情報処理学会論文誌 Vol. 51 No. 2 635–647 (Feb. 2010)

# 機密情報共有に有用な情報フロー制御モデルの提案

荒 井 正 人[†1] 田 中 英 彦[†2]

知的財産や製造ノウハウといった企業の機密情報を取り扱うコンピュータシステムでは，情報漏洩の問題が深刻になっている．原因としては，情報の紛失や盗難，ワーム・ウイルスなどが大半を占める．対策として，可搬記録媒体やインターネットを介した情報共有を禁止することは有効ではあるが，公開可能な一般情報の取扱いにも制限を与えることになる．また，ワーム・ウイルスに感染しやすい環境から機密情報を隔離する技術としてサンドボックスモデルがあるが，機密情報であっても電子メールに添付して配布したり，可搬記録媒体に格納して持ち出したりすることが業務上必要な場合に適さない．本論文では，公開可能な一般情報と，社外に開示してはならない機密情報とが混在するコンピュータシステムにおいて，機密情報の共有と保護を両立させる情報フロー制御モデルを提案する．提案モデルでは，インターネットを利用するためのプログラムの実行環境と，機密情報を扱うプログラムの実行環境を，サンドボックスのような既存技術を用いて分離することで，ワーム・ウイルスに感染しやすい環境から機密情報を保護しつつ，自動ファイル暗号化機能と暗号ファイルのアクセス制御機能を組み合わせることで，必要に応じて可搬記録媒体やインターネットを介して機密情報を暗号文として安全に伝達可能とする．

# A Proposal for an Effective Information Flow Control Model for Sharing Sensitive Information

Masato Arai[†1] and Hidehiko Tanaka[†2]

Information leakage has become a serious problem for computer systems that handle a company's sensitive information, such as intellectual properties and manufacturing know-how. The majority of the causes can be attributed to loss or theft of information or worms and viruses. As a countermeasure, forbidding the sharing of information through removable media or the Internet is effective, but it also places restriction on the handling of general information that can be made public. Also, the sandbox model can be used to segregate sensitive information from environments that can easily be infected by worms or viruses; however, even sensitive information is sent as email attachments to various locations, and this model cannot be applied to business cases where information must be stored and carried out on removable media. In this article, we propose an information flow control model that is suitable for both sharing and protecting sensitive information on computer systems in which general information that can be made public and sensitive information that cannot be exposed outside the company are mixed. In the proposed model, sensitive information are protected from environments that can be easily infected by worms or viruses by segregating the environment for programs that use the Internet and the environment in which programs handling sensitive information are executed, using existing techniques such as the sandbox model. At the same time, by combining automatic file encryption and encrypted file access control, sensitive information can be safely transmitted as encrypted text through removable media or the Internet as the need arises.

## 1. はじめに

e-コマースや電子政府といった社会インフラとして，IT システムやインターネットの利用が拡大する一方で，社内機密情報や個人情報の漏洩といった問題が深刻化している．日本ネットワークセキュリティ協会の調査レポート[1)] によると，個人情報漏洩のインシデントは，新聞やインターネットニュースで報道された件数だけで年間 1,000 件近くにのぼり，その原因としては下記のようなものが大半を占めている．

- 組織または個人の管理ミスによる紛失・置忘れ
- 第三者による PC や記録媒体の盗難
- 宛先違いによる電子メールなどの誤送信
- ワーム・ウイルスの感染による情報流出

つまり，内部の人間に悪意がなくても多発しているのが現状である．これは個人情報に限った話ではなく，企業が所有する機密情報全般についても同様と考えられる．このことから，情報の持出しや電子メールへの添付を厳しく制限している組織が増えてきている．しかし，一般の従業員や職員が操作するクライアント PC には，インターネットアクセスから機密情報の閲覧・編集まで多目的に利用するものであれば，社内の機密情報だけでなく，インターネットからダウンロードした公開情報も存在する．また，機密情報であっても業務上特定の取引先の企業と共有しなければならない情報もある．したがって，情報の持出しや電子

†1 株式会社日立製作所
Hitachi Ltd.
†2 情報セキュリティ大学院大学
Institute of Information Security

メールへの添付を一律禁止することは，公開情報の利用や業務を妨げることにもなりうる．

従来技術として，万が一ファイルが外部に流出しても第三者からの閲覧を不可とするために，保護すべきファイルをすべて暗号化するシステム[2]も実用化されているが，ワーム・ウイルスがファイル作成者の権限でファイルを読み出してインターネットに流出する問題や，公開可能なファイルとの区別については言及していない．

公開情報の利用を制限しないものとしては，機密情報を含むファイルについて電子メールへの添付や可搬記録媒体へのコピーを検知する技術[3]がある．これは，あらかじめ指定された機密ファイルのシグニチャを算出してリスト化し，ユーザがアクセスするファイルのシグニチャとマッチングさせてその類似度から機密ファイルを検出するものである．ファイルの内容が多少改変されても検出可能であるが，事前の機密ファイルの指定に漏れがあると，正しく検出できなくなるという問題がある．

デジタル著作権管理システム[4]では，コンテンツを暗号化しておき，それを復号して閲覧可能なプログラムを専用プログラムに限定することでコンテンツを保護している．しかし，専用プログラムに限定することは，保護可能なコンテンツのデータ形式も限定されるなど，利便性の点で問題がある．

専用プログラムに限定しない方法として，機密情報を読み出したプログラムがそのデータをどう利用し，どこに出力するかを追跡・制御する手法も提案されている[5]．この手法では，読み出したデータに関して，どのデバイスへの出力を認め，どのデバイスへの出力を禁止するかといった出力先デバイスの可否情報を保護ポリシとして決めておき，保護すべきデータとつねに対応付けて保存するものとしている．しかし，ワープロソフトのように機密情報を閲覧・編集するためのプログラムであればネットワークへの出力を禁止してもよいが，電子メールソフトにも同様な保護ポリシを適用すると他拠点へのメール送信が業務上必要な場合に問題が生じる．このように情報の共有を考えると，保護ポリシというのはプログラムの用途によっても変わりうる．

たとえばインターネットを利用するためのプログラムと，機密情報を閲覧・編集するためのプログラムのように，用途の異なるプログラムとその実行環境を分離し，実行環境に応じて機密情報やネットワークのアクセス権（保護ポリシ）を切替え可能なものとして，強制アクセス制御機構[6]を備えたセキュアOS[7)–9)]のほか，WindowBox[10]やSVFS（Secure Virtual File System）[11]のようなサンドボックスモデルがある．いずれの方式も，インターネットを利用して公開情報の送受信を可能としながら，そのようなワーム・ウイルスに感染しやすい環境から機密情報を隔離できる点で有効といえる．しかし，機密情報であっても，隔離するだけでなく電子メールに添付して組織内の各拠点（異なるドメイン）や組織外の関係者に配布しなければならない場合もある．また，可搬記録媒体に格納して持出すことも業務上必要なケースがある．

外部からイントラネットを安全に利用可能とする方式[12]も，機密ファイルの持出しが不要となる点で有効であるが，組織外拠点をまたがったファイル共有や組織外の関係者への配布については言及していない．

以上のように従来方式では，機密情報をリスト化する手間を要するほか，機密情報の保護が機密情報を読み出したプログラムに依拠する部分が大きいといった問題がある．また，インターネットや可搬記録媒体を介した情報共有が制限され，業務に支障をきたす恐れもある．さらに，情報の共有と保護の両立を考えると，保護ポリシが機密情報やプログラムの種類によって変わりうるため，設定が面倒といった問題がある．

そこで本論文では，不特定多数に公開可能な一般情報と，社外に開示してはならない機密情報とが混在する企業のコンピュータシステムでの利用を想定し，従来方式では困難であった機密情報の共有と保護の両立を可能とする情報フロー制御モデルを提案する．さらに，提案モデルの実現の一形態について，既存のクライアントPCからの移行を考慮しながら説明する．なお，ここでの情報共有とは，社内の複数の拠点間や社外の関係者との間で，たとえば可搬記録媒体のほか，電子メールなどインターネットを介して行われるファイルの受渡しを意味している．

以下，2章で情報漏洩の対策方針について述べ，3章では本論文の主たる提案である情報フロー制御モデルについて述べる．さらに4章でその実現の一形態を示し，5章で利用イメージを述べる．そして，最後に提案モデルの有用性の検討結果を6章で述べる．

## 2. 情報漏洩問題の対策方針

紛失・置忘れ，盗難，誤送信，ワーム・ウイルスなどによって起こりうる情報漏洩のリスクを，業務上必要な情報共有を阻害することなく低減するために，具体的には，以下の方針を満たす情報フロー制御モデルを提案する．

- 公開可能な一般情報と，関係者以外に開示してはならない機密情報とを区別可能であること．
- 一般情報の利用はいっさい制限せず，機密情報については専用アプリケーションプログラムを用いなくても利用ならびに保護が可能であること．
- 機密情報であっても，必要に応じて可搬記録媒体やインターネットを利用しながら関係

者間で安全に共有可能であること．

- 上記，機密情報の共有と保護を両立させるためのアクセス権（保護ポリシ）の設定が容易であること

まずは情報を公開可能な一般情報と，非公開な機密情報との 2 つに大別する．機密情報については，関係者間で共有する鍵で情報をつねに暗号化しておき，権限のあるユーザであれば可搬記録媒体やインターネットを介して共有し，復号して閲覧・編集可能とする．そしてその閲覧・編集には，デジタル著作権管理システム[4] のような専用アプリケーションプログラムは不要とする．ただし，プログラムの振舞い（ファイルアクセスやネットワークアクセスなど）を監視・制御する方法[5] では，制御のためのアクセス権設定が容易ではない．そこで，プログラムに対して個別にアクセス権を付与するのではなく，情報の種別（機密と一般）に応じてプログラムの実行環境を構築・分離し，その実行環境に対して付与する．これは，実行環境が同じプログラムには共通のアクセス権を与えることを意味する．また，プログラム実行環境の分離は，ワーム・ウイルスが侵入しやすい環境から機密情報を保護するためにも有効と考える．

## 3. 情報フロー制御モデルの提案

2 章で述べた対策方針を満たす情報フロー制御モデルを，機密情報の暗号化と，機密情報を取り扱うプログラムの実行環境の分離，そしてファイルアクセス制御とネットワークアクセス制御機構の組合せにより構成する．

### 3.1 機密情報の暗号化

**図 1** に示すように，提案の情報フロー制御モデルでは，機密情報が持ち出されても，共有フォルダに格納されても関係者以外に対して隠蔽できるように，機密情報を含むファイル（以下，機密ファイル）はすべて暗号化して保存し，機密情報を含まない一般ファイルは平文で保存する．これはファイルの格納先がいかなる種類の記録媒体でも共通とする．機密ファイルの参照や編集を行う前にはユーザ認証を行い，機密情報の取扱い資格を持つユーザであれば，ファイルの暗号化・復号に必要な鍵の利用を許可する．また，過失により関係者以外の手に渡った機密情報を保護するために，復号鍵は取扱い資格を持つユーザのみが共有可能な鍵管理方式を提供する．

ただし，上記ファイル暗号化が果たす役割は，あくまでも取扱い資格（鍵）を持たないユーザに対して機密情報を隠蔽することであり，取扱い資格（鍵）を持つユーザによる下記の行為により機密情報が平文のまま流出するという問題は残る．

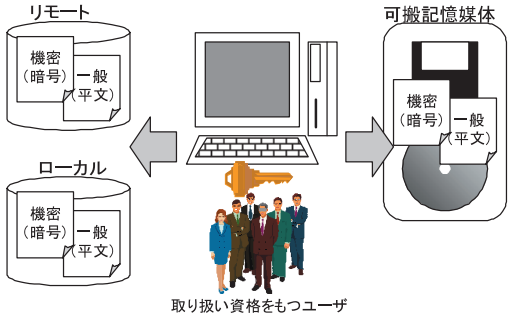


**図 1**　格納ファイルの保存形式
Fig. 1　The format of the stored files.

**(1)　復号された機密情報を平文のまま一般ファイルに保存して持ち出す**
**(2)　復号された機密情報を平文のままインターネットへ送信する**

これらは，ユーザに悪意がなくても起こりうる．特に，(2) はアンチウイルスソフトで検知できない新種のワーム・ウイルスによって発生する恐れもある．そこで本情報フロー制御モデルでは，機密情報が平文のまま流出しないように，プログラムの実行環境を利用目的によって分離し，ファイルアクセスやネットワークアクセスを制限することにした．

### 3.2 プログラム実行環境の分離

提案モデルでは，タイプ A とタイプ B の 2 種類のプログラム実行環境を分離する．

- タイプ A：機密ファイルを閲覧・編集するプログラムや，社内イントラネットを利用してファイル共有サーバなどにアクセスするプログラムの実行環境．本環境で生成するファイルはすべて機密ファイルと見なす．
- タイプ B：インターネットを利用して社内の各拠点に電子メールを配信するプログラムや，社外から情報をオンラインで入手するプログラムの実行環境

また，これら実行環境をまたがったデータ転送を制御することで，ワーム・ウイルスや誤操作により機密情報が平文のまま流出することを防止する．

#### 3.2.1 ユーザの操作をともなうデータ転送

ユーザの操作をともなうデータ転送には，ドラッグ&ドロップによるものと，クリップボードを経由したものがある．ドラッグ&ドロップ操作は，同一画面に表示されている複数のアプリケーションプログラム間でのデータ転送に使える．したがって，上記タイプ A とタイプ B のプログラムが同一画面にあると，タイプ A のプログラムにより表示されている

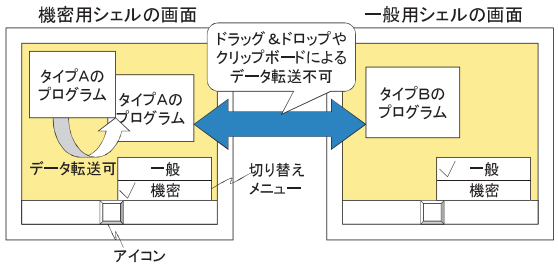


**図 2**　マルチシェル画面のイメージ
Fig. 2　An image of multi-shell screen.

機密情報を，タイプ B のプログラムに転送され，機密情報が平文のまま流出する危険がある．その対策として，本情報フロー制御モデルでは，各アプリケーションプログラムのユーザインタフェースを表示するシェル（デスクトップ）を，上記タイプ A 用とタイプ B 用にそれぞれ生成し，タイプの異なるプログラムのユーザインタフェースが同時に画面表示されないようにする．これによりユーザは，扱う情報のレベルに応じて，機密用のシェルと一般用のシェルを切り替えながら利用することになる．**図 2** は，その画面イメージを示したものである．機密用と一般用の各シェルの画面には，シェル切替え用のアイコンを表示しておき，ユーザが当該アイコンを押下して切替えメニューを選択することで，機密用と一般用のシェルを適宜切替え可能とする．一方，クリップボードとは，ユーザが指定した文字列データや画像データなどを一時的に保持し，他のプログラムのデータ領域へ複写するために使われる共有データエリアのことである．クリップボードを悪用すると，タイプ A のプログラムからタイプ B のプログラムへ機密情報が平文のまま転送できてしまう．そこで本情報フロー制御モデルでは，同一タイプのプログラム間ではクリップボードの共有を許可するが，タイプの異なるプログラム間では機密情報が平文で伝達されないよう制御する．

#### 3.2.2　プロセス間通信を利用したデータ転送

ここでのプロセス間通信とは，たとえば名前付きパイプなどを利用して，プロセスどうしが行うデータ転送を指す．3.2.1 項のデータ転送と異なり，ユーザの操作は介入しない．このようなプロセス間通信がタイプの異なるプログラム間で実行されると，タイプ A のプログラムからタイプ B のプログラムを経由して機密情報が平文のまま流出する恐れがある．そこで本情報フロー制御モデルでは，同一タイプのプログラム間ではプロセス間通信を許可するが，タイプの異なるプログラム間では機密情報が平文で伝達されないよう制御する．

**表 1**　ファイルアクセス権
Table 1　File access control list.

| アクセス対象 | タイプ A のプログラム | タイプ B のプログラム |
|---|---|---|
| 機密ファイル | Read（復号）<br>Write（暗号化） | Read |
| 一般ファイル | Read | Read<br>Write |
| 新規ファイル | Read（復号）<br>Write（暗号化） | Read<br>Write |

Read（復号）：復号を伴う読み出し
Write（暗号化）：暗号化を伴う書き込み

### 3.3　ファイルアクセス制御

機密情報を取り扱うためのプログラム（タイプ A）と，インターネットや一般情報を取り扱うためのプログラム（タイプ B）に，それぞれ**表 1** に示すようなファイル読み出し（Read）と書き込み（Write）権限を与える．

これは，機密ファイルを復号する権限を与えられたタイプ A のプログラムには，一般ファイルへの書き込み権限を与えないことを意味している．また，タイプ A のプログラムによる新規ファイルへの書き込みデータは，3.1 節で述べたように，書き込み先の媒体の種類にかかわらず強制的に暗号化して書き込み，機密ファイルとして保存する．これにより，タイプ A のプログラムの誤操作により，機密情報が平文のままファイルに保存されるといった問題を解決できる．一方，タイプ B のプログラムには機密ファイルを復号する権限を与えないことで，ウイルス感染やユーザの誤操作によって機密情報が平文で流出するリスクを低減させることができる．また，タイプ B のプログラムに，機密ファイルの（暗号文のまま）読み出しを許可する理由は，3.4 節で述べるように，機密ファイルを暗号文のままインターネットなどを介して転送可能とするためである．なお，上記ファイルアクセス制御により制限されないアクセスについては，OS が提供する任意アクセス制御によってアクセス可否を設定することができる．

### 3.4　ネットワークアクセス制御

仮に，機密ファイルを復号して読み出す権限を持つ上記タイプ A のプログラムが通信機能を備えていると，インターネットを通じて平文のまま機密情報が送信されるという問題が

**表 2**　ネットワークアクセス権
Table 2　Network access control list.

| アクセス対象 | タイプ A のプログラム | タイプ B のプログラム |
|---|---|---|
| インターネット | deny | allow |
| イントラネット | allow | deny |

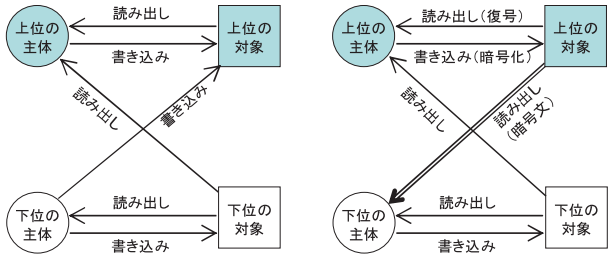


**図 3**　従来モデルとの比較
Fig. 3　Comparison with the conventional model.

生じる．このとき，送信データを暗号化する方法も考えられるが，いったん復号して読み出した機密ファイルを，再度暗号化して送信するのでは効率が悪い．そこで提案モデルでは，**表 2** のようにインターネットの利用をタイプ B のプログラムのみに許可することで，この問題を解決する．たとえば，電子メールソフトをタイプ B と定義しておけば，表 1 に示したように機密ファイルを暗号データとして読み出すことができるため，そのまま添付すれば暗号データの送信となる．これにより，機密情報のネットワーク流出問題と，再暗号化処理のオーバヘッド問題を解決できる．

### 3.5　従来モデルとの比較

従来から機密性を重視したセキュリティポリシモデルとして，High-Water Mark [13] モデルや Bell-LaPadula [14] モデルのような階層型のほか，Chinese Wall [15] のような区分型や束型などがある．提案モデルは階層型に最も近いが，**図 3** に示すように，階層間の情報フローが双方向である点が従来型と大きく異なる．

ただし，下位の主体が上位の対象から読み出すデータは暗号化した状態であり，機密情報を平文で読み出すことはできない．Bell-LaPadula モデルの 2 つ特性のうちシンプル・セキュリティ特性と比較すると，「主体は，対象が同じまたは下位の階層にあり，かつ対象に対して任意読み出しアクセス権を持つならば，対象を読み出すことができる」という点は共通であるが，提案モデルではさらに，「主体は，対象が上位の階層にあり，かつ対象に対して任意読み出しアクセス権を持つならば，対象を暗号文のまま読み出すことができる」も特性に加わることになる．このような特性を持つモデルにより，機密情報を上位の主体どうしの直接的な交換だけでなく，下位の主体を仲介役とした間接的なデータ交換も，暗号化により安全に行えるようになる．つまり，これまで情報漏洩の主な経路であったインターネット，可搬記録媒体，紙（印刷物）を用いても機密情報を上位の主体（タイプ A のプログラム）間で安全に伝達できるようになる．もう 1 つの「主体は，対象が同じまたは上位の階層にあり，かつ対象に対して書き込みアクセス権を持つならば，対象に書き込むことができる」という ∗（スター）特性に対して，提案モデルは「主体は，対象が同じ階層にあり，かつ対象に対して書き込みアクセス権を持つならば，対象に書き込むことができる」という特性を持つ．これと類似した特性を持つものとして Biba モデル[16] があるが，これは下位の主体が上位の対象を不正に変更できないよう，完全性を保つための特性であり，ワーム・ウイルスなどから機密情報の破壊を防止する意味がある．

## 4.　実現の一形態

上記情報フロー制御モデルを実現するうえでの機能要件とその実現の一形態を示し，提案モデルの実現可能性を述べる．

### 4.1　機能要件と実現方針

提案モデルを実現する際に満たすべき機能要件を示す．

- 要件 1：プログラムの実行環境を目的別に複数作成する機能
- 要件 2：実行環境をまたがったプロセス間通信とクリップボード経由のデータ転送を制御する機能
- 要件 3：実行環境に応じてネットワークアクセスを表 2 の保護ポリシに従い制御する機能
- 要件 4：実行環境に応じてファイルアクセスを表 1 の保護ポリシに従い制御する機能
- 要件 5：ファイルを復号する鍵を，関係者以外から不正に取得できないよう管理する機能

以下，上記要件をどのような手段を用いて満たすかを記す．要件 1 については，4.3 節で

述べる WindowBox[10] と同様に，Windows®,⋆1 OS をベースにプログラム実行環境をデスクトップ単位で複数構築する方式を採用する．要件 2 のうち，デスクトップをまたがるプロセス間通信の制御については WindowBox[10] と同様の方式を採用するが，クリップボード経由のデータ転送制御については WindowBox[10] で示されていないため，4.3.2 項において新たに制御方法を示す．また，要件 3 については WindowBox[10] と同等の方式で実現し，要件 4 については，4.3.1 項で述べるファイルアクセス制御モジュールとファイル暗号モジュールにより実現する．要件 5 については 4.2 節で述べるように，グループ暗号システム[17),18)] を採用するが，社内だけでなく社外も含めて関係者（共有メンバ）のみが機密情報の復号に必要なグループ鍵を取得可能とする鍵配布方式については 4.2.3 項において新たに示す．

### 4.2　機密情報の暗号化

紛失・置忘れ，盗難，誤送信，ワーム・ウイルスの問題に備え，提案モデルでは機密情報をつねに暗号化して保存する．そしてその復号用の鍵は，機密情報の取扱い資格を有するメンバで安全に共有する必要がある．そのような鍵の共有は，たとえばグループ暗号システム[17),18)] により実現できる．

#### 4.2.1　グループ暗号システムの概要

グループ暗号システム[17),18)] では，**図 4** に示すように，宛先リストとシステム固有のマスタ鍵とから生成したグループ鍵を用いて情報を暗号化する．

宛先リストとは，情報の開示先となるユーザ名およびグループ名のリストであり，所属や役職などを組み合わせて柔軟に指定できる．グループ鍵生成に用いた宛先リストは，暗号化した情報のヘッダに付加する．復号処理時には，鍵生成装置で権限チェックプログラムが暗号化情報のヘッダから宛先リストを取り出し，復号を試みるユーザの ID 情報が宛先リストに含まれるか否かを確認する．このとき，含まれていればグループ鍵生成プログラムがグループ鍵を生成し，含まれなければ生成しない．復号処理時に生成するグループ鍵は，暗号化に用いたグループ鍵と同一である．

#### 4.2.2　機密ファイルの宛先リスト

グループ暗号システムのユーザには，認証に用いるユーザ識別子とパスワードを発行するとともに，ユーザ識別子には，**表 3** に示すような ID 情報を関連付けて管理する．ユーザ識別子とパスワードを用いて，鍵生成装置に対する認証が成立すると，グループ鍵を生成可能となる．グループ鍵の生成に用いる宛先リストは，ID 情報に含まれる各カテゴリを条件式で連結した形で表現する．たとえば，役職（行番号 6 の Category の値）が主任（Code の値が 9）以上全員で共有したい場合には，宛先リストを 6C ≧ 9（C は Code の頭文字で，行番号 6 の Category の Code の値が 9 より大きいか等しい）と指定する．

提案モデルの実現においては，機密ファイルを社外秘情報と見なせば，組織内の全員で共有可能なグループ鍵が必要となる．そのような鍵は，すべての役職に対して 1 以上のいずれかのコードを割り当てておき，宛先リストを 6C > 0 と指定することで生成できる．

さらに本論文では，グループ鍵を社外関係者と共有するために，上記宛先リストを社内の

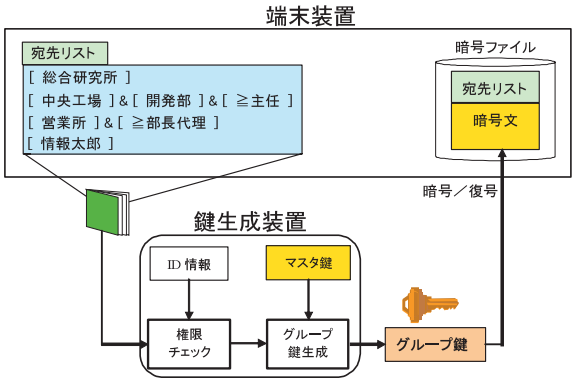


**図 4**　グループ暗号システムの概要
Fig. 4　Outline of group cipher system.

**表 3**　ユーザの ID 情報
Table 3　User ID information.

| No. | Category | Data | Code |
|---|---|---|---|
| 1 | 氏名 | 情報 太郎 | |
| 2 | 生年月日 | 1960/11/07 | 19601107 |
| 3 | 性別 | 男性 | M |
| 4 | 事業所 | システム開発本部 | 301 |
| 5 | 部 | 製品企画部 | 3 |
| 6 | 役職 | 主任 | 9 |

⋆1 Windows は，米国 Microsoft Corporation の，米国およびその他の国における登録商標または商標です．

ID 情報だけでなく社外関係者の ID 情報も組み合わせて指定する方法を新たに提案する．社外関係者の ID 情報は，所属組織名と氏名とメールアドレスから構成し，社外関係者を氏名で選択すると，メールアドレスを宛先リストに含めることとした．たとえば，組織内の全員と社外関係者である情報次郎（メールアドレスは joho.jiro@iisec.com）さんとで共有するグループ鍵は，前記 $6C > 0$ と joho.jiro@iisec.com をカンマ「,」でつなげた宛先リストから生成する．ここで，社外関係者の識別情報としてメールアドレスを用いる理由は，所属する組織にかかわらず大多数の人が所有する一意な識別情報であり，利用頻度が高く，かつ更新頻度が低いことにある．また，宛先リストに含まれるメールアドレスは，機密情報の復号に必要なグループ鍵の配布にも利用する．なお，所属組織名と氏名は，社外関係者のメールアドレスを一覧から選択しやすくするために用いる情報であり，宛先リストには含めない．

### 4.2.3　グループ鍵生成サーバの設置

ファイル暗号製品には，本人確認用のパスワードとファイル復号用の鍵を兼用しているものが多いが，一般にパスワードは人間が覚えやすい（忘れにくい）ものを用いる傾向があり，そのため推測されやすいともいえる．これに対してグループ鍵は，前述のように宛先リストとマスタ鍵とから生成するものであり，ユーザ認証（本人確認）に用いるパスワードとは明確に区別している．つまり，パスワードに比べてグループ鍵は推測しにくいという利点があり，社外の第三者にパスワードを盗まれてもグループ鍵を渡さなければ情報漏洩リスクを低減させることができる．そのための，グループ鍵の配布方法について説明する．グループ暗号システムには，グループ鍵を IC カードで生成する方式[17] とサーバで生成する方式[18] とがあるが，パスワードの推測によって不当に開示されるという問題を考えた場合，情報と同じく紛失・置忘れ，盗難の可能性のある IC カードでは対策できないケースもある．そこで，提案モデルの実現においてはサーバ方式を採用し，パスワード推測によるグループ鍵の不正取得を防止する．具体的には，社内のみで機密情報を共有する場合，グループ鍵生成サーバを社外（関係者以外）からはアクセスできない領域（例：イントラネット）に設置することで，グループ鍵の不正取得を防止する．また，社外関係者と機密情報を共有する場合，社外関係者がインターネット経由でアクセス可能な領域にもグループ鍵生成サーバを設置する．そして，社外関係者へのグループ鍵配布は，ユーザ識別と権限チェックの後に，宛先リストに含まれるメールアドレス宛に送付する．これにより，関係者以外によるグループ鍵の不正取得を防止する．ユーザ識別からグループ鍵配布までの処理シーケンスを**図 5** に示す．

① 暗号ファイル登録者による操作：社外関係者と共有する機密情報を暗号ファイルとして登録し，その URL を社外関係者にメールなどで通知する．

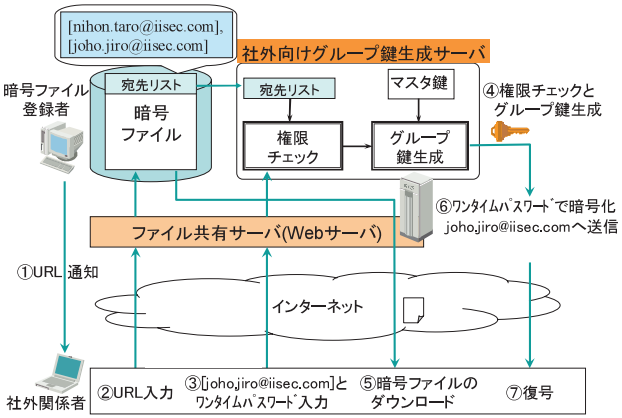


**図 5**　グループ鍵配布の処理シーケンス
Fig. 5　Process sequence of group key distribution.

② ファイル共有サーバの処理：上記 URL へのアクセスに対して，ユーザ ID とワンタイムパスワードを入力するページを表示する．

③ 社外関係者による操作：上記ページにおいて，ユーザ ID として自身のメールアドレスを，ワンタイムパスワードとして任意の文字列を入力する．

④ グループ鍵生成サーバの処理：上記ページから入力したユーザ ID（メールアドレス）が，上記 URL の暗号ファイルの宛先リストに含まれるか否かをチェックし，含まれていればグループ鍵を生成する．

⑤ ファイル共有サーバの処理：上記暗号ファイルの宛先に含まれていれば，暗号ファイルのダウンロードを許可する．含まれていなければ社外関係者にエラーを返して処理を終了する．

⑥ ファイル共有サーバの処理：上記ダウンロードに続く処理として，生成したグループ鍵を上記ワンタイムパスワードで暗号化し，上記ユーザ ID であるメールアドレスに送信する．

⑦ 社外関係者による操作：メールで受信したグループ鍵とワンタイムパスワードを用いて，先にダウンロードした暗号ファイルを復号する．なお，ファイル復号用のプログラムは，ファイル共有サーバからダウンロード可能とする．

以上述べたグループ鍵配布の処理は，社外関係者の認証処理を含まないが，グループ鍵の配布先は宛先リストに含まれるメールアドレスとすることで，関係者以外にグループ鍵を配

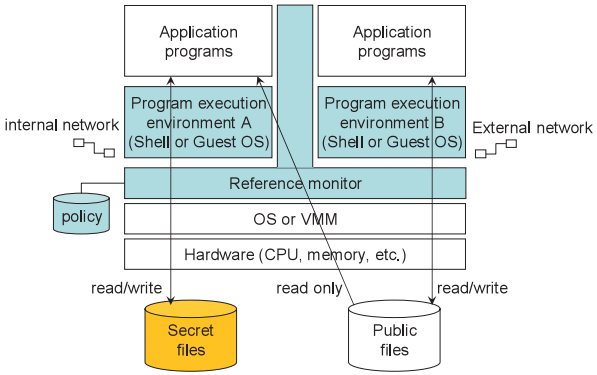


**図 6**　サンドボックスの概念図
Fig. 6　Conceptual diagram of sandbox.

信することはない．また，グループ鍵をワンタイムパスワードで暗号化して送信することでメールの盗聴にも対処できる．任意のワンタイムパスワードを用いることは，あらかじめパスワードを登録・管理する手間が省けるという利点にもつながる．そのほか，ファイル共有サーバにて，各暗号ファイルに対するグループ鍵配布の回数や期限を管理して，それらを越えたアクセスを拒否することも有効である．

### 4.3　プログラム実行環境の分離

機密情報を扱うプログラムと，一般情報やインターネットを利用するプログラムの実行環境を分離することで機密情報を保護するものとして，たとえばセキュア OS [7)–9)] のほかに，WindowBox [10)] や SVFS [11)] のようなサンドボックスと呼ばれる技術がある．クライアント PC での実装方式として，筆者らは上記セキュア OS [7)–9)] がベースとする Linux [⋆1] よりもシェアの高い Windows® OS で実現可能なサンドボックスを選択した．その種類は様々あるが，ここでのサンドボックスとは，プログラムの実行環境を複数作成する機能と，実行環境に応じてプログラムからのファイルアクセスやネットワークアクセスを制限する機能と，実行環境をまたがってプログラム間で行われるデータ転送を制限する機能とを備えたものと定義する．**図 6** は，サンドボックスの概念を示したものである．

サンドボックスモデルの 1 つである WindowBox [10)] は，Windows® OS の上に複数のデスクトップ（シェル）を生成し，たとえば機密情報を扱うプログラムの実行環境と，インターネットを利用するプログラムの実行環境とを，デスクトップ単位で構築して分離するものである．一方の SVFS [11)] は，VMM（Virtual Machine Monitor）の上に複数のゲスト OS （Windows® OS など）を実行し，複数のプログラム実行環境をゲスト OS 単位で構築し分離するものである．既存のクライアント PC への導入を考えると，SVFS は OS から再インストールする必要があるのに対して，WindowBox は，すでに導入済みの OS に，新たな（機密用または一般用の）デスクトップを起動するモジュールや，ファイルアクセス制御モジュールなどをインストールするだけでよく，OS の再インストール作業に比べて容易である．そこで提案モデルの実現手段としては，WindowBox のようにデスクトップ単位でプログラム実行環境の分離を行う方式を選択する．

次に，上記サンドボックスのアクセス制御ポリシに着目すると，ネットワークアクセス制御については提案モデルと共通である．しかしファイルアクセス制御については，先に述べた Bell-LaPadula モデルと同様にタイプ B のプログラムから機密情報を読み出すことをいっさい禁止（図 6 参照）しているため，提案モデルの実現においては，4.3.1 項に示すようなファイルアクセス制御機構に置き換える必要がある．また，WindowBox [10)] は，1 つの汎用 OS 上に複数のデスクトップを生成してプログラムの実行環境を分離するものであるため，デスクトップをまたがってクリップボードを共有可能な OS の場合，タイプ A からタイプ B のプログラムに不正なデータ転送が発生する可能性があるが，その制御については記述されていない．この制御機構については 4.3.2 項で新たに述べる．

#### 4.3.1　ファイルアクセス制御

**図 7** に示すファイルアクセス制御モジュールにより，ファイルシステムを経由するファイルアクセスをすべて監視する．このとき，アクセス要求元となるプログラムのタイプ（A または B）と，アクセス対象となるファイルの種別（機密ファイルまたは一般ファイル）を判別し，表 1 のファイルアクセス権に従い制御する．プログラムのタイプの判別は，そのプログラムが，いずれのデスクトップから起動したものかを確認することで可能となる．具体的には，そのプログラムに割り当てられたユーザ ID（権限）から判別してもよいし，あるいはプログラムの起動のたびに，いずれのデスクトップからの起動であったかを記憶するテーブルを設けて参照する方法もある．また，プログラムの中には OS の起動と同時に実行を開始し，いずれのデスクトップにも属さないものもある．Windows®では，OS 権限を持つ Service と呼ばれるプログラムがそれに該当する．ファイルアクセス制御では，それらの Service をすべてタイプ B と同等に扱うこととし，表 1 のファイルアクセス権に従い，機密

⋆1 Linux は，Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です．

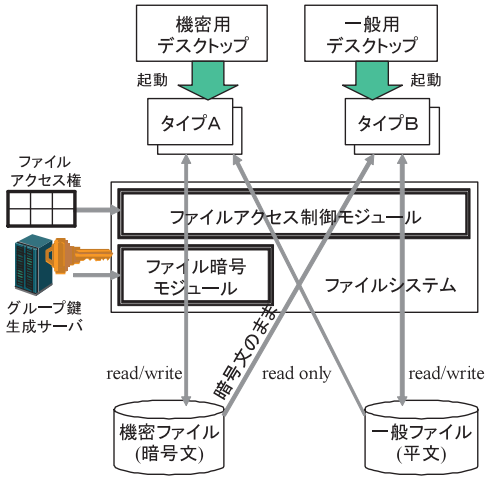


**図 7**　ファイルアクセス制御
Fig. 7　File access control.

ファイルに対する write や復号をともなう read を禁止する．これは，Administrator 権限を持つプログラムに対しても同様とする．一方，アクセス対象となるファイルの種別については，暗号化された機密ファイルの先頭部分に特定のコードを埋めておき，当該コードの有無から機密ファイルか一般ファイルかを判別する．また，上記ファイルアクセス権に従い，正当と判定されたファイルアクセスが暗号化または復号をともなう書き込みや読み出しであれば，ファイルアクセス制御モジュールからファイル暗号モジュールを呼び出す．ファイル暗号モジュールは，グループ鍵生成サーバから権限に応じて配布されるグループ鍵を用いて機密ファイルの暗号化・復号処理を実行する．ただし暗号化の対象はファイルの内容のみとし，ファイル名やフォルダ名は平文とする．これは，機密用と一般用のどちらのデスクトップからも同じファイル名として見えるようにするためである．これにより，機微な情報をファイル名に付けてしまうと一般環境からも見えてしまうという問題があるが，これについては，特定の事物もしくは人物を指すような機微な情報にはコードネームを割り当ててユーザに使用させるといったガイドラインの作成や教育により対処できる．あるいは，ガイドラインを参照しながらファイル名に含まれる機微な情報を自動的に所定のコードネームに変換する機能を備えてもよい．

以上のようなファイルアクセス制御モジュールやファイル暗号モジュールは，当然ながら一般的な OS が備えるファイルアクセス制御とは異質なものであるが，たとえばフィルタドライバ[19] を用いてファイルシステムの機能を追加する形で実現可能と考える．

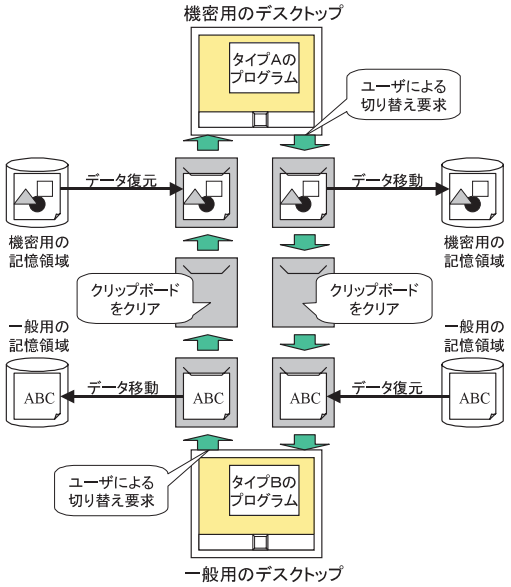


**図 8**　クリップボード経由のデータ転送制御
Fig. 8　Data transfer control through clipboard.

### 4.3.2　クリップボード経由のデータ転送制御

デスクトップをまたがったクリップボードの共有を防止するクリップボード制御モジュールについて述べる．クリップボード制御モジュールは，OS が起動した直後に実行開始し，OS が終了するまで常駐するプログラムとして実装する．本プログラムにより，ユーザのログインによってデスクトップを生成するたびに，クリップボード内のデータを一時的に保持するための記憶領域をメモリ上に作成する．そして，ユーザによるデスクトップの切替えのタイミングで，クリップボード内のデータを上記記憶領域に移動し，切替え先のデスクトップで使用していたデータと入れ替える（**図 8**）．これは，同一デスクトップ内でのクリップボードの利用を制限するものではない．なお，上記メモリ上で一時的に保持している平文の機密データを，タイプ B のプログラムから不正に参照されないよう保護するのはサンドボックスの役割であり，その強度は利用するサンドボックスの仕様に依存する．

上記のようにタイプ A とタイプ B をまたがったクリップボードの共有を禁止したり，WindowBox のようにプロセス間通信を禁止したりする以外に，それらのデータ転送を許可しながらも機密情報の流出を防止する方法もある．たとえば，クリップボードへのアクセスやプロセス間通信をつねに監視するソフトウェアモジュールにより，タイプ A のプログラムからの書き込みデータや通信データをグループ鍵で強制的に暗号化することもその 1 つである．そして，これらのデータを受け取るタイプ B のプログラムには，機密ファイルに対する read 権限（表 1 参照）と同様に，暗号文のまま読み出すことを許可する．これにより，機密情報が平文のまま流出することを防ぎつつ，インターネット（タイプ B のプログラム）を介してタイプ A のプログラムどうしが機密情報を交換可能となる．

## 5. 適用システムの構成と利用イメージ

提案の情報フロー制御モデルを適用した情報システムの構成例とその利用イメージを説明する．

### 5.1 システム構成例

提案システム特有のコンポーネントは，グループ鍵生成サーバと，サンドボックス技術[10]によりプログラム実行環境を二分したクライアント PC であり，これらを一般的な企業情報システム（イントラネット）に設置する．複数の拠点を有する企業の場合，拠点ごとに社内用のグループ鍵生成サーバを設置してもよい．ただし，マスタ鍵は共通とする．社外用のグループ鍵生成サーバについては，図 5 に示したファイル共有サーバや公開 Web サーバ，メール中継サーバなどとともに DMZ（De-Militarized Zone）に設置する．また，クライアント PC が備える 2 種類のプログラム実行環境のうち，機密用のプログラム実行環境（タイプ A）には，機密情報の閲覧や編集に用いるプログラムのほか，社内のグループ鍵生成サーバやファイル共有サーバにアクセスするためのプログラムをインストールして利用する．また，インターネットを利用するためのプログラム実行環境（タイプ B）には，ブラウザや電子メールソフトをインストールして利用する．また，情報共有に先立ち，不特定の人がインターネットを通じてダウンロード可能なファイルを除いて，業務に用いる情報を含むファイルはすべて機密ファイルと見なし，暗号化してファイル共有サーバやクライアント PC に格納する．このとき，社内で共有する機密ファイルの宛先リストは，4.2.2 項で述べたように「$6C > 0$」（すべての役職で共有可能）を既定値とするが，情報の機密レベルに応じて共有範囲をさらに限定することも可能である．一方，社外関係者と共有する機密ファイルについては，当該社外関係者のメールアドレスを宛先リストに含めて暗号化する．なお，

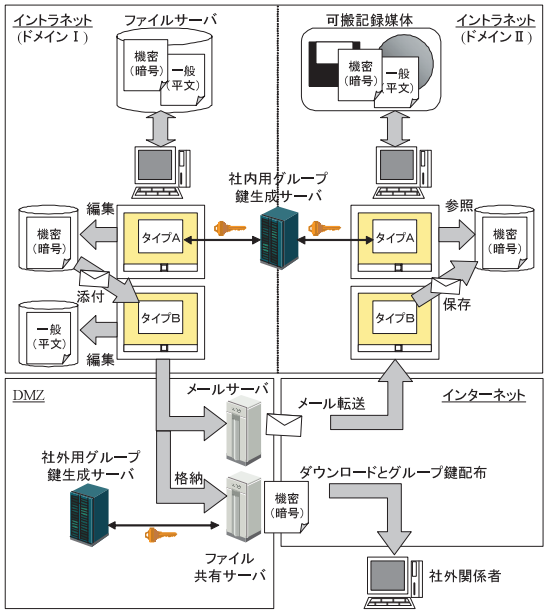


**図 9**　機密ファイルの送受信
Fig. 9　Transmission and receipt of sensitive files.

グループ暗号システム[17),18)]では手動によるファイル暗号化だけでなく，宛先リストが設定されたフォルダにコピーするだけで自動的に暗号化する機能も提案している．以上により，クライアント PC（モバイル用も含む）を紛失したり置き忘れたり，第三者により盗難されたりしても，機密情報はつねに暗号ファイルとして格納されており，機密情報が関係者以外（宛先リストに含まれない人物）に開示されるリスクを低減させることができる．

### 5.2 機密ファイルの送受信

機密ファイルの編集とその送信から受信までの流れを**図 9** に示す．機密ファイルの編集は，機密用のプログラム実行環境（タイプ A）で行い，暗号化して保存する．機密ファイルを電子メールに添付して社内の各拠点で共有する際には，一般用のプログラム実行環境からも任意アクセス可能な共有フォルダに保存する．その後，一般用のプログラム実行環境（タイプ B）に切り替えてから電子メールソフトを用いて，送信先メールアドレス，件名，本文を入力するとともに，先に保存した機密ファイルを添付して送信する．このとき添付した機

密ファイルは，暗号文のまま読み出され送信される．電子メールの受信側は，タイプ B のプログラム実行環境で電子メールソフトを利用して受信し，添付された機密ファイルを，機密用のプログラム実行環境からも参照可能な共有フォルダに保存する．このとき，受信した機密ファイルは暗号化されているため，暗号ファイルとして保存される．暗号ファイルを復号して閲覧するときは，タイプ A のプログラム実行環境に切り替え，閲覧用のプログラムを起動して暗号ファイルを開くと，図 7 で示したファイル暗号モジュールが，暗号ファイルに付加された宛先リストを社内用グループ鍵生成サーバに送付し，復号に用いるグループ鍵を要求する．社内用グループ鍵生成サーバは，ユーザの認証（本人確認）と権限チェックを実行し，宛先リストに含まれる場合に限りグループ鍵を生成して返信する．これにより，宛先リストに含まれる（権限のある）ユーザだけが受信ファイルを復号して情報共有が可能となり，仮に誤って機密ファイルを社外に送信した場合でも情報漏洩リスクを低減させることができる．

社外関係者と機密ファイルを共有する場合は，図 5 に示したように，社外関係者のメールアドレスを含む宛先リストから生成したグループ鍵で機密ファイルを暗号化してからファイル共有サーバに格納する．このとき，ワークフローシステムなどを利用して，宛先リストや機密ファイルの指定に誤りがないことを上長により確認する．仮に，ファイル共有サーバに格納して社外関係者に URL を通知した後に誤りが発覚した場合は，速やかに暗号ファイルを削除するか，グループ鍵を無効にして被害を最小限にとどめる．暗号ファイルの URL を受信した社外関係者は，社外用グループ鍵生成サーバからグループ鍵を取得して機密情報を復号する．ただし，社外関係者が利用するクライアント PC では，サンドボックス技術によるプログラム実行環境の分離はなくてもよい．なお，社外関係者による二次的な情報流出を防ぐためには，既存の DRM（Digital Rights Management）製品の機能を用いて，ファイルの複製や印刷を防止することが有効である．

### 5.3　可搬記録媒体によるファイルコピー

前節で述べたオンラインによるファイル共有手段は，社内および社外関係者との共有に利用できるが，社内においては，PC 間のファイルコピーや簡易なバックアップ手段として可搬記録媒体を利用する場面はある．そこで本節では，社内での利用を想定して可搬記録媒体によるファイルコピーについて説明する．

提案モデルでは，タイプ A のプログラム実行環境から保存するファイルは，その格納先媒体の種別を問わずつねに暗号化して保存される．したがって，USB メモリなどの可搬記録媒体に機密ファイルを格納した場合でも，図 7 で示したファイル暗号モジュールにより自動的に暗号化して保存することになる．一方，タイプ B のプログラム実行環境から機密ファイルを可搬記録媒体にコピーする場合も，表 1 のファイルアクセス権によれば，機密ファイルを暗号文のまま read し，新規ファイルに暗号文のまま write することになるため，コピー先のファイルも暗号ファイルとなる．つまり，可搬記録媒体に保存する機密ファイルは，プログラム実行環境に依存せずつねに暗号ファイルとなる．これにより，可搬記録媒体を紛失したり置き忘れたり，第三者により盗難されたりしても，機密情報の漏洩リスクを低減させることができる．

機密ファイルの受け取り側は，可搬記録媒体から直接ファイルを開いて参照する場合と，クライアント PC にコピーしたファイルを参照する場合とがある．いずれにしても，権限のある（宛先リストに含まれる）ユーザが，タイプ A のプログラム実行環境から開いた場合に限り，復号して read 可能となる．また，可搬記録媒体からクライアント PC への機密ファイルのコピーは，タイプ A，B いずれの環境からも可能であり，コピー先のファイルはつねに暗号ファイルとなる．厳密にいえば，タイプ A のプログラムを用いて機密ファイルをコピーすると，read で復号処理が実行され，write 時に再暗号化によって暗号ファイルが生成されることになるが，タイプ B のプログラムを用いれば再暗号化処理は発生せず，暗号ファイルのままコピーされるという違いがある．

## 6. 有用性の検討

以上述べた情報フロー制御モデルによれば，業務で作成したファイルはすべて機密ファイルと見なして保護するため，フィンガープリント技術[3] のように，機密情報のシグニチャを算出してリスト化する手間は不要となる．機密情報の読み出しについては，デジタル著作権管理システム[4] に比べてアプリケーションプログラムに依拠しない仕組みとなっている．また，機密情報をグループ鍵で自動的に暗号化することで，事前に関係者全員に鍵を配布しなくても，インターネットや可搬記録媒体を用いて安全に情報共有が可能となり，暗号化を忘れる心配もない．さらに，表 1 と表 2 に示したように，アクセス権（保護ポリシ）はプログラム実行環境ごとに固定であり，アプリケーションプログラムごとに変更する必要はない．

日常業務における使い勝手の点でいうと，これまでインターネット用と内部ネットワーク用とで PC を使い分けてきたオフィスであれば，それが 1 台でまかなえるため，利便性は大きく向上すると考える．これに対して 1 台の PC を多目的に利用してきたオフィスの場合，使用するプログラムによってデスクトップの切替え操作をユーザに要求することになるが，たとえばワンクリック操作で切替え可能とすることで負荷を軽減できる．また，機密ファイ

ルの暗号化については，たとえば社外秘（宛先リスト「6C > 0」）を規定値として自動暗号化することで，ユーザによる特別な操作は不要となる．つまり，デスクトップ切替え操作は必要となるが，手動によるファイル暗号化操作を強いるよりも使い勝手は向上するものと考える．実現方式については，情報を機密と一般の 2 つに区分することを前提に素案を示した．社内で共有する機密情報を，極秘と秘のようにさらにレベル分けする場合でも，一般も含めてレベルの数だけデスクトップを作成し，グループ鍵もレベルごとに宛先リストを変えて使い分ければ，3 つ以上の区分にも対応可能と考える．ただし，社外関係者と共有する情報については，対象のファイルやフォルダごとに適切な宛先リストを指定する手間がかかる．また，機密ファイルから機微な情報を削除したものを，一般ファイルとして平文で社外に発信する際の例外機構については記述していない．これについては，たとえばファイルの格下げの申請と承認のプロセスをワークフローシステムにより構築し，上長の承認を得たファイルを，平文ファイルとして申請者の一般用デスクトップから受信可能とする機構が別途必要である．機密ファイルの印刷についても同様に，原則は印刷禁止としつつ，上長の承認を得たものに限り印刷を許可する機構を設けてもよい．あるいは紙の暗号化技術[20]や，印刷先のプリンタを制限する機構との組合せも考えて，業務への影響が少ない手段を選べるようにしたい．

セキュリティ面で考えられる問題点としては，インターネットを利用するためのプログラムが受信した不審なファイルを，機密情報を扱うためのプログラムが読み出したときにワーム・ウイルスに感染する可能性が考えられる．提案モデルに基づくアクセス制御が機能する限りは情報漏洩につながらないが，そのような悪意のあるプログラムがファイルアクセス制御モジュールに対して仕掛ける攻撃として，

(a) ファイルアクセス制御モジュールの強制終了またはアンインストールによる保護機能の無効化

(b) 悪意のあるプログラムのインストールによりタイプ A 側で復号された機密情報の横取り

(c) ファイルやネットワークのアクセス権を改ざんし，機密情報を平文で読み出す権限とインターネットを利用する権限を同時に取得

が考えられる．(a) については，強制終了時やアンインストール時に Administrator 権限やパスワード入力を要求することで対策できる．(b) については，ユーザモードで実行するプログラムと，カーネルモードで実行するプログラムとで対策が異なる．ユーザモードで実行するプログラムであれば，タイプ A 側で機密情報を読み取られても，タイプ A のプログラムにはインターネットを利用する権限を与えていないため，直接情報漏洩につながるわけではない．一方，カーネルモードのプログラムを不正にインストールする攻撃や (c) のような攻撃への耐性は持たないことから，OS やアプリケーションプログラムの改ざん検知機能と組み合わせる必要がある．そのような機能の例として，最新の Windows® OS が備える UAC（User Account Control）のような，OS の重要な設定変更を検出してユーザに通知・ブロックする機能のほか，TPM（Trusted Platform Module）[21] を利用した PC 構成情報の改ざん検知がある．そのほか，完全性や可用性を侵害する攻撃として，たとえばタイプ A 側のプログラムがウイルスに感染して機密ファイルを破壊するという問題や，グループ鍵生成サーバへの DoS（Denial of Service）攻撃などが考えられる．これらについては，情報漏洩対策を目的とした提案モデルとはまったく別の対策手段を講じる必要がある．グループ暗号システム特有のセキュリティ問題としては，特に社外用グループ鍵生成サーバにおけるマスタ鍵への攻撃があげられるが，この問題には，Hardware Security Module（HSM）を用いたマスタ鍵の保管とグループ鍵生成により対策できる．

## 7. おわりに

本論文では，社外関係者も含めた機密情報の安全な共有を可能とする情報フロー制御モデルを提案し，従来モデルとの差異とその効果について述べるとともに，実現の一形態と利用イメージを示しながら有用性と問題点について述べた．提案モデルによれば，可搬記録媒体やモバイル PC を紛失したり置き忘れたり，第三者により盗難されたりしても，機密情報が関係者以外（宛先リストに含まれない人物）に開示されるリスクを低減させることができる．同様に，電子メールの誤送信やワーム・ウイルスにより，機密ファイルが関係者以外の手に渡るという問題に対しても有効である．

今後は，アクセス制御モジュールに対する攻撃への耐性も含めて，提案モデルの実現方式の詳細を策定する予定である．また，社外関係者との機密情報共有において，送信だけでなく受信する機密情報の保護にも有用なファイル共有システムについても検討していく．

## 参考文献

1) JNSA セキュリティ被害調査ワーキンググループ：2007 年度情報セキュリティインシデントに関する調査報告書 Ver.1.6，NPO 日本ネットワークセキュリティ協会（オンライン）．入手先 http://www.jnsa.org/result/2007/pol/incident/index.html (参照 2009-11-11)

2) 田村　卓，神田昌彦：NEC グループにおける IT による情報漏えい防止の取り組み—情報漏えい防御システム ARGUS，NEC 技報（オンライン）．入手先

http://www.nec.co.jp/techrep/ja/journal/g07/n01/070110.html (参照 2009-11-11)
3) Brin, S., Davis, J. and Molina, G.H.: Copy detection mechanisms for digital documents, *Proc. ACM SIGMOD international conference on Management of data*, pp.398–409 (1995).
4) Ku, W. and Chi, C.H.: Survey on the Technological Aspects of Digital Rights Management, *Proc. International Conference on Information Security* (*ISC*), Vol.3225, pp.391–403 (2004).
5) 栗田弘之，塩谷亮太，入江英嗣ほか：動的なインフォメーションフロー制御による情報漏洩防止手法，情報処理学会報告 2007-ARC-172（HOKKE2007），Vol.2007, No.17, pp.227–232 (2007).
6) DoD 5200.28-STD, TCSEC: Department of Defense Trusted Computer System Evaluation Criteria, National Computer Security Center (Dec. 1985).
7) Loscocco, P. and Smalley, S.: Integrating Flexible Support for Security Policies into the Linux Operating System, *Proc. FREENIX Track: 2001 USENIX Annual Technical Conference* (2001).
8) Bauer, M.: Paranoid Penguin: An Introduction to Novell AppArmor, *Linux Journal*, Vol.2006, No.148, p.13 (2006).
9) Hallyn, S. and Kearns, P.: Domain Type Enforcement for Linux, *Proc. 4th Annual Linux Showcase and Conference* (2000).
10) Balfanz, D. and Simon, D.: WindowBox: A Simple Security Model for the Connected Desktop, *Proc. 4th USENIX Windows Systems Symposium* (Aug. 2000).
11) Zhao, X., Borders, K. and Prakash, A.: Towards protecting sensitive files in a compromised system, *Proc. 3rd IEEE International Security in Storage Workshop* (*SISW'05*), pp.21–28 (2005).
12) 三輪吉和，宮田　仁：持ち出し禁止データを外部から安全に利用する一方法，第 1 回インターネットと運用技術シンポジウム IOTS2008 (Dec. 2008).
13) Weissman, C.: Security Controls in the ADEPT-50 Time Sharing System, *Proc. AFIPS Conference*, Vol.35, pp.119–133 (1969).
14) Bell, D.E. and LaPadula, L.J.: Secure Computer Systems: Mathematical Foundations, Technical Report MTR-2547, MITRE Corporation (1973).
15) Brewer, D. and Nash, M.: The Chinese Wall Security Policy, *Proc. IEEE Symposium on Security and Privacy*, pp.206–214 (1989).
16) Biba, K.J.: Integrity Considerations for Secure Computer System, Technical Report MTR-3153, MITRE Corporation (1975).
17) Ito, H., Susaki, S., Arai, M., et al.: Group Cipher System for Intranet Security, *Trans. IEICE*, Vol.E81-A, No.1, pp.28–34 (1998).
18) 荒井正人，鍛　忠司，伊藤浩道ほか：企業情報向けグループ暗号システム，情報処理学会論文誌，Vol.40, No.12, pp.4378–4387 (1999).
19) Microsoft Corporation: Filter Driver Development Guide Version 1.0a, Microsoft (online). available from http://download.microsoft.com/download/e/b/a/eba1050f-a31d-436b-9281-92cdfeae4b45/FilterDriverDeveloperGuide.doc (accessed 2009-11-11)
20) 富士通：印刷後に復号可能な，世界初の「紙の暗号化技術」，富士通ジャーナル（オンライン）．入手先 http://jp.fujitsu.com/about/journal/technology/20081104/ (参照 2009-11-11)
21) 中村智久，東川淳紀：解説 PC 搭載セキュリティチップ（TPM）の概要と最新動向，情報処理学会誌，Vol.47, No.5 (2006).

(平成 21 年 1 月 30 日受付)
(平成 21 年 11 月 6 日採録)

**荒井　正人**（正会員）

1992 年日本大学大学院理工学研究科博士前期課程修了．同年（株）日立製作所入社．システム開発研究所にてネットワークシステム，セキュリティ技術等の研究開発に従事し，製品化に貢献．現在，研究開発本部研究戦略統括センタ所属．2009 年情報セキュリティ大学院大学情報セキュリティ研究科博士後期課程修了．博士（情報学）．

**田中　英彦**（正会員）

1970 年東京大学大学院工学系研究科電気工学専門課程修了，工学博士．東京大学にて計算機アーキテクチャ，並列処理，人工知能，自然言語処理，分散処理，メディア処理等の教育・研究に従事．東京大学工学部教授，同大学院情報理工学系研究科長を経て，2004 年情報セキュリティ大学院大学情報セキュリティ研究科長・教授．情報処理学会名誉員，人工知能学会論文賞，ACM SIGGRAPH'99 Impact Paper Award，人工知能学会功績賞，東京都科学技術功労者表彰，経済産業大臣表彰等受賞．情報・システム研究機構教育研究評議会評議員，日本学術会議会員，IEEE Fellow，東京大学名誉教授．