社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－DPS－134 (17)
2008－CSEC－ 40 (17)
2008／3／6

機密情報の拡散追跡機能における情報漏えいの検知精度の向上手法

大橋 慶† 田端 利宏† 谷口 秀夫† 横山 和俊‡ 箱守 聰‡†

†岡山大学大学院自然科学研究科
700-8530 岡山市津島中 3-1-1

‡株式会社 NTT データ技術開発本部
135-6033 東京都江東区豊洲 3-3-9

あらまし 機密情報の漏えいを防ぐために，様々な手法が提案されている．これらの手法の 1 つとして，我々は，オペレーティングシステムのシステムコール発行を契機とし，機密情報が拡散する経路を刻々と追跡し，計算機外への漏えいを直前で検知する手法を提案している．しかし，提案手法は非機密ファイルを機密ファイルと誤って管理してしまう問題がある．そこで，ここでは，情報漏えい検知時の正答率を向上させる手法を提案する．具体的には，誤って機密ファイルとして管理されやすい各種アプリケーションの設定ファイルや履歴ファイルへの機密情報の拡散を防止し，機密ファイルとして管理させないことで，正答率を向上させる．また，利用者が機密ファイルの管理表を閲覧し編集できるようにすることで，誤って登録されたファイルの確認や除去が行える機構を示す．提案手法を Linux 上に実装し，評価した．これにより，情報漏えい検知時の正答率が向上したことを示す．

Improvement Method of the Detection Accuracy of the Information Leakage in Diffusion Tracing Function of Classified Information

Kei Ohashi† Toshihiro Tabata† Hideo Taniguchi† Kazutoshi Yokoyama‡
Satoshi Hakomori‡†

†Graduate School of Natural Science and Technology, Okayama University

‡NTT Data Corporation
Research and Development Headquarters

Abstract Recently, a leak of classified information has become a serious problem. To prevent the leak, various methods are proposed. We proposed the method to trace the classified information diffusion and to detect an information leak of classified information. As a matter, this method identify non classified file as classified file. In this paper, we propose a method to improve accuracy at the time of the information leakage detection. Specifically, we prevent classified information diffusion to config file and history file. In addition, our proposal function enable user easily to read and edit classified file table. We implement the proposal method on the Linux kernel and evaluate it and show that accuracy of the information leakage detection is improved.

1. はじめに

近年，個人情報や企業情報等の機密情報を電子データとして扱うことの増加と外部記憶媒体の大容量化に伴い，機密情報が計算機外へ漏えいする事例が増えている．これらの漏えいの主な事例[1]として，紛失や盗難に次いで挙げられるのは，不正アクセスをはじめとする外部要因よりも，誤操作や管理ミスといった内部要因が多い．

そこで，内部要因に関する情報漏えいを防止する様々なセキュリティ技術が研究されている．その研究の 1 つとして，ファイルごとにアクセスを許可するプログラムをホワイトリストに登録し，リスト外のプログラムにアクセス制御を設けることでウィルス等を原因とする漏えいを防ぐ方式[2]がある．しかし，この方式はホワイトリストに登録してあるプログラムを利用している際に，利用者の不注意から起こる漏えいを防止できない．

データ保護ポリシとシステムコールの履歴に基づくアクセス制御をオペレーティングシステム (以下 OS) レベルで行うことにより，利用者の不注意による情報の漏えいを防ぐ機構[3]も提案されている．ま

た，Security-Enhanced Linux(SELinux)[4] に代表されるセキュア OS は，計算機上の資源に対するアクセス制御の粒度を細かく設定することにより，不正アクセスにおける資源の不正な利用を防止できる．両者の問題点として，アクセス制御の粒度が細かすぎることや，データ保護ポリシの設定工数が多いことが挙げられる．また，設定が誤っていた際は，利用者の知らないうちに不正アクセスや情報の漏えいの発生を招いてしまう可能性がある．Windows Vista の User Account Control 機能[5] は，権限が必要になる度に利用者に確認を促すことで，利用者の知らないうちに起こるシステムの書き換えや情報の漏えいを防止できる．しかし，確認ダイアログが頻繁に表示されるため，煩わしいと感じる利用者が機能を停止させてしまう可能性も考えられる．

情報の漏えいは，プログラムが機密情報にアクセスしさまざまな経路で外部へ伝達することによって起こる．よって，その対策においては，アクセス制御による資源へのアクセス管理を実施することに加え，機密情報にいつ誰がどのようにアクセスするか，機密情報がどこにあるかを把握することが重要である．また，設定ミスによる情報の漏えいを防ぐため，設定が簡単であり，その把握が簡易な技術が求められる．

以上のことから，我々は，OS のシステムコール発行を契機とし，機密情報が拡散する経路を追跡し，外部への漏えいを検知する機能（以降，情報拡散追跡機能[6]）を提案している．本機能は，外部への漏えいを検知した際に，利用者へ書き出しの可否を促すことで，機密情報の外部への書き出しを制御することが可能である．しかし，非機密ファイルを機密ファイルと誤って管理してしまう問題がある．これを，正答率の低下問題と呼ぶ．この問題により，本来機密情報を持たないファイルが外部へ書き出される際にも，情報の漏えいが発生したと検知され，利用者へ通知される．つまり，信頼性と利便性が低下してしまう．

本論文では，誤って機密ファイルと管理されやすい，各種アプリケーションの設定ファイルや履歴ファイルに着目し，これらのファイルへの機密情報の拡散を防止することで，正答率を向上させる手法を提案する．また，利用環境として個人計算機環境と GUI 上を想定した上で，誤って管理対象となったファイルを利用者が容易に除去できる機構を提案する．提案手法を実装し，評価を行うことで，設定ファイルや履歴ファイルへの機密情報の拡散を防止でき，漏えいの検知精度が向上したことを示す．

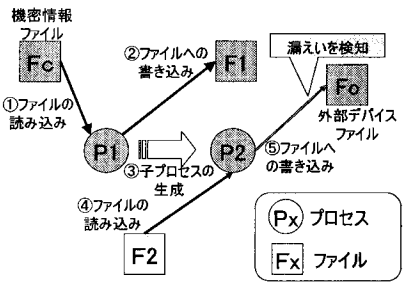

図 1 情報拡散グラフの例と追跡動作

2. 情報拡散追跡機能

2.1 情報拡散のモデル化と追跡法

情報拡散追跡機能は，ファイル操作や，子プロセスの生成に着目し，機密情報が伝達された可能性のあるプロセスやファイルを管理対象としていく．図 1 の情報拡散グラフの例を用いて，情報拡散追跡機能の動作モデルについて説明する．

(1) プロセス P1 が，管理対象である機密情報ファイル Fc に READ 処理を行った際，P1 に機密情報が伝達されることから，P1 は管理対象となる．(2) 次に，P1 がファイル F1 へ WRITE 処理を行う際，F1 は管理対象となる．(3) 管理対象となっているプロセス P1 が子プロセス P2 を生成した場合，子プロセスは親の資源を受け継ぐため，プロセス P2 も管理対象となる．(4) 生成されたプロセス P2 がファイル F2 を読み込んだ場合，機密情報は F2 へと伝達不可能であるため，F2 は管理対象とならない．(5) 最後に，P2 が外部デバイス上のファイル Fo に大して WRITE 処理を行うと，外部への漏えいを検知する．

以上のように，情報拡散の様子を有向グラフでモデル化し，プロセスの動作に応じたグラフの更新処理を行っていくことで情報の拡散を追跡し，漏えいの検知を行うことができる．

2.2 情報拡散追跡機能の処理の流れ

機密情報の漏えいを検知し，書き出しを制御する基本機構を図 2 に示し，以下に説明する．

(1) 情報拡散の契機となるシステムコールをフックし，情報拡散追跡処理を行う．

(2) 漏えいの可能性を検知すると，その書き出しを一時停止し，監視 AP へ漏えいの可能性を通知する．漏えいの可能性を検知しなかった場合，(6) の処理に移る．

(3) カーネルから受け取った情報に基づき，監視 AP が漏えいに関する警告を表示する．

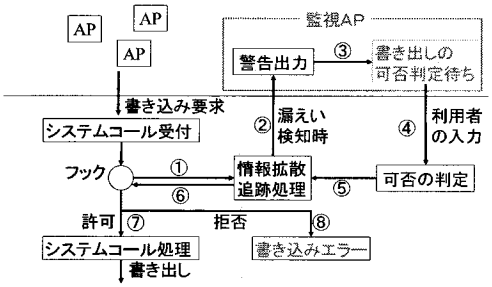

図 2　情報拡散追跡機能の基本機構

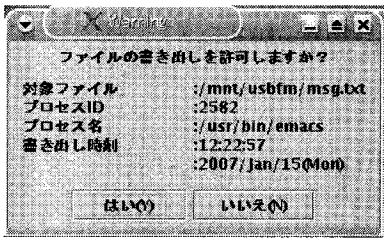

図 3　警告メッセージの例

(4)　利用者は，対象の書き出しの可否を判定し入力する．

(5)　書き出しの可否を情報拡散追跡機能へ返す．

(6)　システムコール処理へ復帰し，可否に応じた処理を行う．(2) で，漏えいの可能性を検知しなかった場合，通常の書込み処理を行う．

(7)　書き出しが許可された場合，通常の書き込み処理を行う．

(8)　書き出しが拒否された場合，書き込みエラーを返し，システムコール処理を終了させる．

上記処理により，漏えいの可能性があるシステムコールの実行前に「漏えいの可能性」を検知して監視 AP に通知できる．監視 AP は，システムコール実行を抑止した状態のまま，利用者の意思に応じた書き出しの制御が可能である．

2.3　計算機外部への書き出し制御手法

情報拡散追跡機能が漏えいを検知した際，漏えいに関する情報は監視 AP へと渡される．監視 AP は，カーネルから受け取った情報に基づき，GUI によるダイアログを表示する．このダイアログ上で，利用者に対して警告を表示し，書き出しを問う．GUI を用いることで，警告内容と操作が分かりやすくなり，利用者の操作ミスの危険性が少なくなる．例として，漏えいの可能性検知時に監視 AP が表示する警告メッセージを図 3 に示す．

対象ファイルは機密情報が書き出されようとしているファイルを示し，プロセス ID，プロセス名は書き出しを行ったプロセスを示す．この例は，テキストエディタ emacs において，管理対象ファイルをオープンした後，USB メモリをマウントしたディレクトリ /mnt/usbfm/ 上の msg.txt というファイルへの書き込みを行った際に表示されたダイアログである．ここで利用者が「はい」,「いいえ」のいずれかのボタンをクリックすると，そのボタンに対応した書き出し制御が行われた後，ダイアログは閉じられる．

3.　正答率の向上手法

3.1　正答率の低下

3.1.1　正答率の低下による影響

正答率の低下問題とは，本来機密情報を持ってないファイルが，機密情報を持つとして管理されてしまうことを指す．情報拡散追跡機能は，機密情報ファイルへのアクセスに着目し，その拡散を追跡する．よって，誤って機密情報と登録されたファイルが 1 つあった場合，その拡散が追跡され，新たなプロセスやファイルが誤って機密情報を持つと判断される．この拡散処理が繰り返されることで，計算機全体の資源に誤った機密情報が拡散する．つまり，計算機内の全てのファイルが管理対象と判断されてしまう恐れがある．これにより，本当の機密情報ファイルはどれであるか，利用者が把握できなくなる．さらに，全てのファイルの外部への書き出しに関して，利用者に可否を行うダイアログが出力されるようになり，利便性の大幅な低下を招くと共に，操作ミスによる情報の漏えいを招く危険性が増加する．また，拡散追跡処理を行う資源が増加することにより，機能の処理時間にも悪影響を及ぼす．

3.1.2　正答率低下の原因

前項で述べたように，正答率低下の原因は，機密情報を持っていないファイルが，管理対象であると判断されることにある．逆に，機密情報を持っているファイルが，管理対象ではないと判断されるということはない．

情報拡散追跡機能において，ファイルは管理対象プロセスからの書き込みにより，書き込まれる内容に関わらず管理対象となる．つまり，管理対象プロセスが機密情報とは全く関係のない情報を書き込むと，対象ファイルは機密情報を待たないにも関わらず管理対象となる．ここでは，この機密情報を待たない管理対象ファイルを，誤った管理対象ファイルと呼ぶ．誤った管理対象ファイルをあるプロセスが読み込むと，そのプロセスは機密情報を持たない管理対象プロセス (誤っ

た管理対象プロセス) になる．このプロセスが，さらに他のファイルへ書き込みを行ったり，子プロセスを生成することで，誤った管理対象ファイルやプロセスが増えていき，正答率のさらなる低下が発生する．

誤った管理対象ファイルになりやすいファイルの例として，文書編集 AP の設定ファイルや履歴ファイル(以降設定ファイルで統一) が挙げられる．その原因として，これらのファイルは，利用者の意図とは関係なく,AP の仕様により書き込みが行われるため，利用者は拡散を制御できないからである．例として，文書編集 AP である Emacs の設定ファイルが原因で起こる誤った機密情報の拡散の処理の流れを説明する．

(1) ある管理対象ファイルを読み込むと，Emacs は管理対象プロセスとなる．
(2) 利用者が管理対象ファイルを編集した後，Emacs を終了させる．
(3) この際，Emacs は，現在編集しているファイルのカーソルの位置を，".emacs-places"というファイルへ書き込む．
(4) ファイル".emacs-places"は管理対象となる．
(5) 利用者が改めて Emacs を起動した際，".emacs-places"は起動時に読み込まれ，Emacs は管理対象プロセスとなる．
(6) Emacs により編集されたファイルは全て管理対象として登録される．

この例では，設定ファイル".emacs-places"が誤った管理対象となることで，以降起動する Emacs が常に管理対象となる．つまり，AP の仕様により，正答率の低下が発生してしまう．この問題に対処するため，次節で設定ファイルが誤った管理対象となることを防止する手法を提案する．

3.2 正答率の向上手法

3.2.1 設定ファイルに着目した手法

設定ファイルが管理対象となることを防ぐために，以下に示す 3 つの手法を提案する．

(手法 1) 設定ファイルを，如何なる場合でも管理対象ファイルとみなさない．管理対象プロセスから設定ファイルへの書き込みは通常通り行う．
(手法 2) 管理対象プロセスから設定ファイルに対する一切の書き込みを防止することで，設定ファイルへの機密情報の拡散を防止する．
(手法 3) 設定ファイルを，如何なる場合でも管理対象とみなさない．管理対象プロセスから設定ファイルへの書き込みを一時的に行い，プロセス終了時に元に戻す．

(手法 1) は，設定ファイルに機密情報が拡散した場合でも，管理対象とみなさない．つまり，機密情報を持つ設定ファイルがあった場合でも，機密情報を持たないと判断される恐れがある．一方で (手法 2)，(手法 3) は，設定ファイルへの変更が加えられないため，正答率の低下は発生しない．しかし，両手法とも管理対象プロセスから設定ファイルへの編集に制限がかかる．さらに (手法 2) に関しては，AP が設定ファイルへ書き込んだ内容を必要とした場合に，AP の動作に影響を及ぼす可能性がある．(手法 3) では，設定ファイルへの書き込み内容は AP の起動中は反映されるため，AP の動作へ影響は少なくなる．しかし，設定ファイルへの余分な入出力処理が増えるため，オーバヘッドと実装コストは高くなる．本論文では，正答率とオーバヘッドの観点から，手法 2 を採用する．

3.2.2 利用者による管理対象ファイルリストの編集

管理対象となったファイルは，カーネル上の管理対象ファイルリストにその情報が登録される．このリストへの登録処理は，利用者へ一切通知されない．つまり，利用者の意図しない書き込みにより，気付かないうちに機密情報が拡散してしまう恐れがある．誤って管理対象となったファイルは，他のプロセスに読み込まれることで，さらなる正答率の低下に繋がる．これを防止するためには，ファイルが誤って管理対象とされたら，利用者は即座にそのファイルをリストから除去する必要がある．そのためには，以下の要件が必要となる．

(要件 1) ファイルが管理対象ファイルリストに登録された際，即座に利用者に通知可能であること
(要件 2) 利用者が管理対象ファイルリストを編集し，誤った管理対象ファイルをリストから除去可能なこと

各要件を満たすため，本論文では利用者の環境を GUI 上と想定した上で，監視 AP の機能を拡充し，メンテナンスツールの実装を行った経緯を報告する．

3.3 実　　装

3.3.1 設定ファイルへの機密情報拡散防止手法

提案方式を linux-2.6.0 上に実装した．本実装では，ファイル名，もしくはそのファイルの属するディレクトリ名が"．"で始まるファイルを設定ファイルとみなす．この設定ファイルに対して，管理対象プロセスが特定のシステムコールを発行した際，その処理を抑止する．具体的には，各システムコールはそのフック先で，本来正常に動作が行われた場合と同じ戻り値を返し，処理を終える．AP からは通常にシステムコール処理が行われたように見せることで，AP への動作への影響を少なくすることを可能にする．ここで，この

図 4 ポップアップメッセージの例

処理の実装を行った各システムコールの名前とその概要を表 1 に示す．管理対象プロセスから write システムコールが発行されることにより，ファイルは管理対象となる．設定ファイルが管理対象になることを防ぐため，管理対象プロセスから設定ファイルへの write システムコールの発行は抑止する必要がある．また，ファイル編集ソフトにおいては，ファイルの上書き処理の際は，既存のファイルを 0 バイトに切り詰める，もしくは，別のファイル名に変更した後，上書きする内容を write システムコールで書き込む．ここで，write システムコールの処理を拒否した場合，元の設定ファイルは空となってしまい，AP の動作に影響を及ぼす．これを防止するため，truncate, rename 各システムコールにも write と同様の処理の実装を行った．

3.3.2 メンテナンスツールの作成

(1) 利用者への拡散通知

既存の監視 AP では，外部への機密情報拡散を検知した場合のみ，図 3 で示したダイアログが表示される．(要件 1) を満たすため，内部への機密情報拡散の際にも，利用者への通知を行う必要がある．この際に，その都度ダイアログを表示していたのでは，利用者が煩わしく感じる可能性がある．そこで，図 4 に示すように，監視 AP のアイコンをシステムトレイに表示させ，拡散の際にアイコンからメッセージを表示させる方式で実装を行った．このメッセージは，表示から数秒後に自動的に画面から消えるため，ボタンをクリックしないと閉じないダイアログに比べ，利便性を損なうことが無い．この機能により，利用者は，意図しない書き込みで機密情報の拡散が発生した際，その確認と，拡散対象のファイルパスを知ることができる．

(2) 管理対象ファイルリストの編集

(要件 2) を満たすため，図 5 に示す管理対象ファイルリストエディタを実装した．このエディタは，左から順に，連番，管理対象となっているファイルのパス，i ノード番号，ファイルに機密情報を拡散させたプロセス名，及びファイルに機密情報が拡散した日時を表示

表 1 処理の実装を行ったシステムコール

システムコール名	処理内容
write	ファイルに書き込みを行う
truncate	指定した長さにファイルを切り詰める
rename	ファイルの名前や位置を変更する

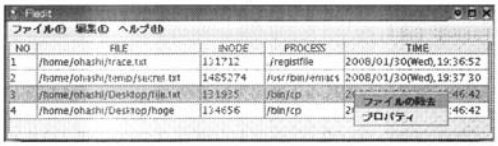

図 5 管理対象ファイルリストエディタ

している．図 5 では，ファイルを選択し，右クリックした際に出るメニューを表示している．このメニューで「除去」を選択する，もしくは Delete キーを押すことで，カーネルの管理対象ファイルリストから，選択ファイルが除去される．ファイルは複数選択も可能で，まとめて管理対象ファイルをリストから除去することも可能である．

簡単な操作でファイルの除去を行えることから，計算機の扱いに慣れてない利用者でも操作のミスが少なくなる．また，エディタ上の"NO","FILE"等のヘッダ部をクリックすることで，そのヘッダに応じたソートが行われる．ファイルのパスや，追加された日時に応じたソートを行うことで，様々な視点で管理対象となっているファイルの確認をできる．エディタは，管理対象ファイルリストへのファイルの追加にも対応している．利用者が誤って機密情報を持つファイルをリストから除去してしまった場合，再度そのファイルをリストに登録することができる．

4. 評　　価

4.1 評価の概要

正答率の向上手法の実装後，実際にファイル編集作業を行う場合を想定し，以下に示す 3 つの事例を適用した．

(事例 1) OpenOffice.org2.2 を用い，管理対象ファイル 1 を編集し，ファイル 2 として保存する．

(事例 2) GNU Emacs を用い，管理対象ファイル 1 を編集し，ファイル 2 として保存する，一度プログラムを終了した後に一般ファイル A を編集し，ファイル B として保存する．

(事例 3) KEdit(KDE の標準テキストエディタ) を用い，管理対象ファイル 1 を編集し，ファイル 2 として保存する．一度全てのファイルを閉じた後に一般ファイル A を開き，ファイル B として保存する．

評価は X Windows 上で行い，それぞれの事例の実行過程で，AP の動作にどのような影響が及ぶか調査を行った．

表 2 事例への適用結果
Table 2 Detection examine results

	事例 1		事例 2		事例 3	
	実装前	実装後	実装前	実装後	実装前	実装後
管理対象となったファイル数	5	2	5	2	8	3
管理対象となった設定ファイル数	3	0	2	0	5	0
操作後の機密ファイル数	2		2		2	
ファイルの正答率 (%)	40.0	100	40.0	100	25.0	66.7
管理対象となったプロセス数	1	1	2	1	2	2

4.2 評価結果と考察

4.2.1 正答率の評価

事例 1〜3 を適用した結果を表 2 に示す．正答率向上処理の実装後は，全ての事例において，設定ファイルが管理対象となることはなくなった．これにより，実装前の（事例 2）で起こった，設定ファイルが管理対象となることで，その後起動した関連プロセスが全て管理対象となる事態を防止することが可能となった．本実装により，（事例 1）と（事例 2）に関しては，正答率は 60%向上し，100%となった．その一方で，（事例 3）に関しては，設定ファイル以外のファイルが 1 つ誤った管理対象となっているため，正答率の向上は 40%程度に留まった．この原因は，管理ファイルをクローズしても，プロセスは管理対象となっており，通常ファイルを編集し保存した場合，そのファイルは管理対象となってしまうからである．

ファイルが管理対象となる際は，監視 AP により，ポップアップメッセージが表示され，利用者に機密情報の拡散が通知される．よって（事例 3）のように，設定ファイル以外のファイルが誤って管理対象となった際は，利用者は図 5 に示すファイルリストエディタを用いることで，管理対象ファイルリストからファイルの除去を行うことができる．

4.2.2 AP の動作への影響

正答率の向上手法を実装することにより，管理対象プロセスは，設定ファイルへ変更を加えることはできない．これにより，管理対象プロセスが編集していたファイルのカーソル位置が保存されない，履歴ファイルにファイル名が残らないといった影響が生じた．また，管理対象プロセスにより，設定ファイルそのものを意図的に編集することもできない．この場合は，一度プロセスを起動しなおし，管理対象ではない状態で編集を行う必要がある．AP の動作への影響という点においては，評価に用いた 3 つの AP への事例の適用過程で，その動作に深刻な支障を来たす事態は発生しなかった．この影響に関しては，様々な AP を用いてさらに詳しく調査する必要がある．

5. おわりに

本論文では，機密情報の拡散追跡機能を実際に利用する上で問題となる，正答率の低下に関しての対策を述べた．まず，誤った管理対象となりやすい設定ファイルへ着目し，これらのファイルへ機密情報の拡散を防止する手法を示した．そして，機密情報の拡散を利用者に通知し，誤った管理対象ファイルを利用者が除去できる機構の提案を行った．提案手法の実装により，ファイルの正答率は 40%以上向上することを示した．

残された課題として，オーバヘッドの評価，及び AP の動作の影響への検討がある．

謝辞 本研究の一部は，C&C 振興財団 若手研究員助成，及び中島記念国際交流財団日本人若手研究者研究助成の支援を受けて行った．

参考文献

1) 日本ネットワークセキュリティ協会，2006 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 ver1.0，(2007)．
2) 喜田弘司，坂本 久，島津秀雄，垂水浩幸，ファイルアクセス制御エージェントの提案—P2P 型ファイル共有システムのセキュアな利用を目指して，情報処理学会論文誌：コンピューティングシステム，Vol.48，No.1，pp.201-212，(2007)．
3) 鈴来和久，一柳淑美，毛利公一，大久保英嗣，Privacy-Aware OS Salvia におけるデータアクセス時のコンテキストに基づく適応的データ保護方式，情報処理学会論文誌：コンピューティングシステム，Vol.47，No.SIG 3(ACS13)，pp.1-15，(2006)．
4) NSA，Security-Enhanced Linux，URL = http://www.nsa.gov/selinux/.
5) MSDN, Windows Vista デベロッパー センター - セキュリティ, URL = http://www.microsoft.com/japan/msdn/windowsvista/security/.
6) 大橋 慶, 箱守 聰, 田端 利宏, 横山 和俊, 谷口 秀夫, 機密情報の拡散追跡機能を利用した書き出し制御手法, マルチメディア・分散・協調とモバイル（DICOMO2007）シンポジウム論文集, pp690-697, 2007.