情報漏洩防止プラットフォーム

横田侑樹† 塩谷亮太† 五島正裕† 坂井修一†
†東京大学

1 はじめに

近年，情報セキュリティ事故の中でも情報漏洩の被害は甚大なものとなっている。

情報は，一般に，所有者・権利者から，利用者へと配布されて，そこで利用される。あるいは，所有者・権利者自身が利用者であることもある。現在問題となっている情報漏洩は、以下のいずれかに属する：

- 所有者・権利者自身が利用者として情報を利用するにあたり，過失により，意図した範囲外へ情報を流出させた
- 所有者・権利者との間で利用契約を交わして情報を受け取った利用者が，故意や過失により，契約に定められた範囲外へ情報を流出させた

いずれも，利用者（の使用するコンピュータ）から，当初意図された範囲外へ情報を流出させたものである。

利用者における漏洩は，配布段階における漏洩より対策が難しい。利用者は契約により利用が認められており，利用者がその内容を知ること自体を妨げることはできない。利用者が知ること自体を妨げるのではなく，利用者が知った内容を第三者に漏らさないように保障することが必要となる。

本稿では，プログラムの信頼性に依存しない情報漏洩対策手法を考察し，情報漏洩防止機能を持ったプラットフォームの提案を行う。

2 関連手法

2.1 アクセス制御

OS等によって提供されるアクセス制御機構は、保護の意思決定がファイルの所有者に委ねられており、所有者は保護の権限を自由に設定できるため，漏洩を防ぐことは難しい。

これに対して、強制アクセス制御 (MAC) システムでは，ファイルの所有者による保護設定に加えて，システムのセキュリティポリシーに基づく保護を加える。これは、単一のシステム自身の一貫性を守ることには有用であるが、情報の権利者と情報の利用者とが異なる状況での情報漏洩を阻止することは出来ない。

2.2 デジタル著作権管理手法

デジタルデータとして表現されたコンテンツの著作権を保護する技術として、デジタル著作権管理 (DRM) 手法がある [1]。例えば，コンテンツを暗号化された状態で配布し，特定のプログラムでしか復号化・再生を出来ないようにすることで，第三者による複製や利用を困難にさせる等の方法が取られる。

しかし，コンテンツの保護は復号プログラムの信頼性—プログラムの耐タンパ性—に依存している。一般的に，プログラムの耐タンパ性は，暗号強度に比べて向上が難しく，リバースエンジニアリング等によってコンテンツが漏洩する可能性がある。また，コンテンツの復号が特定のプログラムに依存するので，ユーザの利便性が損なわれる可能性もあり、包括的な情報漏洩対策手法であるとは言い難い。

2.3 インフォメーションフロー追跡

個々のアプリケーションの信頼性に依存しない情報漏洩対策手法に，インフォメーションフロー追跡手法がある [2]。アプリケーション実行環境によりプログラム実行時の情報の流れを解析し、入力と出力の依存関係を把握することで、保護すべきデータが誤って出力されてしまう事態を検出し防止する手法である。

しかし、実行時の制御フローによって起こる暗黙的フローなど、追跡が困難なインフォメーションフローがあり、あらゆる漏洩を防ぐことは出来ないのが現状である。

3 提案手法

3.1 アプローチ

本手法では、各データが、どのデバイスへの出力が認められ、また防止されるのかをポリシーとして定義し、それに基づいて入出力を制限する。本手法では、出力制限をセキュリティ・レベル (S.L.) と呼称し、制

A Platform for Preventing Information Leakage
Yuki Yokota†, Ryota Shioya†, Masahiro Goshima†and Shuichi Sakai†
†The University of Tokyo

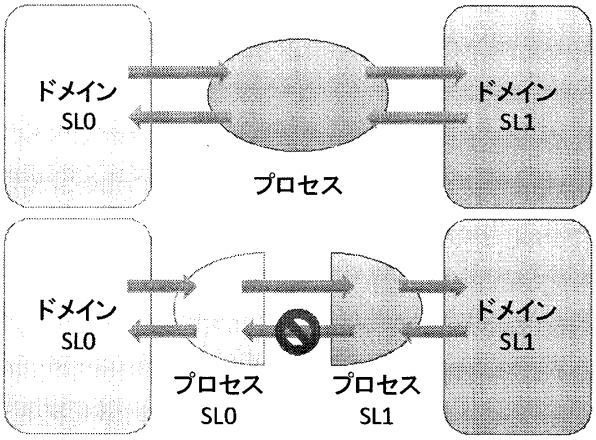

図 1: S.L. による分割

限の緩いものを S.L.0、厳しいものを S.L.1 とする。プラットフォーム上の各データに S.L. を関連付け、S.L.1 から S.L.0 への漏洩を防止する。

情報の操作を行うのはプロセスである為、プロセスの状態を S.L. に基づいて分割する（図 1）。そして、情報漏洩は、S.L.1 プロセスによるシステムの変更が S.L.0 プロセスによって観測される事であると考える。この視点に基づき、システム上のデータも S.L. に基づいて分割し、S.L.0 プロセスによる S.L.1 データの観測と、S.L.1 プロセスによる S.L.0 データの変更を禁止する。

3.2 データへの S.L. 付与

S.L. の付与対象は、プロセスとプロセスの操作するデータである。プロセスへの S.L. 付与の為、プロセス構造体に S.L. を表現するフィールドを追加する。

プロセスの操作するデータは大別してファイルとメモリ上のページである。ページへの S.L. 付与は、プロセスの所持するページテーブルを用いて行う。このため、プロセス構造体にページテーブルを二つ用意し、プロセスの各 S.L. 毎にページへのアクセス権限を示す。

ファイルへの S.L. 付与は、パーティションを分割して S.L. 毎に割り当て、ファイルを同じ S.L. を持つパーティションに配置することで行う。

3.3 S.L. によるアクセス制御

あるプロセスが、プロセスと異なる S.L. を持つデータを含むページやファイルを操作する場合、制限を掛ける（表 1）。これは、S.L.1 から S.L.0 へのインフォメーションフローを禁止するためである。

3.4 プロセスの S.L. の切り替え

プロセスは自身の S.L. を変更しながら動作を行う。この S.L. の変更に際し、制御フローからの依存性を排除し、また S.L. 変更のアトミック性を保証するため、専用のシステムコールを導入する（図 2）。この専用システムコールは、コンテクスト（レジスタ、ファイル構造体、ページテーブル）の保存／復帰とジャンプを不可分に行う。

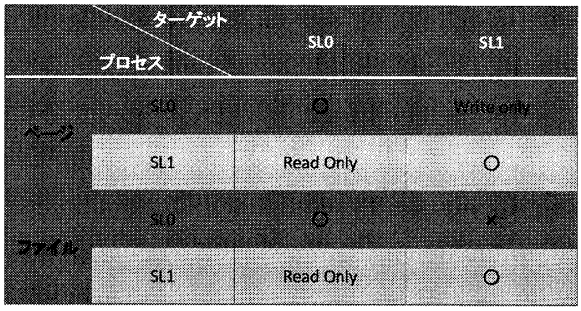

表 1: S.L. によるアクセス制御

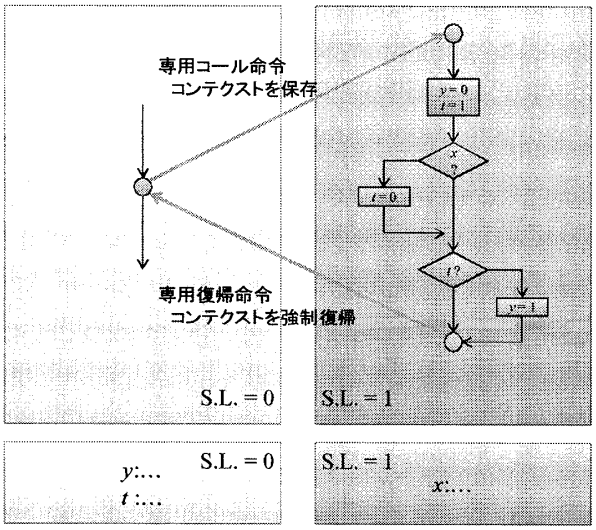

図 2: S.L. の切り替え

参考文献

[1] Jamkhedkar, P. A. and Heileman, G. L.: Drm as a layered system, in *DRM '04: Proceedings of the 4th ACM workshop on Digital rights management*, pp. 11–21, New York, NY, USA (2004), ACM Press.

[2] 栗田弘之, 塩谷亮太, 入江英嗣, 五島正裕, 坂井修一：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, 於 北海道大学学術交流会館, Vol. 2007, No. 17, pp. 227–232 (2007).