NPO 日本ネットワークセキュリティ協会
Japan Network Security Association

S/MIME検討WG
2004年度活動報告
「S/MIMEメールクライアントの機能検証結果報告」

磐城 洋介
(NTTコムウェア)

2005 年 6 月 13日

はじめに

インターネットが一般的になり、保険の申請などに電子メールアドレスを記載することも日常的になりつつあります。一方、現在の電子メールは「なりすまし」などセキュリティの問題が徐々に表面化してきています。

S/MIMEは、PKI(公開鍵基盤)により実現された、電子メールのセキュリティオプションです。

メールソフトにこの機能がどれだけ正確に実装されているか検証を行うことは、電子メールを安全に活用するためには、非常に重要なミッションになっています。

目次

電子メールの技術的な問題

現在一般的に用いられている、インターネットの電子メール（RFC2821）は、

- 発信者を示す「From」フィールドは発信者により任意のアドレスを設定できる。
  →偽メールの発信ができる
- 通信路上の電子メールは、プレーンテキストで伝送されており、ネットワーク上、あるいは中継サーバ上で内容の閲覧が容易にできる。
  →内容を、いつ、どこで、誰に見られるか分からない

などの、セキュリティ上の課題があります。

S/MIMEとは？

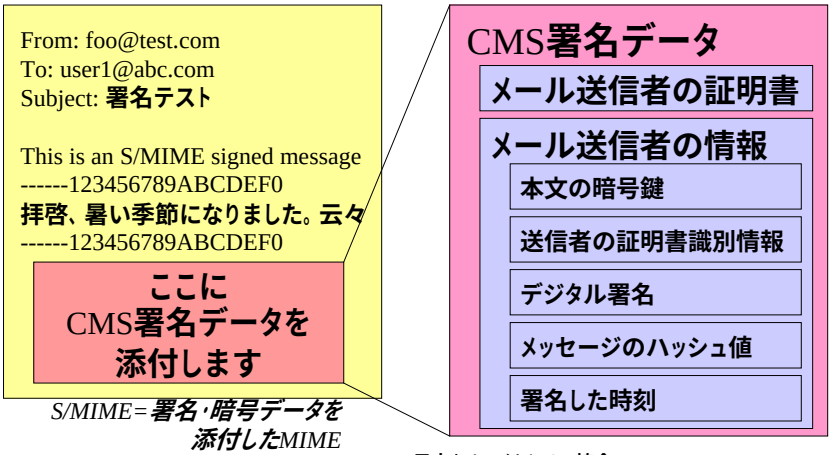
S/MIMEとは…

- 本人を特定する「署名」の機能がある
- 相手にしか内容を見られなくできる「暗号」の機能がある
- 伝送プロトコルには、SMTPを用いる
- S/MIME対応のメーラ（端末上のメールソフト：MUA）で利用できる
- メールサーバ上で、送信相手毎にメールを暗号化するタイプもある
- メールアドレスを含む個人の電子証明書が必要

などの特徴があります。

S/MIMEのメッセージ

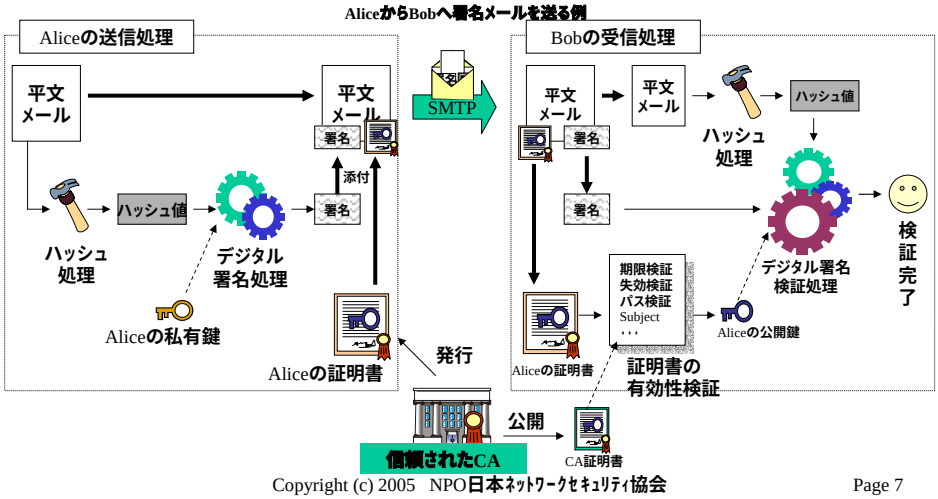
S/MIME – MIMEで署名・暗号データを添付したもの
CMS – S/MIME署名・暗号データの構造データフォーマット



S/MIME=署名・暗号データを添付したMIME

S/MIMEの機能（署名）

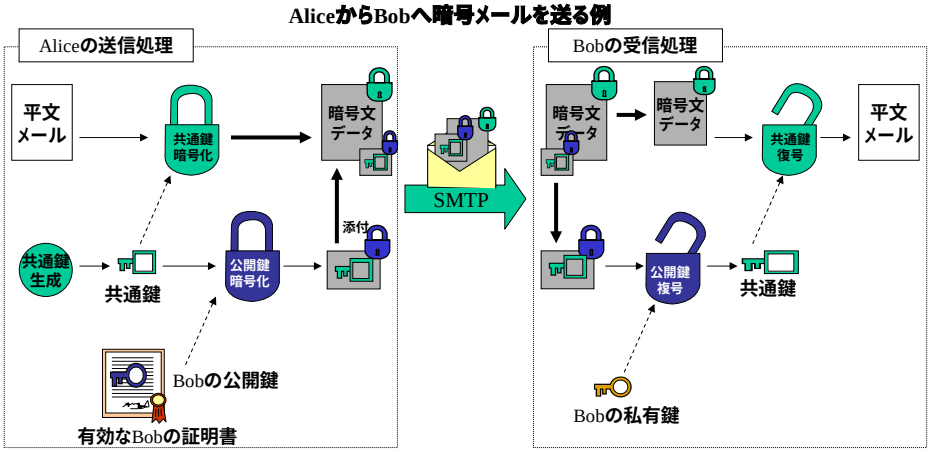
送信者はメールに対して自分の秘密鍵を用いて「デジタル署名」を施します。受信側では相手の電子証明書中の公開鍵を用いて署名を検証します。検証が成功すると、「メールが改ざんされていない事」、および「送信者が偽称されていない事」が確認できます。受信者の信頼点（トラストポイント）が正しく、送信者の電子証明書の有効性を検証できることが前提になります。



 Page 7

S/MIMEの機能（暗号）

送信者は都度作成する共通鍵でメールを暗号化し、その共通鍵自体を事前に入手した有効な相手の電子証明書中の「公開鍵」を使って暗号化します。受信者は自分の秘密鍵を用いて送られた共通鍵を復号し、メールを解読できます。



 Page 8

検証対象メーラー一覧

今回、検証を実施したメーラー一覧です。

系統	名称＋版数	記号	備考
MS系・MS製品	Outlook Express 6	OE6	
	Outlook 2003 SP1	OL23	
MS系・CAPI利用	鶴亀メール 3.71 Win	鶴亀	
	Justsystem ShurikenPro3 R2	Sken	評価版
	Becky! + 正規S/MIMEプラグイン	Becky	
NS系	Netscape 7.1 E	NSMS	
	Mozilla 1.7.1 E	Mzlla	
	Mozilla Thunderbird 0.7.2 J	TBird	
独立系	Winbiff + SGomaV2	Wbiff	
OpenSSL系	N Gnus 0.2 + OpenSSL-0.9.7a	Gnus	On Linux
	OpenSSL smime 0.9.7a	OPSSL	On Linux
	mutt-1.5.6i-ja + OpenSSL-0.9.7d	mutt	On Linux

 Page 9

対象メーラの選定について

S/MIMEの検証なので、まずはS/MIME機能を実装していることが必要なのですが…

意外に少ない！（特にフリーのもの）

個人的、emacsベースのメーラをずっと愛用し続けていたのですが、ここで手放すことになるとは（とほほ）。

と言う訳で、無理やり「mutt」なるメーラの検証を行ったのでした。

＃Macユーザの方なら「Eudora」とか。
＃ZAURUSユーザなら、どうなんだろうか。

 Page 10

検証項目の選定基準

検証項目は、「署名検証」と「暗号復号」など、S/MIMEメール受信者側の機能を優先しました。理由は下記のとおりです。

- 昨今の「偽メール」対策に注目が集まっており、署名の機能は重要だから
- その割には、PKIでは極めて重要な署名検証機能の実装が、アプリ（メーラ）によりバラつきがある（との噂があった）ため
- 検証方法が比較的簡単だから

機能については、

- 添付された発信者の証明書検証が正しくできているか？
- 添付された署名の検証が正しくできているか？
- 検証結果が分かりやすく利用者に表示されているか？

と言う観点で、９６項目の検証を行いました。

検証項目一覧

主な機能分類毎に、下記の項目数を設定しました。

	項目数	
基本的な処理・送信者表示	22	証明書の期限切れ、改ざん検知、送信者（メルアド）の確認など
証明書拡張の正しさ	11	証明書拡張に関するPKIの標準をどれだけ守っているか。偽証明書対策など。
署名アルゴリズムの種類	12	対応している署名アルゴリズム（公開鍵暗号・ハッシュなど）の種類の多さ。
証明書の失効検証	19	CRLやOCSPなど失効方法の種類の多さ。
複雑なPKIモデルの証明書検証	16	相互認証などの証明書検証。
各種アルゴリズムによる暗号化と署名	16	暗号化アルゴリズムの種類の多さ。
	96	

評価方法基準

各検証項目は、セキュリティ的な観点でクリティカルとそれほどでもないものとに分類できます。
- •証明書とメールヘッダのアドレス一致を見ていない
- •証明書の検証をしていない

などは、明らかにセキュリティホール相当と言えますが、
- •証明書主体者情報の大文字･小文字の一致
- •HTTPプロキシ経由のOCSP検証

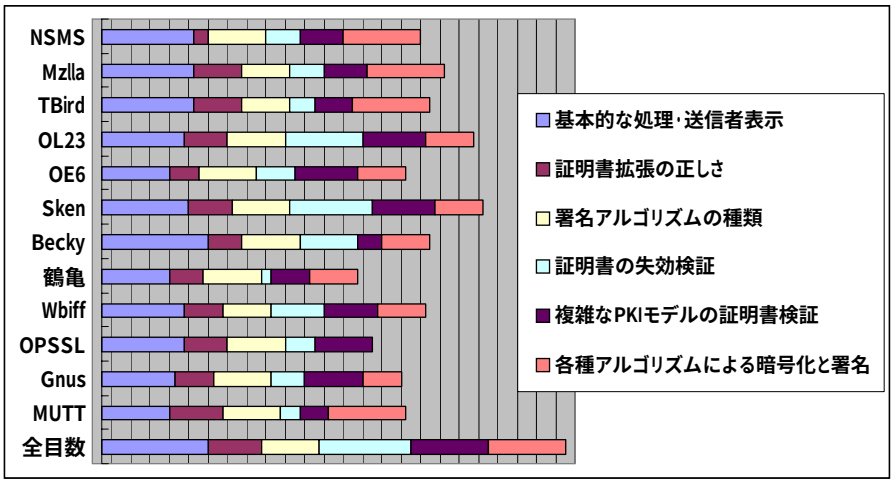
などは、運用上の工夫などで解決できます。

上記の観点で各検証結果に重み付けを行うことも検討しましたが、S/MIMEメーラの評価は、これら安全性に関する機能的な検証の他に、利用者に対する操作性や結果表示の適正性など総合的に判断する必要もあり、次年度の課題としました。

今回、各検証項目について期待値どおりであった場合に、「1ポイント」として加算し、結果としました。

評価結果

検証項目毎に結果が期待値どおりだった数。最下段は全項目数。



総評

【結果の総評】
- Microsoft系（MS製品、CAPI利用製品）がなかなか優秀。
  → 失効検証・複雑モデルで高得点。
- 証明書の失効検証は課題（未サポート製品多し）。
- 送信者情報（メールアドレス）について、From、Senderヘッダに関して問題あり（次項）。

【次のテーマ・課題など】
- ユーザインタフェースの定量的な判断方法の検討
- 検証項目毎の重み付け
  → セキュリティホールと言える内容の線引き

From、Senderヘッダの問題

- From、Senderヘッダでは以下が表示できる
  - メールアドレス
  - フレンドリー名 <メールアドレス>
  - メールアドレス (コメント)
  - ここで、メールアドレス、フレンドリ名、コメントをわかりやすい区別無しで表示するために、フレンドリ名やコメントに虚偽のメールアドレスが書かれていた場合、ユーザは惑わされる。
  - 製品ベンダーには、このわかりやすい表示や警告を望む。

S/MIME相互運用性のための推奨プロファイル（結論として…）

今回の検証結果から、S/MIME用の証明書やメール発信のルールを下記の条件で行った場合に、より多くのメーラでS/MIMEが利用できると考えられます。

- Fromに複数のメールアドレス指定は不可にする
- Senderのメールアドレスによる署名は不可にする
- 未知のクリティカル拡張は不可(RFC違反)にする
- 失効検証のためには（MS系、NS系の双方で相互運用）、CAがHTTPURI CRL+OCSPを提供する
- basicConstraintsは必須にする
- keyUsageはOE等見てくれないので期待しない。CA証明書では本拡張を使わない。NS系では存在すると証明書ストアに自動登録できないので特に指定しない
- MS系、NS系双方PrintableStringはバイナリマッチングなので、DNマッチングはバイナリ比較されると仮定して証明書を発行する
- 全てのS/MIMEクライアントはクリア/オペイクどちらも対応しているが非S/MIMEクライアント向けにクリア署名を使う
- 信頼モデルは基本か階層モデル(相互承認を含む)にとどめる
- RFCではメールアドレスをsubjectAltNameに入れる(SHOULD)としているが、NS系が表示できないのでsubjectにも入れる。subjectAltNameに複数のメールアドレスは入れない
- 共通鍵にAESを使わない

謝辞

「漆嶌（セコム）」さんによる多大な功績（検証の環境と試験データ作成、さらにmutt以外の全メーラの検証）により本報告が実現しました。
「水落（NECソフト）」さんには、検証用の利用者証明書発行環境（水落CA）のご提供を頂きました。
「宮川（ネットアーク）」さんには、S/MIME機能の解説ページの編集をご担当いただきました。

大変ありがとうございました。