暗号化アルゴリズムの動的選択システム ·EMAAS

Dynamic Selection of Encryption Algorithm ·EMAAS

祢宜 知孝[†] 伊藤 純一郎[††] 所 真理雄[†††]
Tomonori NEGI Junichirou ITOH Mario TOKORO

[†]慶應義塾大学 理工学研究科 計算機科学専攻
[††]慶應義塾大学 理工学研究科 計算機科学専攻/ソニー CSL
[†††]慶應義塾大学/ソニー CSL

概 要

現在提供されている暗号システムを用いた通信では，受信者が使用された暗号化アルゴリズムに対応した復号アルゴリズムを所持していなければ，送られて来た暗号文を復号することができなかった．すなわち現在の暗号通信方式は，送信者が，受信者が所持していない暗号化アルゴリズムを使用できるような柔軟性を有していない．そこで，本稿では送信者が受信者の所持していない暗号化アルゴリズムを用いて暗号化した暗号文を送信した際も，受信者による復号が可能なシステムを提案し，実装する．

1 はじめに

近年のインターネットの広がりと共に，従来は軍隊や金融機関など，特殊な組織の間だけで重要視されていたセキュリティ技術が，一般ユーザにも重要なものになってきた．情報の盗聴や，改竄，なりすまし等コンピュータネットワークを通じてさまざまな不正が行われており，ユーザは自らこれらの不正から身を守らなければならない．

不正に対する防衛策として，情報の盗聴に対してはデータの暗号化，情報の改竄に対しては一方向ハッシュ関数を用いた電子署名，なりすましに対しては認証などがある．暗号化アルゴリズムとしては，DES [2]，SAFER [3] 等多数の方法が提供されている．また，暗号システムとしては，PGP，PEM 等が提供されている．

しかし暗号化一つをとってみても，上記のようにさまざまなセキュリティ技術が提供されているにもかかわらず，ユーザは，不正対策をせずに平文のままデータをネットワーク上に流している．原因の一つはユーザの多くが不正行為を特殊なケースと考え，不正行為に対して認識が甘いことがである．また，現在提供されている暗号システムは，あらかじめ使用する暗号化アルゴリズムを決めておかなければならないなど，使用における厳しい制約が存在していることも原因として挙げられる．

そこで本稿では，受信者が所持していない暗号化アルゴリズムで暗号化されたデータであろうと受信者による復号が可能であり，面倒な操作を必要としないシステムを提案する．

2 背景

現在提供されている暗号システムを用いて暗号通信を行う際には，送信者 ·受信者間で使用する暗号化アルゴリズムに関する合意が取れており，かつ両者がそのアルゴリズムを所持している必要がある．例えば CPU パワーのある計算機 A と CPU パワーのない携帯計算機 B間で暗号通信を行うと，マシン Bのユーザは重要度の低いメッセージについても計算量の多い暗号化 ·復号アルゴリズムを用いなければならない．この時，マシン Bのユーザはメッセージの重要度に合わせて計算量の少ない暗号化アルゴリズムの使用を希望するが，マシン A のユーザとそのアルゴリズムを利用することについて合意していないため不可能である．

また，暗号化アルゴリズムの強度は，鍵の取り得る値の範囲の広さ」と『復号化にかかる計算量」によって定められている．現在の暗号化アルゴリズムは暗号の安全性が計算量に裏付けされている以上，時が経ち計算機の計算パワーが増大すると共に，暗号強度が相対的に弱くなって行くという致命的な問題を抱えている．

現在提供されている暗号システムには，利用者がメッセージの重要性とマシンの性能に合った暗号化アルゴリズムを送信時に自由に選択し，使用するといった柔軟性はない．

3 方針

本節では，2節で述べた問題点をまとめ、現在提供されている暗号化アルゴリズムを利用した上でこれらの問題点を解決するようなシステム，**EMAAS(Encrypted Message And Algorithm System)** の方針について述べる．

3.1 問題に対するアプローチ

2節で述べたように，現在の暗号システムでは大きく以下の 2 点が問題として挙げられる．

問題点 1 あらかじめ送信者 ･受信者間で使用する暗号化アルゴリズムの合意が必要．

問題点 2 暗号化アルゴリズムの強度は，CPU の性能向上に伴い，時と共に相対的に弱くなっていく．

上記の問題点 2 つを解決する方法として EMAAS は次のような方法を取る．使用した暗号化アルゴリズムに対応する復号アルゴリズムを，暗号文と共に受信者に送信する (図 1)．これにより，あらかじめ送信者 ･受信者間で使用する暗号化アルゴリズムを合意しておく必要がなく，両者がそのアルゴリズムを所持している必要もない．アルゴリズムを送信することによって，暗号化アルゴリズムを他人に知られてしまうということに一見問題があるように思える．しかしアルゴリズムの強度は暗号解析時間のみによって決まるので，アルゴリズムを知られることによるアルゴリズムの強度自体への影響はない．

また，従来のシステムでは送信者 ･受信者があらかじめ使用する暗号化アルゴリズムに関して合意をした上で，両者がそのアルゴリズムを所持していなければならかった．従って，暗号化アルゴリズムの強度が弱くなってしまっても，より強度の強い暗号化アルゴリズムに変更することは困難であった．しかし，EMAAS では送信者が自由に使用する暗号化アルゴリズムを選択できるので，その時々にあったアルゴリズムを使用することが可能となり，アルゴリズムが計算機の性能の向上と共に弱くなっていくことに容易に対処できる．

3.2 基本動作

ここでは EMAAS の動作手順について述べる．

送信者 ･受信者は幾つかの暗号化アルゴリズムを所持しており，送信者は暗号化アルゴリズム $E_1, E_2, \cdots, E_m, \cdots, E_n$ とそれぞれに対応すれぞれに対応する復号アルゴリズムを $D_1, D_2, \cdots, D_m, \cdots, D_n$ を所持しているとする．また送信者 ･受信者は各自が使用するマシンの性能に関する情報を所有しているとする．

送信者が，受信者に暗号文を送る際，メッセージは以下の手順で送信され，受信側で復号される．

1. 送信者側の EMAAS によって E-mail を用いた受信者とのネゴシエーションが行われ，送信者は受信者のマシンの性能，受信者の所持している暗号化アルゴリズム ･復号アルゴリズムに関する情報を入手する．
2. 送信者は，送信するデータの重要性とネゴシエーションによって得た情報を基に，使用する暗号化アルゴリズム E_m を選択する．
3. 送信者は，選択した暗号化アルゴリズム E_m で平文を暗号化する．この時，暗号化アルゴリズム E_m を受信者のみが所持している時には，送信者は受信者から暗号化アルゴリズム E_m を貰ってから暗号化する．
4. 送信者は，暗号文と復号アルゴリズム D_m を受信者に送信する．
5. 受信者は，送信されてきた復号アルゴリズムと暗号化されたデータを分割する．
6. 送信者は復号アルゴリズム D_m を実行し，暗号化されたデータを復号する．

4 実装

本稿では，EMAAS の実装例として E-mail 配送システム上に EMASS を組み込んだ場合について述べる．ユーザインターフェイスには，Mew(Message interface to Emacs Window) [1] を拡張して利用する．

E-mail を配送する際には，E-mail の符合化方式を決めなければならず，いくつかの E-mail の符合化

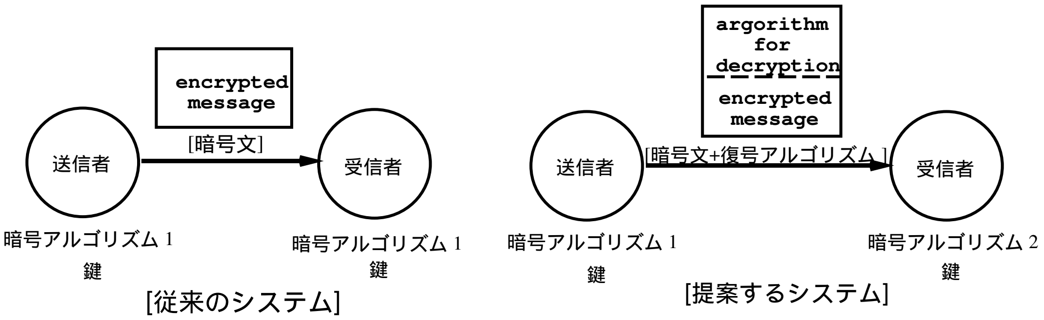

図 1　従来のシステムと提案するシステム

方式が規格化されている．そこで本稿では，E-mail の符合化方式として MIME [4] を選択し，新たな MIME の Content-Type を提案する．新たに提案する Content-Type は，既に統一規格となっている “Application” を利用した “Application/emaas” である．

暗号化アルゴリズム・復号アルゴリズム記述言語として，Emacs-Lisp を用いる．Emacs-Lisp を用いる理由として次のことが挙げられる．

- Emacs-Lisp は C 言語等のようにマシンによって異なるソースプログラムを用意しておく必要がなく，マシン依存の少ない，互換性の高い言語である．
- Emacs-Lisp は Mule か Emacs さえ動作していれば実行可能であり，Mule や Emacs は多くのユーザに使われている.
- Emacs-Lisp は記述性が高い言語である．
- EMAAS が，インターフェイスとして Emacs，Mule 上で動作する Mew を使用するため，Emacs-Lisp は EMAAS と相性が良い．

E-mail 配送における EMAAS の動作図を図 2に示す．

5　評価・検討

表 1に示すように，従来は送信者・受信者両者が同一の暗号化アルゴリズムを所持している時のみ，メッセージを暗号化して送信することが可能であった．しかし，本稿で提案した EMAAS は送信者・受信者のいずれか一方が暗号化アルゴリズムを所持していれば，メッセージを暗号化して送信することが可能である．これにより，データの重要性，マシン性能に合わせた暗号化アルゴリズムの使用が可能となり，柔軟性に富んだ暗号化が可能となる．

次に Emacs-Lisp で記述した暗号化アルゴリズム SAFER の実行時間を表 2に示す．SPARC station 20(75MHz, SuperSPARC II) で実行した．

表 2　**SAFER** の実行時間

Datasize	暗号化時間 [sec]	復号時間 [sec]
256 Byte	0.63	0.58
512 Byte	1.21	1.20
1024 Byte	2.40	2.41
2048 Byte	4.73	4.70

Emacs-Lisp で記述しているため実行時間が多少かかるが，十分使用に耐えられると思われる．

6　考察

本稿の実装では，改竄対策を行っていないが，今後改竄対策を行わなければならない．暗号文は改竄されると，復号した際に文章を読めず改竄されたことを知ることができる，また文章を読むことができないというだけで深刻な問題は起こらない．しかし，復号アルゴリズムの部分を改竄されると復号アルゴリズムを実行した時にパスワードの漏洩などさまざまな深刻な問題が起こり，しかも改竄の発見が困難である．

そこで復号アルゴリズムが改竄されても深刻な問題が起こらないように，アルゴリズム記述言語のオペレーションを限定する必要がある．つまり，アルゴリズム記述言語は暗号化・復号に必要最低限のオペレーションのみにし，E-mail の送信，ファイルの書き込み等に繋がるオペレーションを禁じる必要がある．オペレーションを制限した言語として Safe-Tcl 等の使用を検討する．

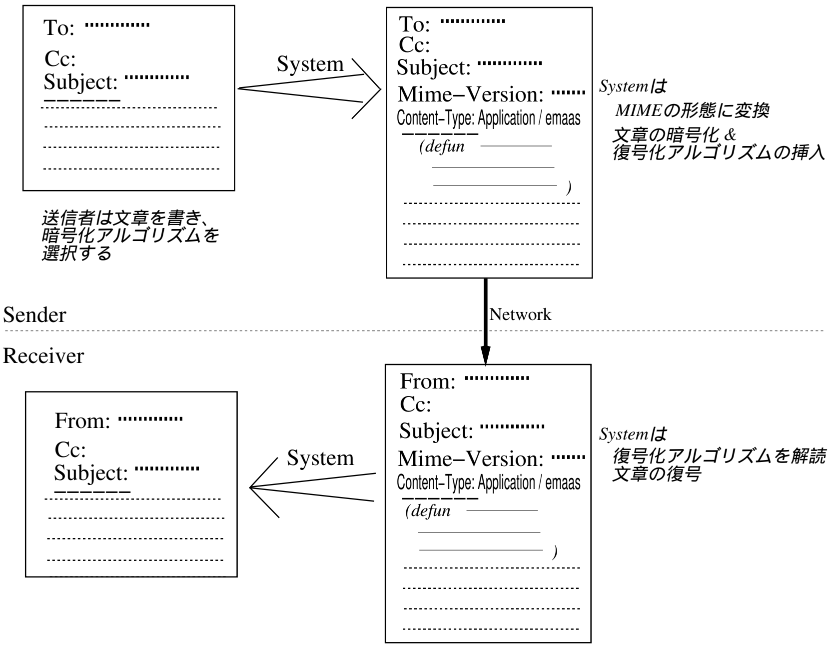

図 **2　EMAAS** の動作

表 **1**　従来のシステムと **EMAAS** の柔軟性

暗号化アルゴリズムの所持・不所持		通信可能	
送信者	受信者	従来	EMAAS
○	○	◎	◎
○	×	×	◎
×	○	×	◎
×	×	×	×

7　今後の課題

今後の課題として以下のことが挙げられる．

1. 本稿では，既に鍵を共有していることを前提としたので，今後鍵を安全に配送するプロトコルを，EMAAS に組み込む．
2. 本稿では，EMAAS の E-mail 配送システムへの実装を行ったが，EMAAS は E-mail に限らず様々なネットワークプロトコル上へも応用が可能であるので，他のプロトコルへの応用について研究する．また，暗号化アルゴリズムに限らず，様々な処理プログラムもデータと共に送信することが可能であると思われる．
3. 送信されてきた復号アルゴリズムの正当性を保証する機構について検討する．

8　結論

本稿で提案した EMAAS は，以下のような特徴を持つ．

- ユーザは通信相手がどのような暗号化アルゴリズムを所持しているかを考慮せずに任意の暗号化アルゴリズムを用いて暗号通信を行える．
- E-mail 配送システムに限らず，他の様々な通信プロトコルにも応用可能なシステムである．

参考文献

[1] 山本　和彦，　桜井　三子，及村　能成．Mew - 構造を持つメールへのインターフェイス．マルチメディア通信と分散処理ワークショップ，10 月　1994．

[2] Whitfield Difie and Martin E. Hellman. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10(6):74–84, June 1977.

[3] James Masscy. SAFER K-64 A Byte-Oriented Block-Ciphering Algorithm. *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer Verlag, 1–17, 1994.

[4] Ned Freed and Nathaniel Borenstein. *Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies.* Technical Report, May 1995.