FleaMarket 情報流通システムのグローバル化 *

明石 修　　森保 健治　　寺内 敦†
NTT ソフトウェア研究所 ‡

概要

情報は非常に小さなコストで複製や移動が可能である点に特徴があり、本質的に従来の物理的な流通システムの制約を受けない。我々はこの性質に注目し、インターネット上で登録された情報を暗号化すると同時にカプセル化により保護して自由に配布し、ユーザが必要な時に復号鍵を取得することにより情報を購買する FleaMarket 方式情報流通モデルを提案し、その基本機能を実装した。

本稿では、FleaMarket 情報流通システムを大域的な系に適用するための要求条件を、1) 各サービスモジュールの複数サーバ化とその相互接続性、2) 複数の暗号化形式を扱うためのカプセルの構造変更と機能拡張、3) 情報の登録および管理系、の観点から述べ、その解決方法を示す。

1　はじめに

電子化された情報は、非常に小さなコストで複製や移動が可能であり、本質的に従来の物理的な流通システムの制約を受けずに電子的な手段を用いて柔軟に流通させることが可能である。一方情報には著作権が存在し、情報を保護しアクセスを制御する必要がある [1, 2]。

インターネット環境でディジタル情報の特質を活用し、なおかつ著作権を保護する方法として、我々は FleaMarket 方式による情報流通モデル [3, 4] を提案し、その基本機能の実装に関する問題点と解決方法 [5] を述べた。

本モデルでは、情報提供者はインターネット上の情報登録サーバ (=FleaMarket) に情報商品を登録し、FleaMarket は情報商品を保護するため暗号化し、カプセル化する。商品情報を復号化する復号鍵は FleaMarket から鍵サーバに送られる。カプセルは、インターネット上で自由に配付するため、デジタル署名と認証子関数により、生成者の確認と、改竄の検知を可能とした。

ユーザは復号時に、認証、購買手続きを 1 トランザクションとして実行し、鍵サーバから復号鍵を入手する。この鍵配送は通常のインターネット上のデータ転送経路や利用者端末が攻撃される危険性を考慮した Key Delivery Protocol(KDP) を定義して用いる。また端末側モジュールは、アプリケーション層に直接鍵を見せない形のインタフェースとする。

このような FleaMarket 情報流通システムを、インターネットのような大域的な系に実サービスの枠組みとして適用するためには、システムを構成する各サービスモジュールを複数化することと、柔軟なカプセル化の機能を提供することが必要である。また、情報を登録し、その属性を管理する機能も重要である。

本稿では、まず FleaMarket 情報流通モデル、システムの概要を説明し、各サービスモジュールごとに必要な機能を述べる。次に本システムを大域的な系に適用するための問題点と要求条件を、

---

*Enhancement of FleaMarket Information Distribution System for Global Environment

†Osamu Akashi, Kenji Moriyasu, Atsushi Terauchi
akashi@nuesun.ntt.jp, {moriyasu,terauchi}@slab.ntt.jp
‡NTT Software Laboratories

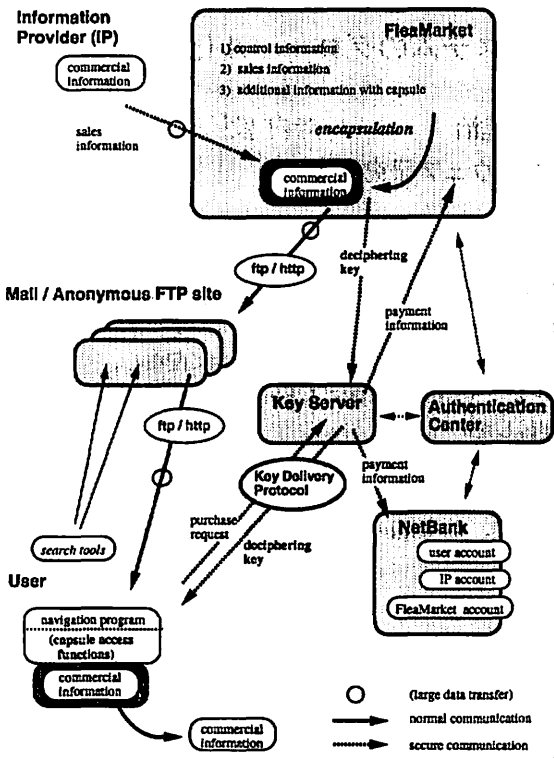

図 1: FleaMarket 情報流通モデル

1) 各サービスモジュールの複数サーバ化とその相互接続性、2) 複数の暗号化形式を扱うためのカプセルの構造変更と機能拡張、3) 情報の登録および管理系、の観点から述べ、その解決方法を示す。

2 FleaMarket 情報流通モデル / システムの概要

本章では、FleaMarket 方式による情報流通モデル (図 1) の概要を説明する。ここでは簡素化のため、各サービスモジュールは 1 つとする。

本モデルでは、情報提供者はインターネット上の情報登録サーバ (=FleaMarket) に接続し、認証手続きを行なったのち、情報商品と、定価 (= 登録時の実買価格)、販売期間、商品説明文を登録する。オプションとして、デモ情報、鍵配送時に無料で送付する付加情報も登録可能である。

FleaMarket は情報商品を保護するため、暗号化と同時にその他の制御情報を埋め込みカプセル化する。その後、復号鍵を販売情報と共に鍵サーバに送付する。情報商品をインターネット上で自由に配付可能とするため、カプセルは以下のような機能を、公開鍵暗号を用いた電子署名と認証子関数を用いて提供する [5]。

- ユーザは商品が正規の FleaMarket で作られたことが確認可能
- ユーザは商品が改竄されていないことが確認可能

ユーザはカプセルアクセス関数を通じて、平文でカプセルに埋め込まれた定価[1]、商品説明文、デモ情報にアクセスすることが可能である。カプセルは、FleaMarket から、インターネット上の商店、あるいは簡易に anonyous FTP サイトに自由に配付される。これは前述したカプセルの機能により、セキュアな環境で保存する必要がないためで、標準的な ftp あるいは HTTP を用いてダウンロードすることが可能である。またカプセル自体をデータキャッシュとして扱ったり、ローカルなホストに置いておき、Pay Per View 的な使い方をすることも可能である。

ユーザは、カプセルアクセス関数中の商品復号関数を、1 トランザクションとして実行して認証 / 購買手続きを行い、同時に鍵サーバから復号鍵を取得し商品を得る。このように大きな商品データ (= カプセル) の流通と、その購買手続き (= 鍵配送) をモデル上で分けることで、システム負荷の分散も行なう。

この一連の手続きは、セキュアな鍵配送プロトコル (KDP)[6, 5] を TCP/IP 上に定義し、電文の暗号化による第三者による盗聴防止に加えて、改竄検知、成りすましの防止機能を提供し、アプリ

[1] 実売価格は鍵サーバが持つ

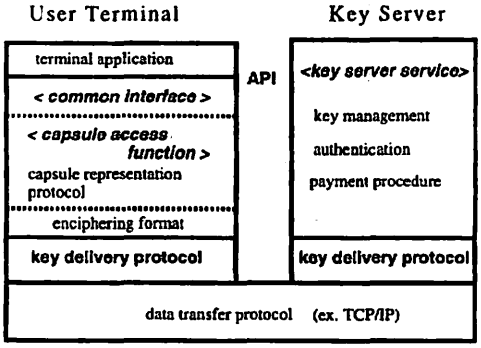

図 2: プロトコル階層

ケーション層に復号鍵を出さない形で行なわれる。システムのプロトコル階層は、図2に示す通りである。

決済系は、モデル上 **NetBank** として抽象化して定義する。実装にあたっては、鍵サーバからは独立したモジュールである決済処理関数を呼び出し、鍵サーバでは解釈しない暗号化されたままの決済データを引数として渡す。その内部では、決済種別を識別し、クレジットカード決済、あるいは電子プリペイド [7] といったさまざまなサービス固有の決済システムを呼び出す。

3 サービスモジュールの複数化

FleaMarket モデルを大規模な系へ適用するためには、負荷の分散や異なる運営主体によるサービスを可能とするため、それぞれのサービスモジュールを複数化し独立して動作させる必要がある。サービスモジュールを複数化する場合、識別子 (ID) を割り振り、そのIDを処理の各フェーズで管理し、相手の識別やアドレス検索を行なう必要がある。このとき、既に実装されているカプセルの構造や鍵配送システムとの整合性を考慮する必要がある。要求条件としては、以下が上げられる。

- 各サービスモジュールが、正規の (= 登録された) ものであることが、証明可能
- カプセルの改竄検知機能等、既に実装されている検証システムが正しく動作する

CD-ROM 情報流通システムでは、CD ごとに一意な CD-ROM ID(= 発行者ごとに一意に割り振った ID+ 発行者が管理するロット ID)、および情報ごとに割り振る情報 ID の組で情報を一意に識別していた。

FleaMarket システムでは、CD-ROM と同様に、カプセル毎に一意なカプセル ID (発行者 ID+ 発行者が管理するカプセルシリアル番号)、およびカプセル毎に割り振る情報 ID の組で情報を識別する。FleaMarket に割り振る発行者 ID は一意になるように鍵サーバ群で管理する。FleaMarket は、この発行者 ID_i を認証システムに登録し、正規の契約した FleaMarket であることの証明に用いる公開鍵暗号の秘密鍵 $K_{f,i}^{-1}$ を得る。この ID_i はカプセルに埋め込まれ、$K_{f,i}^{-1}$ により電子署名がされる。

なお認証システム自体の認証は、認証システムの公開鍵 K_a を広く既知であると仮定し、その鍵を用いてメッセージを暗号化して送る。すなわち K_a^{-1} を持つ認証システムしか読めないことで相手を特定する。

ここで正規の FleaMarket が、誤って、あるいは意識的に異なる発行者 ID_j を用いたと仮定する。この場合は ID_j と対となる $K_{f,j}^{-1}$ を知らないため、その FleaMarket の作ったカプセルは、購買時のカプセル発行者同定機能で誤りが発見される。

また情報の登録時には、異なった組織が運営する複数の FleaMarket サーバがあるため、正しく ID を割り振られた正規の FleaMarket であることが認証できる必要がある。これも同様に、ID_i に対応した $K_{f,i}$ を使うことで成りすましの防止を行なうことができる。

また復号鍵を管理する鍵サーバも複数化する。これは、負荷分散が主な目的であり、運営は 1 つの組織で行なうことを前提とする。FleaMarket の ID は、このサーバ群で一意になるように管理する。どの鍵サーバに復号鍵を預けるかは、カプセル作

成サーバの運営ポリシー(鍵サーバ運営主体との契約)による。本モデルでは、カプセル生成時に預け先である複数の鍵サーバを確定し、その名前のリストをカプセルに埋め込む。IPアドレスではなく、名前を用いるのは、物理的なアドレスの変更に対処可能とするためであるが、ネームサーバが必須となる。

カプセルに埋め込まれた複数の鍵サーバのうちどれを使用するかは、アプリケーションがカプセルアクセス関数により操作可能である。提供されるAPIは以下の通りである。

- 鍵サーバの名前リストとインデックスを取り出す
- 使用する鍵サーバを、インデックスで指定する…直接アドレスを入力するAPIは提供しない

アプリケーションによる鍵サーバの選択は、ユーザ端末の接続されたネットワークの環境や、ユーザのポリシー等によりさまざまであるため、カプセルアクセス関数の外から指定する形とする。従って、帯域等の関係から、指定した複数の鍵サーバに対して鍵購買トランザクションを順次行ない、成功した時点で終了するという購買手続きを実現する場合は、アプリケーション側でトランザクションの終了コードに基づき、順次使用する鍵サーバのインデックスを変え、KDPを実行する関数を呼ぶ、という形を取る。

決済方法は、現実装においても、クレジットカード決済か電子プリペイド方式かを選ぶGUIがある。鍵サーバの基本部分は、決済系あるいはサービス依存部分との独立性を保つため、KDPで運ぶ決済情報の中味は解釈しない。しかしそこから呼び出す決済処理関数[2]が決済種別を識別し、更に実際の決済システムに依存した固有の処理関数を呼び出す。従って、決済種別の増設は、KDP中の新たな課金種別を定義し、それを識別し実決済システム固有の処理関数を課金データと共に呼び出すように決済処理関数を変更する必要がある。

[2]別モジュールとして設計

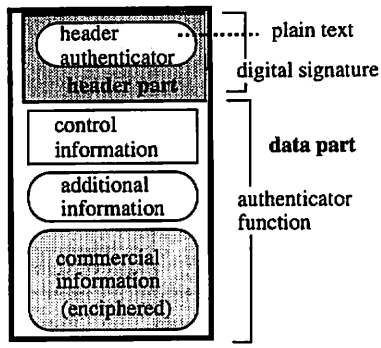

図3: カプセルの構造

4 カプセル機能の拡張

カプセルは公開鍵暗号を用いた電子署名と認証子関数を用いて、生成者の確認と、改竄検知機能を実現する[5]。基本的な構造を図3に示す。

基本機能の確認に加えて、カプセルをプラットフォームとして実サービスに適用するに当たって、以下のような要求が存在する。

- 複数暗号化フォーマットへの対応
- 1つのカプセルで、複数の鍵で暗号化された複数のファイルを扱う機能
- 複数の復号鍵の同時配送と、その状態管理

4.1 複数ファイルの取り扱い

従来は暗号化された1つのファイルを扱うのみであったが、以下のような要求が存在する。

- 商品情報が自体が、ディレクトリ構造をもち、複数のファイルからなるような場合、そのままの形で流通させたい
- 複数のファイルのうち、任意のファイルを指定し、場合によってはそれぞれ異なった暗号

化形式で暗号化できるような柔軟性を持たせたい

- カプセルファイル中のディレクトリ構造やファイル属性は、いつでもアクセス可能 (= 平文)

そこで、ディレクトリ構造やファイルの属性を表すデータ[3]をカプセル化し、カプセルアクセス関数を通じて、API からそのデータにアクセス可能とする。またファイル毎の属性の一つとして、暗号化形式の値を持たせる。

ディレクトリ表現としては、UNIX ファイルシステムの i-node 式の構造体を用いる。すなわちファイルの属性を表す構造体と、その構造体へのポインタを用いて、ディレクトリ構造を表現する (図 4)。ディレクトリ情報は、暗号化しないが、カプセル自体の改竄検知機能によりデータの改竄は不可能である。具体的な構造体の中味は、以下のとおりである。

- ファイル名またはディレクトリ名
- ファイルの属性 ··· ファイル種別 (ファイル / ディレクトリ)、暗号化の有無、暗号化形式 (暗号関数、フォーマット)
- 販売情報 ··· 情報 ID、定価
- 他の構造体の名前とポインタの対、またはデータの実体へのポインタ

カプセルアクセス関数は、"/" から始まる階層構造を持ったパス名としてファイルを表現する。ディレクトリ情報をアクセスするために提供する API は以下の通りである。

```
scan_cap_dir(カプセルファイル名, パス名, *
ディレクトリが含む名前のリスト)
struct ディレクトリ情報 {
  u_long  実体へのポインタ
  char [] 名前;
}
```

[3]以降、ディレクトリ情報と呼ぶ

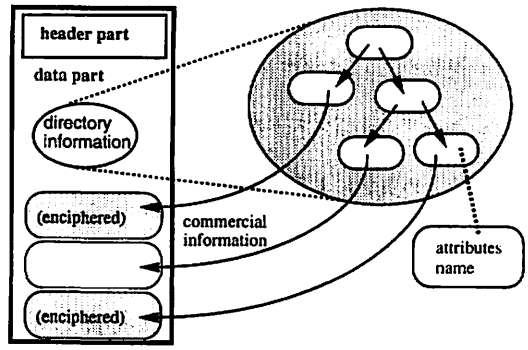

図 4: カプセル内のディレクトリ情報

```
get_cap_finfo(カプセルファイル名, パス名, *
ファイル情報構造体)
struct ファイル情報構造体 {
  u_long ファイル属性;
/* ファイル / ディレクトリ、暗号化の有無、暗号
化形式 */
  u_long ファイルサイズ;
  u_long 平文ファイルサイズ;
  u_long 情報 ID;
  u_long 定価;
  u_long 実体へのポインタ;
}
```

その他、カプセル全体の情報をアクセスする get_cap_info() は、ほぼ従来どおりであるが、情報 ID 等の詳細情報取得は上の get_cap_finfo() を用いるように変更する。購買を行なう decode_cap_file() は、カプセルファイル名に加えて、カプセル中のファイルをパス名で指定するように API を変更する。

4.2 複数鍵配送

カプセルが複数のファイルに対応したのに伴い、同一ファイル内の複数の情報商品をまとめ買いす

る機能を追加する。目的は以下の通りである。

- 複数商品/1カプセル化、への対応
- ユーザの通信コストの削減
- まとめ買いによる値下げ

例えば電子出版で、各章ごとに暗号化されたファイルからなる構成を考える。特定の章だけを選択して買う場合に比べて、何章かまとめて買うユーザにディスカウントして販売するサービスに適用可能である。このようなまとめ買い(KDPによる同時複数鍵配送)で問題となるのは以下の点である。

- ディスカウントの種別
  - 情報提供者の入力と変更
  - カプセル内での表現とサーバでの表現
- トランザクションの管理
  - コミット/アボートの定義と管理
  - 途中中断と再会
- 複数のファイルを指定する購入モジュールAPI
- 決済DBの集計システムの変更

まとめ買いによるディスカウントは、現在、まとめ買いするファイルの個数、まとめ買いする販売価格、の2種の軸を設定する。それが一定値x_iを超えた場合、割り引き率$d[i](i = 1 \ldots n)$が適用される。nの最大値は、1カプセル中の暗号化ファイル総数である。ファイルの個数はカプセル内の情報を用いてユーザ端末で計算可能であるが、割り引き率は後で情報提供者により変更可能なため、サーバのデータの値が優先される。販売価格はもともと鍵サーバが持つ(=カプセルの持つ値は定価)ので、これもサーバの値を優先する。なお場合によっては、定価と割り引き率の組合せの方が、サーバの持つ販売価格と割り引き率の組合せより安価になる可能性もあるため、値段の確認画面を用意する。

複数鍵配送は、KDPのプロトコルでは既に規定されている。鍵要求時には購買する情報IDの数とそのリスト、鍵配送時には同様に対応する復号鍵の数とそのリストを送る。トランザクションのコミット/アボートは、単一鍵の場合と同様に、複数鍵でまとめて扱う。購買を実行するユーザ端末上のモジュールAPIには、カプセルファイル名の指定に加えて、カプセル内のディレクトリ表現で表した購買対象のファイル名のリストを加えることとする。

またサーバ側では、鍵サーバが記録する、売り上げ情報を管理するDBのレコードが簡易な(ユーザID, 値段, 情報ID)形式から、(ユーザID, 値段, トランザクションID)と(トランザクションID, 情報ID_i)のリストの集合に変更する必要がある。

5 情報の登録と管理系

5.1 商品情報の登録

情報の登録は、情報提供者はインターネット上の情報登録サーバ(= FleaMarket)で、ユーザ認証とセキュア通信機能を持つ鍵配送プロトコル(KDP)[6, 5]を拡張したプロトコルを用いる。

情報提供者は、あらかじめ配付された情報登録プログラムを用いて、情報購買時と同様に、ユーザID、パスワードを打ち込む(図5参照)。次に登録するファイルを選択し、使用する暗号化形式を指定する。

認証システムは、銀行、FleaMarketと共通の認証センタを用いることとする。SET[8]と同様に、鍵サーバはユーザ認証のための情報は暗号化されているために解釈不可能であり、データをそのまま認証センタに送る。鍵サーバは認証センタからのOK/NGに基づき、トランザクションを継続する。SETはデータの転送経路である商店が個人のクレジット番号等の情報を読まないために、このような手法を用いるが、本システムでは認証モジュールをFleaMarket、鍵サーバ、銀行と複数のサービスモジュールで共通化し、認証モジュールの独立

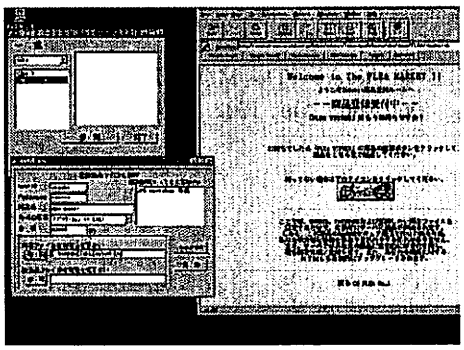

図 5: 情報登録画面

性を保つためでもある。また認証モジュールを独立させておくことにより、さまざまな認証システムとサービスを独立した形で、組込むことも可能である。

認証モジュールを共通化するのは、複数のサービスをユーザが利用する上で、それぞれのパスワードを管理する負荷を軽減する目的であるが、パスワードが漏れた場合の被害の範囲とトレードオフの関係にある。将来的にはレベルをいくつか設け、重要な銀行口座にアクセスするために必要なパスワードを多重化する等の防御作が必要となるが、本稿ではパスワードは 1 レベルとする。

ユーザが商品を購買する時は、KDP を用いたトランザクションを行ない、コミットした場合は、ユーザの口座から情報提供者の口座へ振り込みが行なわれる。

5.2 販売情報の変更

本モデルでは、販売情報でカプセル本体に埋め込まれているのは定価のみであり、実販売価格、鍵付加情報、販売期限という情報は、FleaMarket からデータを送付された鍵サーバが持つ。

要求条件としては、以下のような点が上げられる。

- 変更しようとしている情報の所有者であることが証明可能
- 複数回の変更トランザクションが識別可能

登録時には、FleaMarket から情報登録証

$$\{\{カプセルID, seq番号, IP名\}_{K_{f,i}^{-1}}\}_{passwd}$$

が情報提供者に発行される。これは、FleaMarket$_i$ の持つ公開鍵暗号方式の秘密鍵 $K_{f,i}^{-1}$ による電子署名であり、カプセルの所有者が誰であるかを証明する。seq 番号は、情報提供者がこの証明書を用いて、販売情報を変更する毎にインクリメントする管理番号である。*passwd* は、この証明書が情報提供者の端末に保存されている間にコピーされたとしても、*passwd* を知らない第三者には使用できないようにするためである。なおここで用いる *passwd* は、認証のためのパスワードとは無関係であり、カプセル毎に設定可能である。

情報提供者が販売価格、販売期間、鍵付加情報等の商品の販売情報を変更する時は、登録時と同様のアプリケーションを用い、変更する情報、証明書とその passwd を入力し、

$$\{カプセルID, seq番号, IP名\}_{K_{f,i}^{-1}}$$

を FleaMarket にセキュア通信を用いて送付する。

FleaMarket は、$K_{f,i}$ を用いて確認後、変更トランザクションを継続し、*seq*番号 をインクリメントした新たな証明書を情報提供者に発行する。その後、変更した販売情報を鍵サーバに送付し、変更が有効となる。なおこの時点で、パスワード情報は変更可能である。

6 おわりに

本稿では、登録された情報商品を暗号化し、カプセル化により保護した状態で、インターネット上を自由に配付し、ユーザが必要な時に購買手続きと復号を行なう FleaMarket 情報流通システムにおいて、インターネットのような大規模な系に適用する場合の問題点とその解決方法を、1) 各サービスモジュールの複数化、2) 複数の暗号化形式を

扱うためのカプセルの構造変更と機能拡張、3) 情報の登録および管理系、の観点から述べた。

FleaMarket システムの実装は、ユーザ端末は Windows95 上に、その他のモジュールは Unix マシン上で行い、基本機能の確認は行なった。本稿で述べた拡張機能は順次実装し、実サービスによる適用実験を行なう予定である。

参考文献

[1] R. Mori and M. Kawahara. Superdistribution: The concept and the Architecture. *The Transactions of the IEICE*, Vol. E73, pp. 1133–1146, 1990.

[2] 金井敦, 三宅延久, 明石修, 生沼守英. マルチメディア情報流通システム (InfoKet). In *95-DPS-70*. IPSJ, May 1995.

[3] 明石修, 森保健治, 寺内敦. FleaMarket 方式による情報流通. マルチメディア通信と分散処理ワークショップ. IPSJ-DPS, Oct 1995.

[4] Osamu Akashi, Kenji Moriyasu, and Atsushi Terauchi. Information Distribution by FleaMarket System. In *Third Int'l Workshop on Services in Distributed and Networked Environments*. IEEE, June 1996.

[5] 明石修, 森保健治, 三宅延久, 寺内敦. FleaMarket 方式による情報流通システムの実装. In *96-DPS-76*. IPSJ, May 1996.

[6] 森保健治, 明石修, 寺内敦, 三宅延久. 情報流通システムにおける鍵配送通信の構成法. マルチメディア通信と分散処理ワークショップ. IPSJ-DPS, Oct 1995.

[7] 寺内敦, 森保健治, 明石修. 情報流通システムにおける課金方式. マルチメディア通信と分散処理ワークショップ. IPSJ-DPS, Oct 1995.

[8] Master International. Secure Electronic Transactions. http://www.mastercard.com/set/set.htm.

[9] 池野信一, 小山謙二. 現代暗号理論. 電子情報通信学会, 1986.

[10] R.L. Rivest, A. Shamir, and L. Adleman. A method of obtaining digital signature and public key cryptsystems. *CACM*, pp. 120–126, Feb. 1978.

[11] S. Miyaguti. The FEAL Cipher Family. In *Proc. of Crypto '90*, 1990.

[12] R. Rivest. The MD5 Message-Digest Algorithm, Apr. 1992. RFC1321.