズ (k byte) | 時間(s) |
|---|---|
| 2.2 | 0.04 |
| 8.0 | 0.05 |
| 66 | 0.06 |
| 110 | 0.05 |

## 6. おわりに

サーバ側で証明書の廃棄状態を検証する方式を提案し、S/MIME メールにおける検証サーバを開発した。本方式は S/MIME のみでなく、SSL などにも適用可能である。

今後は、証明書の廃棄状態のチェックのみでなく、過去に受け取った署名がその時点で有効であったことを明らかにする公証サーバへの発展も考えられる。

## 参考文献

[1] S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, L. Repka: "S/MIME Version 2 Message Specification", RFC2311, 1998.

[2] RSA Laboratories: "PKCS #7: Cryptographic Message Syntax Standard", 1993.