# Linux 上のグループ管理が可能な暗号ファイルシステム

ルアンルアンリット・サワット † 杉岡一郎 †
室蘭工業大学大学院生産情報システム工学専攻 † 室蘭工業大学情報工学科 ‡

## 1 はじめに

CFS、Cryptfs や BestCrypt など、今までの Linux 上の暗号ファイルシステムの多くは、秘密情報は所有者にしかアクセスされないという想定で開発されてきた。しかし、ソフトウェア共同開発など、秘密情報が何人かの人に共有される使用形態においては、このような前提が厳しい制限となる。TCFS [1] は UNIX 標準のグループ機構である GID のサポートが備えられているものの、各ユーザ個別のアクセス権の概念がないため、本格的な「暗号ファイルの共有」の使用には向いていない。

そこで本研究では、この問題を解決するために Linux 上でグループ管理が可能な新しい暗号ファイルシステムの設計・実装に取り組んでいる。本システムは以下の性能を保証する機能の実装を目指している。

- 暗号ファイルの所有者が各利用者のアクセス権を細かく管理できる。
- パスワードや公開鍵暗号の技術などを使用した柔軟な認証方式が用意されている。
- 利用者が不便と感じない程度の処理速度を有する。

## 2 実装方針

### 2.1 カーネル・モジュールとしての実装

本研究では、暗号ファイルシステムの機能をカーネル・モジュールとして実装する方法を採用している。カーネル空間で実装した暗号ファイルシステムは、ユーザ空間で動作するプロセスと違って、コンテキスト・スイッチングなどの影響を受けないため、高いパフォーマンスを期待できるという利点がある。また、モジュールとして実装した場合では、暗号ファイルシステムが使われているか否かに応じて、カーネルはそのモジュールを自動的にロード /アンロードできるので、メモリなどの資源が有効利用される。

### 2.2 仮想ファイルシステム層での実装

本研究では、File System Stacking [2] と呼ばれる技法を用いて、カーネルの仮想ファイルシステム (VFS: Virtual File System) 層に暗号ファイルシステムの機能を実装する方法を採用している (図 1)。

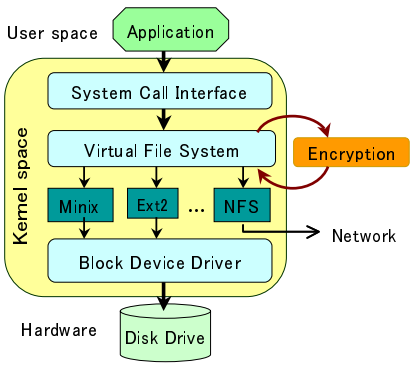


図 1: VFS 層での実装

この方法では、ディスクへの実際の書き込みを下位レベルのバックエンドファイルシステムに任せ、暗号・復号化の処理の流れをコーディングが行なわれる。また、現在使用されている多くのファイルシステムをそのままバックエンドファイルシステムとして使うことが可能となり、利用者にとって受け入れやすく、信頼性の面においても有利である。この時、バックエンドファイルシステム層においても、ファイルとして認識されるので、例えば、そのバックエンドファイルシステムに対応したバックアップツールを使えば、ファイルが暗号化されたまま簡単にバックアップすることができる。また、NFS を通じて公開すれば、ネットワークに対応した暗号ファイルシステムと同等な機能を得ることができる。

## 3 本システムの概要

### 3.1 システムの構成

まず、本暗号ファイルシステムの構成要素である所有者、利用者及び不正利用者について説明する (図 2)。

1. **所有者** (Owner)
   秘密情報 (以下、秘密ファイルと記す) の所有者。**所有者パスワード** *(OP: Owner Passphrase)* を持ち、暗号ファイルシステムをマウントする時に、それをカーネルに提供する。
2. **利用者** (User)
   所有者からアクセス権限が与えられた秘密ファイルの利用者。暗号ファイルシステムにアクセスする時に認証に使われる**利用者パスワード** *(UP: User*

---

**Cryptographic File System with Group Management**
†Sawat LUENGRUENGRIT, Div. of Production and Information Systems Engineering, Muroran Institute of Technology
‡Ichiro SUGIOKA, Dept. of Computer Science and Systems Engineering, Muroran Institute of Technology

*passpharse)* を持つ。

3. **不正利用者** (Adversary)
   秘密ファイルのアクセス権が与えられていないにも関わらず、それをアクセスしようとする者。

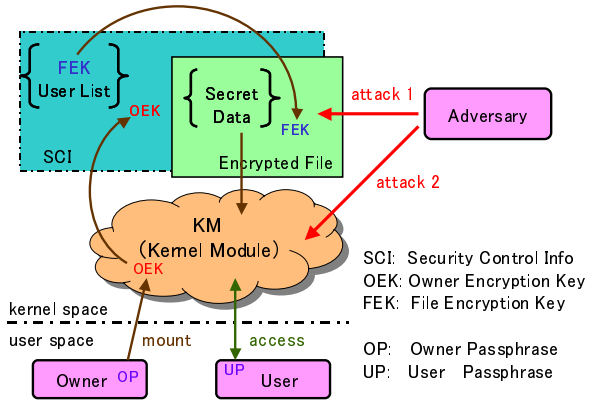


図 2: システム構成図

次に、暗号復号化の処理、利用者のアクセス管理を行なうカーネル・モジュール (以下、KM と記す) について述べる。KM は所有者から与えられた OP を一方向ハッシュ関数で**所有者暗号鍵** *(OEK: Owner Encryption Key)* に変換する。利用者が秘密ファイルにアクセスする時、KM は利用者に代ってそのアクセスを代行する。この時 KM はまず、その秘密ファイルに付属した**セキュリティ制御情報** *(SCI: Security Control Information)* を調べる。SCI は OEK によって暗号化されているが、KM はそれを復号化し、利用者が SCI が規定している本人認証を完了したかどうか、また、その利用者はアクセス権を持っているかどうかを確認する。そして、KM は SCI に含まれる情報の一つである**ファイル暗号鍵 (***FEK: File Encryption Key***)** を使って、秘密ファイルの内容を復号化し、アクセス権を持つ利用者に提供する。

このように OEK と FEK の 2 重鍵による暗号方式を使う理由として、各々のファイルには異なる鍵 (FEK) が使うことができ、情報の秘密性が高まることが挙げられる。また、所有者が OP を変更する時、比較的に小さい SCI を再暗号化するだけで済み、秘密ファイルそのものを再暗号化する必要はない。

### 3.2 不正利用者への対処

不正利用者がハードディスクに直接アクセスできる状況にあるとしても、不正利用者は OEK を知らないためファイルの内容を解読することはできない。これはハードディスクが物理的に盗まれても、本暗号ファイルシステムに格納する情報の秘密性が守られていることを意味している。

また、もし、不正利用者が KM がすでに稼動しているシステムにアクセスでき、しかもシステム管理者権限を取得してしまっているのであれば、OEK を入手することができ、このファイルシステムにある全ての秘密ファイル内容を解読できることを意味するが、所有者パスワード (OP) を知ることまではできない。そのため、不正利用者は所有者が管理している、まだマウントされていない他の暗号ファイルシステムにアクセスすることはできない。

### 3.3 グループ管理機構

各暗号ファイルに付属している SCI は、OEK によって暗号化され、三つの部分に分けられる。

1. 暗号方式: 秘密情報が暗号化されているかどうか、暗号方式や FEK などが記述される。
2. 利用者アクセス権のリスト: アクセスできる利用者情報、アクセス権と認証方式が記述される。
3. 秘密情報の完全性チェック: 秘密情報が改ざんされていないかどうかをチェックする部分である。

この SCI 機構はカーネル 2.6 系統で提供される Extended Attribute、CryptoAPI、LSM (Linux Security Modules [3]) などで実装する予定である。

## 4 おわりに

実装及び評価の方針については、概ね三段階に分けることができる。

1. カーネルの VFS 層でのインタフェースの実装
2. SCI などグループ管理機構の実装
3. 性能評価

現在は、第一段階まで実現している。 実装したプロトタイプはカーネル 2.4 系統であり、ioctl システムコール経由で暗号鍵の受け渡しを行なった。暗号アルゴリズムは Blowfish の CFB モードであり、そのバックエンドファイルシステムは ext2 である。

しかし、カーネル 2.6 系統に導入された新機構の中に今後の本暗号システムの実装で使用予定の機能が含まれているため、プロトタイプの移行が必要である。

そこで現段階では、移行作業と並行しながら第 2 段階のグループ管理機能をサポートする SCI 機構などの実装に取り組んでいる。性能評価では、主な評価項目として処理速度と安定性などを取り上げる予定である。

## 参考文献

[1] G. Cattaneo, et al. The Design and Implementation of a Transparent Cryptographic File system for UNIX. In *Proceedings of the Annual USENIX Techincal Conference, FREENIX*, pp. 245–252, June 2001.

[2] J. Heidemann, et al. File-system development with stackable layers. *ACM Transactions on Computer Systems (TOCS)*, Vol. 12, No. 1, pp. 58–89, 1994.

[3] C. Wright, et al. Linux Security Modules: General Security Support for the Linux Kernel. In *Proceedings of the 11th USENIX Security Symposium*, August 2002.