コンピュータセキュリティ 15－2
（２００１．１２．２１）

電子メール自動暗号化サーバの拡張について（4）

山口光平† 畑中雅彦† † 室蘭工業大学

これまで我々は、個人向け暗号化電子メール・パッケージにおける暗号化鍵の管理等の面倒な作業を解消する目的で、ネットワーク組織を単位とする自動暗号化処理サーバ(Crypt-Mail Secretary: CMS)について検討・試作・改良を行なってきた。しかし、CMSによるメール暗号化はドメイン内部を対象に行なうため、外出先からノートPC等の携帯端末を用いてメール通信を行なう場合、CMSを利用することは出来なかった。そこで、外出先からでも携帯端末を用いてCMSによる暗号化メール通信を行なえるよう、暗号化メールを自動的に送信先に転送するプログラム(User-Agent Proxy: UA-Proxy)の作成およびCMSへの実装を行なった。本稿では、外出先からの暗号化メールをCMSにより転送する方法の検討と、UA-Proxyを用いた転送機能の実現について報告する。

An Expansion of Automatic Cryptograph Server for Mail (4)

Kouhei Yamaguchi†, Masahiko HATANAKA†
† Muroran Institute of Technology

We had considered and made trial products about an automatic cryptograph server for mail (Crypt-Mail Secretary: CMS) to reduce personal trouble-some work of mail encryption (ex. management of encryption keys). Since mail encryption/decryption is based on the domain name of the mail addreess in this server, mobile user, who is out of his office, cannot use this server. To remove this limitation, we try to add a crypt-mail forwarding function in this server. Our crypt-mail forwarding is achieved by a "user-agent proxy", which automatically receives a mail through CMS, exchanges its mail address and send it.

1 はじめに

インターネットを利用した情報通信において最も一般的に利用されている電子メール・サービスは、盗聴に弱いことが指摘されている。そのため、電子メール本文を暗号化処理することは、セキュリティの面で非常に有効な手段とされている。こうしたことから、これまで数多くの電子メール暗号化パッケージが公開・製品化されてきた。しかし、これらの多くは個人を対象としたものであり、メールの暗号化可否判断や多数の通信相手との暗号化のための情報の交換・管理を利用者自身が行なわなければならず、利用者には非常に大きな負担となっていた。

そこで我々は、LAN等で区切られているドメインを単位として、ドメイン毎の共通の暗号化情報に基づいて電子メールを自動的に暗号化／復号化処理するメール・プロキシ・サーバ(Crypt-Mail Secretary: 以下CMSと略す)に

ついて検討 試作[1] し、改良を行なった[2][3]。

CMS を利用することの主な利点としては、

1. 組織のセキュリティ・ポリシーに基づいて、ドメイン間を流れる メールの暗号化を徹底できる。
2. 管理・運用すべき暗号鍵の数を減らすことができる。
3. 利用者に特別な操作をさせる必要が無い。
4. プライベートな メールに対する、既存の個人用 メールパッケージの利用を妨げない。

が挙げられる。

一方、モバイル通信網が発達した現代において、外出先などからモバイル PC 等の携帯端末を利用した電子 メール通信が盛んに行なわれている。これらの メールの暗号化の際にも、相手先に関する暗号化情報の交換・管理などが問題となる。

そこで、携帯端末の 1 つである ノートPC を用いて、CMS による暗号化 メール通信を行なうための方法について検討を行なった。そして、暗号化 メールを CMS 経由で送信して、CMS に送信先への メール送信および暗号化処理を行なわせるという方法を考案した。また、考案した手法を実現するために必要な機能を CMS に実装した[4]。

本稿では、我々が考案した CMS を経由する暗号化 メール通信の実現方法、および CMS の機能拡張について報告する。また、宛先の複数併記や送信先の宛先が無い場合への対処を行なうために、新たに CMS に追加した機能についても報告する。

2 CMS の概要

CMS の主な特徴には、以下のような ものが挙げられる。

- ○ 送信元・送信先の メールアドレスに含まれる ドメイン名 (‘@’ 以下の文字列) を基にして暗号化可否や暗号化方式を判断する。
- ○ ドメイン毎の暗号情報は CMS が管理する。
- ○ 既存の電子 メールシステムの構造に大きな変更を加えること無く暗号化／復号化処理を追加できる。

図 1 に一般的な メールシステムの流れと CMS を追加した メールシステムの流れを示す(動作については参考文献[1] を参照)。

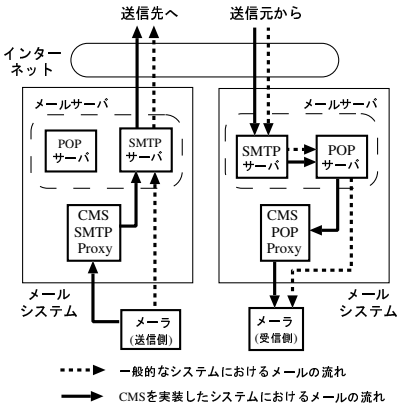

図 1: CMS を追加した メールシステムの構成図

3 CMS を経由する暗号化 メール通信

携帯端末を用いて メール暗号化通信を行なう時には、個人用の暗号化 ソフトを用いるのが一般的である。しかし、メール送信のたびに、メール送付相手一人一人に対する暗号化可否判断や暗号鍵の設定を行なわなければならず、利用者には非常に面倒な作業となる。

そこで我々が考案した方法は、携帯端末にて暗号化 ソフトを用いて暗号化した メールを、利用者自身が属する ドメインに設置された CMS 経由で通信相手に送るという ものである (図 2)。送信元では、あらかじめ メール本文の文末に通信相手全員の メールアドレスを付け加えておき、それを含めた メール全文を暗号化 ソフトで暗号化して、自身が属する ドメインの CMS へ送信する。メールを受けた CMS では、メールの送信先を、復号化した メールから取得した通信相手の メールアドレスに変更し、CMS 内で再び暗号化してから相手に送信する。すなわち、CMS に メールを中継する機能を持たせるのである。

この方法を用いることで、送信元が通信対象とする相手は CMS のみとなり、送信元の端末にて管理すべき暗号情報は 1 つで済む。また、通信相手への暗号化 メールの送信を CMS に一

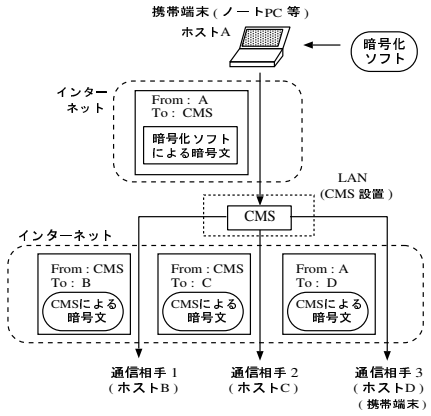

図 2: CMS を経由する暗号化メール通信方法

括して任せる事が出来るので、送信相手毎に暗号情報を設定するとい う面倒な作業が不要となる。

4 CMS 機能の拡張

携帯端末をクライアン ト対象とする暗号化メール通信を実現させるために、以下に示す2つの機能を CMS に追加した。

4.1 携帯端末 -CMS 間の双方向暗号化通信

CMS をメール中継に用いた暗号化メール通信を行な うには、CMS となるホス トと暗号化ソフ トを入れた携帯端末の間にて、個人単位による双方向暗号化通信を行な う必要がある。

そこで、CMS の鍵管理システムに、以下の機能を追加した。

○ 携帯端末側が生成した鍵情報を取得して、CMS が所持する暗号情報テーブルに登録する。
○ 携帯端末向けの鍵情報を CMS 内部で生成する。

これらの機能によ り、互いに生成した暗号鍵を交換することで、携帯端末と CMS 間での双方向暗号化通信が可能となる。

4.2 メール中継処理の自動化

CMS が メールの中継を行な うには、メール通信のクライアン トとなる UA(User Agent) が発する メール取 り込みおよびメール送信要求が必要となる。そこで、これら 2 つの要求を自動的に発信する機能と メール送信先を変更する機能を持つクライアン トプロ グラム (UA-Proxy) を作成し、CMS に実装した。UA-Proxy は、メールシステムおよび CMS の構成に極力変更を与えないよ うに、既に CMS に実装してある暗号化処理サーバ (SMTP-Proxy および POP-Proxy) とは独立して動作する。

図 3 に、UA-Proxy を実装した CMS を用いた、メールの暗号化通信の流れを示す。図 1 のメールシステム構成では、メールサーバに対するクライアン トはメー ラであ ったが、メールの中継を行な う際には、図 3 のよ うに CMS に追加した UA-Proxy がクライアン トとなって、メールの自動取 り込みおよび自動送信を行な う。

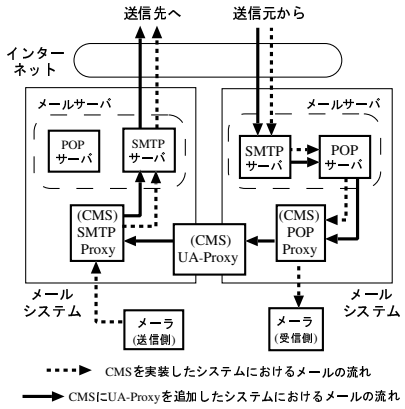

図 3: UA-Proxy を追加したシス テム構成図

4.2.1 UA-Proxy の機能

UA-Proxy が持つ主な機能について、以下に示す。

・ メール取 り込み要求の発信

図 4 は UA-Proxy による メールの取 り込み処理の流れを示している。UA-Proxy では、一定時間 (今回は 30 秒に設定) ごとに メール取 り込みのための POP コマン ド [5] を作成し、CMS 内にある POP-Proxy に送信する。POP-Proxy は受け取ったコマン ド群を POP サーバに中継し、サーバからの返答を UA-Proxy に返す。

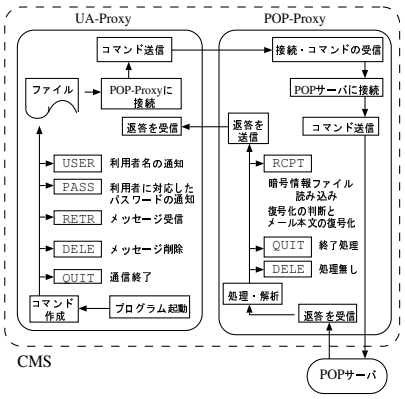

図 4: メール取 り込み処理の流れ

コマン ドの送信は、始めにユーザ認証と メール数取得のコマン ドを送る。その後、メール数取得コマン ドの返答として、メール数が 0 (メッセージボ ックス内に メールがない) と返 ってきた場合は、通信終了のコマン ドのみを送る。メール数が 1 以上の場合は メール取 り込み 削除と通信終了のコマン ドを送 り、POP サーバから メール本文が返されると、POP-Proxy にて送信元の メールア ドレスに対応した暗号情報に従 って メールの復号化処理を行な う。

・ メールア ドレス取得とヘ ッダ書き換え

UA-Proxy では、携帯端末から送られてきた メールを実際の送信先に送るために、送信先の メールア ドレスを取得し、メールヘ ッダの内容を書き換える処理を行な う。

送信側では、暗号化処理を行な う前に、メール本文中に実際の送信先の メールア ドレスを記述しておく。メールを受けた CMS では、復号化した メール本文から送信先の メールア ドレスを取 り出し、その メールのヘ ッダ情報にある送信先を示す欄 (To : で示される行) に取得した メールア ドレスを書 く。そして、CMS ホス トの メールア ドレスを、ヘ ッダ情報の送信元を示す欄 (From : で示される行) に書 く。また、元々の送信元の メールア ドレスは、ヘ ッダ情報から取 り出して メール本文に追加する。なお、送信先が外部接続の端末の場合は、送信先の変更のみを行な う。

・ メール送信要求の発行

図 5 は UA-Proxy による メールの送信処理の流れを示している。UA-Proxy は、メール取 り込みおよびヘ ッダ書き換え処理を終えた後に、メールを送信するための SMTP コマン ド [6] を作成し、SMTP サーバの代理サーバ (SMTP-Proxy) に送信する。コマン ド群を受けた SMTP-Proxy は、SMTP に沿った独自の返答を UA-Proxy に返し、送信先の宛先毎にコマン ドを格納する。SMTP-Proxy が メッセージ送信のコマン ドを受けとった場合は、送信先の メールア ドレスから得られる暗号情報を基に メールの暗号化処理を行な う。その後、SMTP サーバへの接続を行ない、格納していたコマン ド群を SMTP サーバに送信する。

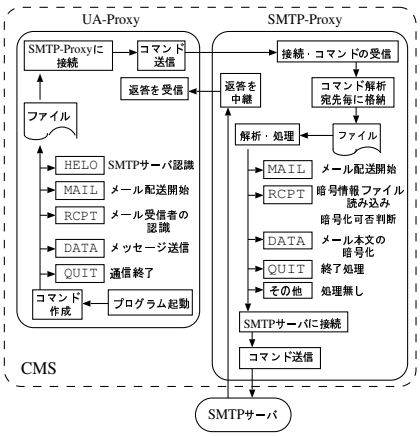

図 5: メール送信処理の流れ

4.3 宛先の複数併記への対応

CMS では、既に複数の宛先が併記されている場合についての対応がなされている [1]。そのため、携帯端末から複数の送信先に メールを送る場合でも、メール本文中に全ての送信先 メールア ドレスを書き込んでおけば、CMS にてそれぞれの宛先毎に暗号化／復号化の可否を判断して暗号化／復号化処理を行な う。

4.4 宛先を書いていない メールに対する処理

メール本文中に送信先の メールア ドレスが書かれていなかった場合は、メール中継の時に、エ ラーを知らせる メッセージを メール本文中

に追加し、送信元のメールアドレスから判断される暗号化方式によって暗号化したメールを返送する。エラーを知らせる場合でもメールは暗号化されるので、メール本文に対するセキュリティは確保される。しかし、送信先のメールアドレスが間違っている場合の送信元への対処がまだ成り立っておらず、今後検討を行なう必要がある。

5 実装

5.1 開発言語・開発環境

CMS を構成するプロキシ・サーバ群は、インターネットプログラミングが容易で移植性も優れている Java 言語で書かれている。そのため、今回実装を行なった UA-Proxy の構築やプログラムの修正にも Java 言語を用いた。また、Java の開発環境には JDK(Java Developers Kit) の Ver1.1.8 を利用した。

5.2 暗号化ライブラリと暗号化ソフト

CMS では、メールの暗号化方式として、共通暗号鍵方式の DES (Data Encrypt Standard) 、IDEA (International Data Encrypt Algorithm) 、次世代暗号化標準 (Advanced Encrypt Standard: AES) として選定された Rijndeal *、および一般的に用いられている暗号プログラム PGP (Pretty Good Privacy) の 4 種類を採用している。これらのアルゴリズムの実装には、暗号化ライブラリである Cryptix3.1.3 † を利用している。

また、ノートPC で用いる暗号化ソフトには、インターネット上にて簡単に入手できる PGP Ver2.6.3i を採用した。

6 動作確認実験

機能拡張した CMS の動作確認を行なうために、電子メールの送受信実験を行なった。

6.1 実験環境

実験を行なうに当たり、学内 LAN 上にメールシステムを四つ用意し、それぞれサイトA、B、C、D とした (図 6 参照)。サイトB、C は SMTP サーバと POP サーバおよび CMS で構成し、サイトA、D は SMTP サーバと POP サーバのみで構成する。各サイトに設置した SMTP サーバと POP サーバは同一ホスト上にて動作する。また、クライアントとして、サイトA、D にはノートPC の代わりとなるデスクトップ端末を置き、サイトC にはデスクトップ端末を置いた。

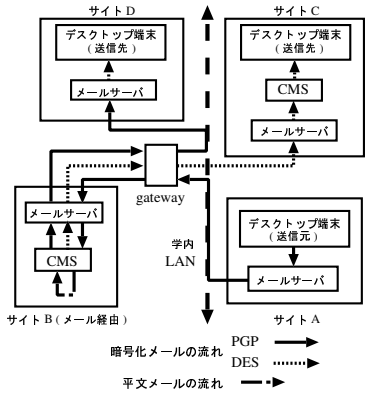

図 6: 実験環境

6.2 使用ソフトウェアと計算機

本実験にて各サイトで用いたソフトウェアおよび計算機の OS を表 1 に示す。

サイトA	OS	ソフトウェア
デスクトップ端末	HP-UX 10.20	mew-1.93, PGP 2.6.3i
メールサーバ	HP-UX 10.20	sendmail 8.9.3, qpoper 2.5.3

サイトB	OS	ソフトウェア
CMS	FreeBSD 4.2	JDK1.1.8, Cryptix3.1.3
メールサーバ	KONDORA MNU/Linux 1.1	sendmail 8.9.3 qpoper 2.5.3

サイトC	OS	ソフトウェア
デスクトップ端末	FreeBSD 3.4	mew-1.94.2
CMS	FreeBSD 3.4	JDK1.1.8, Cryptix3.1.3
メールサーバ	FreeBSD 3.4	sendmail 8.9.3, qpoper 2.5.3

サイトD	OS	ソフトウェア
デスクトップ端末	WindowsNT 4.0	WinYAT32 Ver 4.0, PGP 2.6.3i
メールサーバ	SunOS 5.5.1	sendmail 8.9.3 YAT/POP 4.01p4

表 1: 使用ソフトウェアと計算機 OS

6.3 実験方法

以下の 3 種類の方法で実験を行なった。

1. サイトA から、サイトC あるいはサイトD の宛先を書いたメールをサイトB の CMS

*http://www.esat.kuleuven.ac.jp/ rijmen/rijndeal
†http://www.cryptix.org/

に送り、CMS のメール中継動作を見る。
2. メール本文中にサイトC とサイトD の宛先を併記し、そのメールをサイトA からサイトB の CMS 経由で送信する。
3. メール本文中に送信先を書かずにメールを送信し、サイトB の CMS の動作を見る。

各サイト間のメール暗号化方式は、サイトB-C 間をDES、サイトB の CMS とサイトA、D の端末間を PGP に設定した。暗号化／復号化処理およびメール中継処理の動作については、メールサーバおよび CMS を構成するサーバ群の間で送受信されるデータを標準出力で表示することによって確認した。

6.4 実験結果

前節の項目 1 については、サイトA の端末から送られたメールが、サイトB の CMS にて、送信先をサイトC またはサイトD のメールアドレスに書き換えられた後に、そのメールがサイト間で定めた暗号化方式によって暗号化され、それぞれのサイト宛に送信されることを確認した。そして、メールの内容が途中で変化することなく通信されていたことも確認した。

また、項目 2 のように複数の宛先を併記した場合でも、サイトB の CMS が宛先ごとに暗号化処理を行なってから、それぞれのサイトに送信していることを確認した。

項目 3 については、サイトB の CMS が、メール本文にエラーを知らせるメッセージを追加し、それをサイトA-B 間で決めた暗号化方式を用いて暗号化してからサイトA に返送したことを確認した。

7 まとめ

今回、CMS にノートPC を用いた暗号化メール通信を行なうための機能を実装し、その機能が正常に動作することを確認した。これにより、ノートPC を始めとする携帯端末を外出先のネットワークに接続して用いても、CMS を設置したドメイン内部から行なう場合と同等の状態で暗号化メール通信を行なう事が可能となる。さらには、新たに宛先の複数併記への対処と送信先が書かれていないメールへの対処を行なうための機能を追加したことで、実際のメールシステム上で利用できる仕様に近づけることが出来たと言える。

また、CMS でメール中継する際に、メール送信先が組織内部の端末か外部接続の端末かを判別した上で暗号化処理を行なうことで、送信先の端末形態を気にせずに暗号化メール通信を行なうことが出来ると言える。

今後の課題としては、現在、暗号鍵の交換 登録を手作業で行なっている CMS の鍵管理システムについて、大幅な見直しを行なっていく予定である。また、見直しに当たって、当研究室で研究が進められているセキュリティパッケージ IPsec(Security Architecture for the Internet Protocol) を用いた自動鍵交換機能 [7] の実装も考慮している。

参考文献

[1] 永江由紀子 他 :“電子メール自動暗号化処理サーバの構築 (2), ” 情報研報 99-CSEC-4, pp.61-66 (1999)

[2] 永江由紀子 他 :“電子メール自動暗号化処理サーバの構築 (3), ” 平成 11 年度電気関係学会北海道支部連合大会講演論文集, p.425 (1999)

[3] 山口 光平 他 :“電子メール自動暗号化処理サーバの構築 (4), ” 情報処理北海道シンポジウム 2000 講演論文集, pp.104-105 (2000)

[4] 山口 光平 他 :“電子メール自動暗号化処理サーバの拡張について (3), ” 平成 13 年度電気関係学会北海道支部連合大会講演論文集, p.425 (2001)

[5] Pop Office Protocol, Request For Comment (RFC) 1939

[6] Simple Mail Transfer Protocol, Request For Comment (RFC) 821

[7] 太田 貴雄 他 :“自動鍵交換機能つき CryptMail Secretary の開発, ” 平成 13 年度電気関係学会北海道支部連合大会講演論文集, p.425 (2001)