暗号化電子メール PEM(Privacy Enhanced Mail) の実装と課題*

1M− 3

菊池 浩明　　森下 哲次†

(株) 富士通研究所‡

1 はじめに

計算機の相互接続による広域ネットワークの商用化に伴い，ネットワークアプリケーションのセキュリティが問題となっている．これは，広域環境では，ネットワークの盗聴やパケットの改竄が技術的に容易なためであり，こういった不正行為への対策として暗号技術の導入が強く求められている．そこで，実際に，電子メールを暗号化するシステム [1] が試作されているが，広域環境での運用には次の問題点がある．

1. **大規模ユーザの鍵管理** 世界中の全ユーザについて，暗号鍵を個人管理することは非現実的である．
2. **処理速度** 公開鍵暗号は処理時間がかかり実用的でない．
3. **公開鍵データベースの必要性** 公開鍵を全世界に安全に分配するサービスは未開発である．公開鍵を偽れば容易にユーザのなりすましが出来るため，公開鍵の配布にも安全性が必要である．

本稿では，Internet で標準化が進められている 電子メールの暗号化形式である PEM(Privacy Enhanced Mail)[2, 3] を取り上げる．これは，さまざまなネットワークの相互接続で構成されるという Internet の性質を考慮し，メールシステムや機種に特定しないで，通常のメールと併用できるという特徴を持っている．ただし，これまでは特許や輸出規制に関する問題などがあり，パッケージの公開を限定していたため，限定的な運用 (S.Kent によれば 92 年 9 月で 100 人未満) にとどまっている．そこで我々は，運用に向けての課題を明らかにするため，PEM の独自実装と評価を行なった．

2 Privacy Enhanced Mail

2.1 概要

先の広域環境における暗号化メールの問題点に対して，PEM では次の対策をとっている．

1. PEM では鍵管理に公開鍵暗号 RSA を利用している．これにより，各ユーザで管理する情報は 一部の公開情報と個人の秘密鍵のみとなる．
2. DES によるデータ暗号鍵 DEK と，RSA による交換鍵 IK の二種類の組合せで暗号化処理を行なう (2.2 節)．メール本文は DEK で高速に暗号化し，DEK のみを IK 復号化するためでパソコンでも実行可能な処理量となっている．
3. OSI で勧告されている X.509 の証明書 [4] を導入することにより，公開鍵データベースを不用としている (2.3 節)．

*Some Considerations on the Implementation of a Privacy Enhanced Mail System

†Kikuchi Hiroaki, Morishita Tetsuji

‡Fujitsu Laboratories Ltd.

2.2 暗復号化手順

ユーザ A がユーザ B に PEM を送信する手順を示す．

step1: データ暗号鍵 DEK をランダムに作り，DEK を用いてメール本文 P を DES 暗号化する．

step2: 受信者 B の公開鍵 IKP_B について，DEK を暗号化する．

step3: P をハッシュ関数 MD にかけて圧縮し，送信者 A の秘密鍵 IKS_A で署名する．

こうして計算された値をプリンタブル符号化し，暗号制御フィールド `Key-Info:`(DEK の暗号化), `MIC-Info:`(署名) として，通常のメールの本文にカプセル化して送信する．続いて受信者 B が PEM を復号化する手順を示す．

step4: `Key-Info:` フィールドを，B の秘密鍵 IKS_B で復号化し，DEK を取り出す．

step5: DEK で暗号文 C を復号化する．

step6: `MIC-Info:` フィールドを，A の公開鍵 IKP_A で復号化し，完全性を確認する．

以上の手続きにより，実用的な処理速度で盗聴と改竄を防止することが出来る．ところが，公開鍵 IKP_A,IKP_B を偽れば，不正ユーザによるなりすましを許してしまうために，次に述べる公開鍵証明書を導入する．

2.3 証明書による鍵管理

証明書とは，ユーザ識別名，公開鍵，発行者名，有効期限などの情報を 信頼できる認証局 CA の秘密鍵で暗号化した値である．CA の公開鍵を知っている者なら誰でもその正当性を確かめることが出来，一種の身元証明と考えられる．ユーザ A と B が共に CA の公開鍵 IKP_C を持っている時，A が自分の公開鍵を正しく B に伝える手続きは次のようになる．

step1: CA は，CA の秘密鍵 IKS_C を用いて，IKP_A を含む A の証明書を事前に発行しておく．

step2: A は B に証明書を，暗号制御フィールドとして送信する．

step3: B は，IKP_C を用いて A の公開鍵 IKP_A を取り出す．

これにより，各ユーザで IKP_C を管理するだけで，公開鍵のデータベースは必要がなくなる．ただし，一つの CAM で管理できるユーザ数には限度があるため，CA もより上位の発行局に発行局証明書を発行してもらい，CA を階層化する考えが提案されている．これにより，受信者が発行局の異なる別組織に属している場合でも，メールの接続性を保証することが出来る．

3 実装

RFC 標準案に従い，運用上の課題を検討するために ワークステーション (Sun SPARC Station/2) 上に PEM を試作した．これは，C で記述されており，一部の暗号処理にフリーソフトを利用している．基本的には RFC 標準案に従っているが，発行局を単一としたり，独自形式の証明書を用いる点などを簡略化している．

3.1 基本仕様

表 1: 暗号処理仕様

秘密鍵	p, q(256bit の素数), d(512bit 素数)
公開鍵	$n = pq$ (512bit)
共通鍵	$e = 65537$ (17bit)
符号化	RFC-1113 Printable Encoding
ハッシング	RFC-1321, MD5 Message-Digest
データ暗号方式	DES-CBC
交換鍵暗号方式	RSA
復号方式	Quisquarter(中国人剰余定理利用)
証明書形式	独自形式 (X.509 の形式は未対応), 1536bits
ユーザ識別名	RFC-822(メールアドレス形式)

3.2 実行例

```
From: Steve Kent <kent@bbn.com>
To: Hiroaki KIKUCHI <kikn@flab.fujitsu.co.jp>
Subject: Re: some questions on PEM
Date: Tue, 13 Oct 92 14:12:41 -0400
----PRIVACY-ENHANCED MESSAGE BOUNDARY----
Proc-Type: 3,ENCRYPTED
DEK-Info: DES-CBC,715657442D627330
Sender-ID: kent@bbn.com:Fujitsu:0
Certificate:
9e5TCcJJZsVZXegGTV8Uf4pt2teqa2nXLJY4ahnjiCaeTnKJ5ZLT/03pWxyP2cpK
R1K37TziDNCD+NTNJxBc4cOu8zewC+PXen94e11d4oYFjlFG+UAe7y8WgmMR37Wu
feow4OcTkbusrX/y81Pk5c5UGo86T9FRmmekS1N5QDGPX27iDXjNft6dOvhi2EX8
OXbSd6PKz/1596RWfoWt6gW5KU1qPFTt/QsN1gOvJxOpQuWVmeR2tgtvQptaZUsI
MIC-Info: RSA-MD5,RSA,
g+G+fjdlDw2/vBsHLhqFc2VMFLAyq+E7Zjm3gOblhOrdIi1JfslG1P+qMVKp14To
swZrrDHuEJR5xqD1I2JWCG==
Recipient-ID: Fujitsu
Key-Info: RSA,
H7205uOrZy2wRpv9pq717FfWL5atPfra9Puq7hvvPMf3E5Eo/wOnwhLiqX+beceb
mERTzkULeIKh2wMHSnRcx1==

94Dw02MMDFqMzCCX7Z6cmVitFqZTsGFPGjbCWFOyU5elVXKf186QTeSf46PTpdfp
qsF5o3e66jiEGgSpdCgoMIIqbNxri+aISxS2tky2gSwPRIfvejdScF8Szxj0fiEJ
q4M6UVKp8zjhz5EgnHmCCJ71bXLhPF1K4L5PmQWE5wb6I9gmiILNkhsX11DPGL3z
F6xOYgZFyDiYfV/NxHHhPxPozcufSIXtVkEyDisFay3HAkIaiX5187n53PBLwes/
5xJR5E7NgYeaXPor020cNTMiN7KByrZE5T26ta8qk6fsI21byc/5Kr==
----PRIVACY-ENHANCED MESSAGE BOUNDARY----
```

3.3 処理速度

Sun SPARC Station 2(30MIPS) で測定した各処理時間を表 2 に示す．ただし，測定は組み込みの time コマンドを用い，処理毎に複数回行ない平均を取ったものである．表中の処理速度は，ファイル長に依存する動的な処理を Kbps で，毎回固定の静的な処理を sec で示している．

表 2: 処理速度

	処理時間 [sec]			処理速度	
ファイル長	9612	106496	2598812	動的 [Kbps]	静的 [sec]
PEM 暗号化	1.9	2.9	28.9	767.2	1.80
PEM 復号化	1.7	2.4	19.6	1157.2	1.63
RSA 暗号化	17.8	177.8	4826.7	4.524	
RSA 復号化	197.6	1972.1	47890.6	0.2519	
DES 暗号化	0.0	0.5	10.6	1974.1	
DES 復号化	0.0	0.5	10.9	1917.1	
符号化	0.0	0.1	2.8	7384.6	
復号化	0.0	0.2	3.5	8668.9	
MD5	0.0	0.3	6.6	3164.8	

4 検討

1. **処理速度** PEM の処理では，多くの処理時間が必要な RSA の暗復号化処理が，ファイル長に関係なく一定である．よって，比較的大きなメールでも高速に処理することができ，実用的であると考える．なお，一般的なメールのファイル長を 2Kbytes (約 2 画面分) とすると，暗号化に 1.8sec，復号化に 1.63sec の時間が必要となる．

2. **CA の階層化と証明書の大きさ** 本実装では，CA は一階層，証明書は 512 × 3 bits の大きさに簡易化しているが，これを，n 階層，$512m$ bits の証明書に拡張すると，PEM の静的な処理時間が，暗復号共に $0.11mn$ sec 増加する．n, m の増加は，動的な処理速度には影響せず，RSA 復号化より比較的高速な RSA 暗号化の処理のみを増やす．従って，CA の高階層化や証明書の拡張による運用は可能である．

3. **安全性** 広域環境で想定される盗聴と改竄に対しては，暗号化と署名で防止しており，秘密鍵さえ管理していればなりすましも不可能である．ただし，秘密鍵をローカルにどうやって安全に管理するかという問題がある．

4. **証明書分配** 初めてメールする場合は，まず空のメールで証明書を交換しなくてはならず不便である．証明書は，公開情報であり，偽造できないという意味で改竄にも頑強であるため，ディレクトリサービスなどの分散データベースによる分配が 有効であると考えられる．また，ユーザが廃止されて無効になった証明書を広報するためにも，公開データベースは有益である．

5 おわりに

暗号化電子メール PEM を実装し，それについての速度評価を行なったところ，一般的なメールの暗復号化を 2 秒以内で処理できることが明らかになった．また，発行局の高階層化により組織間の通信を行なう場合にも，十分実用的な処理速度で対応できることを見積もった．残された課題としては，次のものがある．

- メイリングリストの鍵配布の問題．
- CA(発行局) の自動運用と発行形式の確立．

参考文献

[1] 曽根, 宇野, 森井, "電子メールシステムでの公開鍵暗号の適用とその一考察," 電子通信学会技術研究報告 (情報セキュリティ), ISEC91-44, 1992

[2] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I - Message Encipherment and Authentication Procedures," *RFC-1113*, 1989

[3] Kent, S. and Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part II - Certificate-Based Key Management," *RFC-1114*, 1989

[4] CCITT Recommendation X.509, "The Directory Authentication Framework," 1988