マルチメディア通信と分散処理　101－8
コンピュータセキュリティ　12－8
（２００１．２．２０）

ユーザによる証明書管理を削減するためのS/MIMEアプレット実装

若山公威†　　中山幹浩†　　村瀬晋二‡　　鈴木春洋‡　　岩田彰†

†名古屋工業大学　電気情報工学科
‡株式会社シーティーアイ SI事業部
E-mail: wakayama@elcom.nitech.ac.jp

あらまし

S/MIMEを利用する場合、自分の証明書と通信相手の証明書が必要となる。さらに、相手の証明書が廃棄されていないか確認するためにCRLが必要である。現在のS/MIMEクライアントソフトでは、この作業を自動化できないことが多いためユーザの負担となっている。本稿では、クライアントに証明書を保管するのではなくディレクトリサーバに保管しておき、必要なときにダウンロードして使用する機能を有したS/MIMEクライアントソフトをJavaアプレットにより実装した。この結果、クライアントには鍵ペアとCA証明書のみを保管しておけば良く、他の証明書を保管・管理する必要がなくなり、ユーザの負担を軽減することができた。

Implementation of S/MIME Applet to Reduce Certificate Management by User

Kimitake WAKAYAMA†, Mikihiro NAKAYAMA†, Shinji MURASE‡,
Shunyo SUZUKI‡, Akira IWATA†

†Nagoya Institute of Technology
‡CTI Co.,Ltd.

Abstract

It is necessary to obtain a user certificate and peer certificates when a user makes use of S/MIME. In addition, the user has to get CRLs to verify whether peer certificates are revoked. This work is a burden for the user because general current S/MIME client software can not automatically get and import them. Hence, we implemented S/MIME client software with Java applet which downloads certificates and CRLs from the directory server when they are necessary. As a result, the user does not need to manage peer certificates and CRLs, only has to manage his own key pairs and CA certificates.

1. はじめに

証明書のフォーマットとしてはITU-TのX.509 Ver.3 が一般的となっており、この証明書を発行できる CA(certification Authority)パッケージは多く存在する。フリーのものでは OpenSSL が有名であり世界中で広く使われている。また、我々は Windows 上で GUI 操作により利用できる CA パッケージを開発し公開している[4]。これら以外にも、有料のものでは多くの製品が発売されている。

証明書を用いるクライアントソフトとしては、Internet Explorer や Netscape Communicator のように無料で入手できる Web ブラウザやメールクライアントに、暗号メール S/MIME (Secure/Multipurpose Internet Mail Extensions)と SSL (Secure Sockets Layer)上の HTTP 通信機能が備わっている。

標準化に関しては、IETF の PKIX Working Group で証明書フォーマットや証明書管理プロトコルなどに関しての作業が進められており、製品の相互接続性が進むと思われる。

ただし、世界中で誰もが信じるルート CA が存在しないため、不特定人物同士の証明書のやりとりは難しい状況ではあるが、一企業やグループ企業のみなどの閉じた環境でプライベート CA を立ち上げて使用することは可能である。

しかし、実際にはそれほど証明書とそれを用いたアプリケーションが普及していない。この理由として、運用方法が複雑である点が挙げられる。我々はこれまで、証明書技術を普及させるために、証明書技術を用いる場合にユーザの負担を削減するための運用方法の検討とシステム開発を行ってきた[1][2][3]。本論文では、イントラネットやエクストラネットにおける S/MIME の運用にターゲットを絞り、現状の利用上の問題点を挙げて、その対策方法の提案と実装を行った。

2. S/MIME 利用上のユーザへの負担

ここでは、S/MIME を利用するためにユーザが行うべき作業について順に述べる。

まず暗号メール送信時は、通信相手の公開鍵を得るために証明書を取得する必要がある。さらに、この証明書が期限前に廃棄されていないかどうか確認するために CRL (Certificate Revocation List)の取得と検証が必要である。CRL は定期的に発行されるため、常に入手しなければいけない。メールクライアントによっては CRL の入手が自動化できないものがあるため、ユーザの負担となる。証明書も有効期限があるため、期限が切れたら新しい証明書を入手する必要がある。現在のメールクライアントでは、LDAP (Lightweight Directory Access Protocol)対応ディレクトリサーバから検索しクライアントへ保存することができるものもあるが、この証明書の有効期限が切れたら再度入手しなおす必要がある。

電子署名付きメールを送信する場合は、送信者本人の証明書情報が必要となる。そのため、本人証明書も常に有効なものを保持している必要がある。期限が過ぎたら CA に新しい証明書を発行してもらい、インポートする必要がある。

電子署名付きメールを受け取って検証を行うには、送信相手の証明書と CRL が必要となる[1]。過去に受け取った電子署名付きメールを検証するには、メール作成時点の相手証明書、CRL を入手することが必要となるが、入手が難しいことが多い。

証明書や CRL の入手と検証が全て自動で行えれば良いが、上記に述べたように、現状の S/MIME クライアントソフトでは自動化されていないところがあり、ユーザへの負担となっている。我々はこれまでに、証明書廃棄に関する問題を解決するために、サーバ側で証明書の廃棄状態を検証する方式の提案をした[1]。また、ユーザによる S/MIME 用クライアントソフトの設定をチェックするための機構も開発した[3]。しかし、依然としてユーザによる証明書管理作業やクライアントソフトの設定作業が残っている。

---

[1] 署名付きメール内に証明書が含まれている場合もある。

LDAP 対応ディレクトリサーバは証明書のリポジトリとして注目されており既に実際の運用環境で利用されている。ただし現状では、証明書を組織内で公開するための単なる保管場所として使われているのみである。証明書をディレクトリサーバから取得した後は、各ユーザの Web ブラウザや S/MIME クライアントなどで証明書を保管しており、証明書更新時にユーザの手を煩わすことになる。

本研究では、ディレクトリサーバでのみユーザ本人の証明書と通信相手の証明書を管理することを試みる。この結果、証明書の更新などの証明書管理作業からユーザが開放されることが期待できる。

3. システムの構成と実装

3.1 システム概要

提案システム構成を図 1に示す。ユーザの負担を減らすため、電子署名付き Java アプレットにおいてメール本文の暗号化と電子署名を行う。証明書と CRL は必要なときにディレクトリサーバからアプレットへダウンロードして使用する。秘密鍵はクライアントにて保持する。

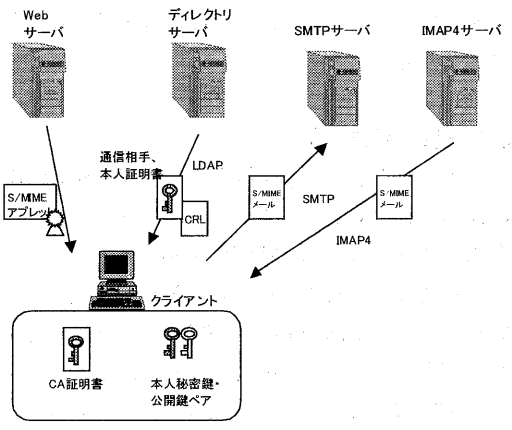

図 1 システム構成

電子署名付きアプレットからローカル資源への詳細なアクセス制御を設定するため、Web ブラウザには Java2 Plug-in が必要となる。アプレットから CA 証明書とユーザ秘密鍵・公開鍵ペアを読み込めるように、ポリシーファイルを設定しておく。アプレットの電子署名を検証するためにクライアントには CA サーバ証明書が必要となる。クライアントにはこの CA 証明書は必要であるが、これ以外の本人や通信相手のユーザ証明書は不要となる。

クライアントには、複数の鍵ペアと CA 証明書を保存可能とする。

3.2 ディレクトリサーバでの保管データ

ディレクトリサーバには、図 2 のようにイントラネットあるいはエクストラネット内メンバーのユーザ証明書と CRL を保管しておく。

```
dn: cn=Kimitake Wakayama, ou=Elcom, o=NIT, c=JP
objectClass: inetorgperson
sn: Wakayama
cn: Kimitake Wakayama
・・・
userCertificate;binary:: (ユーザ証明書1)
userCertificate;binary:: (ユーザ証明書2)

dn: cn=NIT CA, ou=Admin, o=NIT, c=JP
objectClass: certificationAuthority
・・・
certificateRevocationList;binary:: (CRL1)
certificateRevocationList;binary:: (CRL2)
```

図 2 ディレクトリサーバ内エントリ例

過去に受け取った電子署名付きメールを検証する場合は通信相手の過去の証明書と CRL が必要となるため、ディレクトリサーバに過去の証明書と CRL 全てを保管する。新旧証明書で有効期間が重なっている場合や、公開鍵が同じ場合と違う場合のどちらにも対応可能とする。

公開鍵と秘密鍵のペアを変更しないでユーザの本人証明書の有効期限のみを延長する場合は、ディレクトリサーバ管理者がユーザエントリーに更新した証明書を追加するだけで済み、ユーザの作業は全く必要ない。

3.3 暗号メール送信時の動作

1. ユーザがWebブラウザにより指定URLへア

クセスする。Webサーバからブラウザへアプレットがダウンロードされる。

2. メールの送信先アドレスやメール本文記入し、暗号/署名/署名+暗号/平文の指定をする。
3. 送信先のメールアドレスをもとに、ディレクトリサーバから通信相手の証明書を検索し、対応するCRLとともにクライアントへダウンロードし暗号処理に用いる。有効証明書が複数存在する場合は、有効期限切れ(notAfter)が最も先の証明書を用いる。

他のS/MIMEクライアントソフトと同様に、送信する暗号メールは自分も後々読めるように自分の公開鍵を用いて暗号しておく。今後の秘密鍵の更新後もこの暗号メールを読めるように、クライアントで新旧の秘密鍵を保持しておく。

3.4 電子署名付きメール送信時の動作

1, 2は暗号メール送信時と同様である。

3. 送信時は、ディレクトリサーバからメールアドレスをもとに本人証明書を検索してダウンロードする。これを本文に添付して、クライアントで保持している秘密鍵を用いて電子署名を施す。有効な証明書が複数存在する場合は、有効期限切れ(notAfter)が最も先のものを利用する。

証明書内の公開鍵に対応する秘密鍵が分かるように、クライアントには秘密鍵とその対応した公開鍵を保持しておく。

3.5 暗号メール受信時の動作

1. ユーザがWebブラウザにより指定URLへアクセスする。Webサーバからブラウザへアプレットがダウンロードされる。
2. アプレット上にメール一覧が表示される。読みたいメールをクリックする。
3. クライアントで保持している秘密鍵を用いて復号を行い、アプレット上にメール本文を表示する。複数の秘密鍵を保持している場合は、以下の手順で使用する秘密鍵を特定する。まず、PKCS#7 EnvelopedDataのrecipientInfos内のissuerAndSerialNumberにより暗号時に使われた証明書の発行CAとシリアル番号を得る。この証明書を受信者のメールアドレスをもとにディレクトリサーバから入手し、証明書から公開鍵を取り出す。ローカルで保持している公開鍵と秘密鍵ペアから対応する秘密鍵を特定する。

3.6 電子署名付きメール受信時の動作

1, 2は暗号メール受信時と同様である。

3. 電子署名の検証をする際、メールに署名者の証明書が含まれている場合はこれを用いるが、含まれていない場合はPKCS#7 SignedDataのsignerInfos内のissuerAndSerialNumberから証明書情報を得て、ディレクトリサーバからダウンロードして使用する。証明書が廃棄されているかどうかのチェックには、対応するCRLをディレクトリサーバからダウンロードして使用する。相手証明書のCA署名検証には、クライアントで保管しているCA証明書を用いる。

電子署名付きメールのDateヘッダの日時以降に証明書が廃棄されている場合は、署名された時点で鍵が有効であったか確定できないため、その旨をアプレット上に表示し判断はユーザに任せることにする。

3.7 実装と動作確認

今回は、現在多く利用されているS/MIME version2[5]を対象とした。メールスプールサーバとのプロトコルにはIMAP4を用いた。

開発には、JDK 1.2.2とJavaMail 1.1.3[6]を用いた。現在のJavaMailにはS/MIMEを扱うクラスは含まれていないため、RSAセキュリティ社のBSAFE Cert-J 1.0のPKCS#7クラスを用いてS/MIMEの処理を実装した。ユーザとCAの鍵ペアと証明書はEasyCert 0.86b1を用いて発行した。ディレクトリサーバはOpenLDAP 2.0.6を使用した。

現時点では CRL 検証や証明書の更新対応など一部機能以外のプロトタイプ実装が済んでおり、Internet Explorer 5.5 と Netscape Communicator 4.75 に付属の S/MIME クライアントそれぞれと相互にメール（暗号のみ、電子署名付きのみ、電子署名付き＋暗号の３通り）のやりとりが正常にできることを確認した。

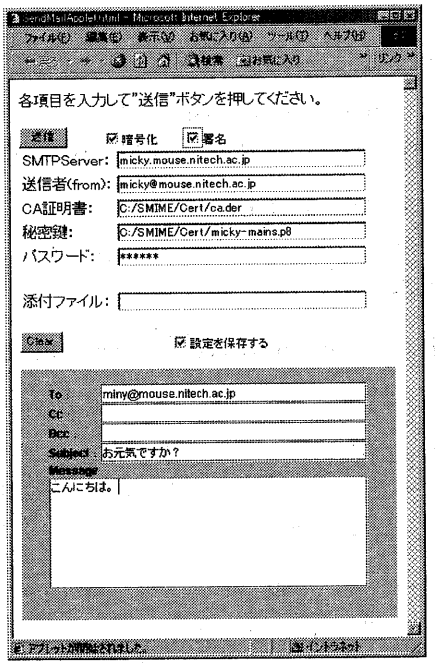

図 3　送信アプレット画面

3.8 今後の改良点

今回のアプレットでは平文メールと S/MIME メールのどちらでも送信できるようにしたが、経理担当者や人事担当者などの重要なデータを扱う人物が不注意により重要データを平文メールで送付しないように、あらかじめ S/MIME の暗号メールのみに固定することも可能である。

アプレットは電子署名されている必要があり、クライアントで電子署名の検証を行って、信用できる人物により署名されたかどうか確認するので、アプレットの安全性を保障するものではないが不正な動作をしないと仮定できる。しかし、秘密鍵はユーザ本人しか保持していないことを完全にするために、秘密鍵へのアクセスは避けたほうが良い。ローカル端末上で秘密鍵管理プログラムを起動させておき、アプレットから秘密鍵へのアクセスを避けるにする。依然、総当り攻撃はありうるが、回数制限を行ってロックすることで防ぐことができる。将来的には、Java カードなどの IC カードを用いる方式を検討したい。IC カードを利用しることにより、どの端末でも S/MIME の利用が可能となる利点もある。

4. 他方式との比較

4.1 Netscape Communicator のローミングアクセス機能

証明書と秘密鍵をディレクトリサーバ(または Web サーバ)上で集中管理しておき、パスワードによる認証で Netscape Communicator へダウンロードすることができる。パスワードベースのためセキュリティレベルが下がる。また、他のブラウザでは使用できない。

4.2 Entrust Technologies 製品[7]

キーペアと証明書の自動更新が可能である。ローミング機能もありセンターサーバにおいて秘密鍵も集中管理が可能である。しかし、これらは Entrust Technologies 専用クライアントのみしか使えない。我々の方式では標準的な LDAP 対応ディレクトリサーバと Web ブラウザを使用しており、既存のイントラネット環境への容易な導入が可能である。

4.3 PEPOP[8][9]

サーバにおいてユーザの秘密鍵と証明書を集中管理して、クライアントの認証はワンタイムパスワードを利用する。サーバとクライアント間の改竄や盗み見があり得る。

外部へのアクセスの安全を守るためのイントラネット内での管理のための手段としては意味がある。しかし、本来秘密鍵はユーザが保持すべき

であるが、この方法ではサーバで秘密鍵を保持しておりパスワード情報によりユーザを識別するため、安全性が低下する。さらに、ユーザの秘密鍵をサーバで管理することにより、サーバが攻撃にあった場合の被害が大きくなる。

4.4 SECRETsweeper[10]

SECRETsweeper サーバでサーバの秘密鍵と証明書を保持しており、クライアントから平文メールを受け取り暗号化や電子署名処理を行う。暗号時にはサーバとクライアント間は平文メールとなる。電子署名時にはサーバの秘密鍵による署名となるので、受取人による送信者の特定が不可能である。

5. まとめ

クライアントに証明書を保管するのではなく、ディレクトリサーバに保管しておき必要なときにダウンロードすることによって、ユーザによる証明書管理を削減する機能を有した S/MIME のアプレットを実装した。これにより手軽に S/MIME を使用することが可能となった。今回はイントラネットやエクストラネットでの使用を想定したが、今後はインターネットで使用する場合の検討を行いたい。

参考文献

[1] 若山公威, 福岡雄治, 岩田彰, 村瀬晋二, 鈴木春洋: 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, 第 58 回(平成 11 年前期)情報処理学会全国大会, 1999

[2] 高須紀樹, 若山公威, 岩田彰, 村瀬晋二, 鈴木春洋: 証明書と秘密鍵の集中管理のための SSL Proxy サーバ開発, 第 60 回(平成 12 年前期)情報処理学会全国大会, 2000

[3] 村瀬晋二, 若山公威, 鈴木春洋, 岩田彰: 暗号化メール自動診断機構の設計と実装, 第 60 回(平成 12 年前期)情報処理学会全国大会, 2000

[4] 奥野琢人, 若山公威, 村瀬晋二, 鈴木春洋, 岩田彰: 認証局パッケージ EasyCert の開発と評価, インターネットコンファレンス 2000, 2000

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

[6] Sun Microsystems: http://java.sun.com/products/javamail/index.html

[7] http://www.entrust.co.jp/products/products.htm

[8] 山崎直洋, 菊池浩明, 中西祥八郎: 暗号化サーバによる電子メールプライバシ強化の提案, SCIS, 1997

[9] 認証実用化実験協議会: http://www.icat.or.jp/pepop/

[10] Baltimore Technologies: http://www.us.mimesweeper.com/products/secretsweeper/techspec.asp