

Emacs における構造化メールの統合インターフェイス

3D-2

櫻井三子

山本和彦

日本電気株式会社

奈良先端科学技術大学院大学

1. はじめに

RFC822 で定義されたヘッダと本文以外に構造を持たないメールに代わって、本文に構造を持つ多目的メール(MIME)^[1]や、プライバシー強化メール(PEM)^[2]が定義されている。MIME も PEM も RFC822 との互換性は重視しているが、互いの利用に関しては十分な考慮がなされないまま、独立に規定されている。

様々なデータ形式を扱う MIME の利用は、今後 RFC822 に代わり普及するであろう。従って、MIME 形式のメールに対して PEM を適用するような場合が増えることが予想されるため、MIME や PEM のような構造化メールを統合する必要性が高まってくる。

また、構造化メールの普及を図るためには、Emacs などユーザの多いエディタ上で使い易いインターフェイスを提供することが重要である。このような観点から、本稿では、MIME と PEM を統合する方式や、Emacs 上のメールインターフェイスにおける実装について議論する。

2. MIME と PEM の統合

MIME は様々なデータ形式を扱うための枠組であるのに対し、PEM は電子署名や暗号化といった特化したサービスを提供するための枠組である。よって、構造化メールの統合は、PEM をより広い枠組である MIME の形式で捉えるのが自然である。そこで、MIME と PEM が構造的に合わない点についてまず整理し、PEM を MIME の中で扱うための方式について考察する。

Integrated Interface for Structured Mail on Emacs

Mine Sakurai, NEC Coporation

Kazuhiko Yamamoto, Nara Institute of Science and Technology

2.1 両者の構造上の相違点

MIME と PEM の構造の特徴を比較すると次のような相違点がある。

- 構造化のための境界の定義が両者で異なる。
- PEM の場合、PEM であることを示すヘッダフィールドが定義されていない。

MIME には、本文に複数のデータ領域を持たせることができる multipart という構造があり、任意の境界で part を区切る。この境界は、メール本文内で一意となる文字列として与えられる。一方、PEM では、PEM の領域を示すための境界として、始めと終りを示す固定文字列が使われる。

MIME では、Content-type ヘッダフィールドから各データのデータ形式を容易に検出できるが、PEM では、メール本文内で PEM 用の境界と同じ文字列が現れる可能性があり、本文が PEM であることを確実に識別する方法は存在しない。

2.2 統合方式の概要

PEM を MIME の枠組で捉えるためには、項 2.1 で述べた相違点を解決しなければならない。そのために、MIME の枠組の中で PEM 用の part (以下 PEMpart と呼ぶ) を定義する方法が考えられている。すなわち、内容が PEM であることを示す Content-type を定義し、PEMpart に PEM の結果を挿入するのである。

PEMpart のための Content-type や PEMpart 内の構造の定義等については、文献[3]や文献[4]において議論されてきた。現状では、PEMpart を multipart として構成する方式^[3]に統一されつつある。

PEMpart が定義されると、メール本文中の PEM の検出が正確に行える。また、PEMpart が multipart に含まれる場合は、MIME で定義された境界が利用されるため、境界の相違点を吸収できる。

ところで、PEM を MIME の中で扱う場合には、PEM の適用範囲に注意する必要がある。PEM では改ざんの検出を行うため、PEM を適用する領域の内容は、電子署名前と復号後で変化してはならない。よって、MIME の処理を受けても内容が変化しないように、PEM の適用範囲を選ぶ必要がある。例えば multipart の場合、multipart 全体、あるいは、個々の part ごとに PEM を適用しなければならない。PEM と MIME の統合インターフェイスを実装する場合、ユーザが PEMpart を正しく構成できるための工夫が必要である。

3. 統合インターフェイスの実装方法

節 2. で述べた統合方式では、PEM を MIME の枠組で捉えるため、統合インターフェイスは、基本的に MIME のインターフェイスである。

本文の構造を意識したメールのインターフェイスに対するユーザの要求として以下の点が挙げられる。

- メールを表示・構成の際の構造の可視化
- 構造解析の速さ
- 操作の粒度の細かさ

さらに、PEM を統合するインターフェイスへの要求として、次のような要求がある。

- PEM の自動復号
- PEM の処理の速さ

このような要求を満たしたインターフェイスの実装方法を、表示と構成に分けて述べる。

3.1 表示

操作の粒度を細かくするためには、multipart の解析結果を part 別に表示し、任意の part をユーザの好みの順序で読めるようにすべきである。また、MIME の構造解析や復号、あるいは、PEM の復号化は時間がかかるため、処理後にこれらをキャッシュするとよい。そうすれば、同じメールを再度読む場合にこれらの処理を省略することができる。

3.2 構成

ユーザが構造化メールの構成を行っている際に、一旦指定した構成を変更することはしばしば起こると予想される。従って、メール本文の構造が構成

時に見える機能を提供し、構造を簡単に変更できるようにすることが重要である。例えば、multipart の part 単位の削除機能等が考えられる。

また、PEMpart を構成する際には、multipart の part 単位で PEM の適用範囲の指定や取消を指定できる機能が必要である。これには、PEM の適用範囲をユーザが正しく指定できるという効果がある。さらに、メール本文中の全ての PEMpart を送信直前に一括して構成すれば、PEM の暗号処理を呼び出す回数を最低限に押える効果が得られる。

4. おわりに

節 3. で述べた方式の実装は、Emacs 上のメールインターフェイスである Mew (Message interface to Emacs Window)で行っている。文献 [3] では PEM の仕様が全面的に改定されているので、既存の PEM を使用することができない。そこで既存の PEM を利用する一時的な解として、文献 [4] で定義された “application/pem-1421” を Content-type として利用した。Mew では Content-type の値が異なれば、それぞれの処理を独立に定義できるように設計してある。そこで、文献 [3] で定義された方法に代わっても、柔軟に対応することが可能である。今後は文献 [3] の進展に着目しながら、統合インターフェイスの実装を進める予定である。

最後に、WIDE プロジェクト security WG および multimedia WG のメンバー、ならびに Mew メーリングリストのメンバーに感謝する。

参考文献

- [1] N. Borenstein and N. Freed, “MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies”, RFC 1521, September 1993
- [2] J. Linn, “Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures”, RFC 1421, February 1993.
- [3] Steve Crocker, Ned Freed, Jim Galvin, and Sandy Murphy, “PEM Security Services and MIME”, working draft, July 1994.
- [4] J. I. Schiller, “An Alternative PEM MIME Integration”, working draft (obsoleted), October 1993.