

5D-6 MOCFS : 移動計算機上の情報保護を 目的としたファイルシステム

磯貝 悟[†] 中村 嘉志[†] 多田 好克[†]
電気通信大学 大学院情報システム学研究科[‡]

はじめに

本発表では、移動計算機上での使用を目的とした暗号化ファイルシステム (MOCFS*) について議論をする。MOCFS は、ディスクへの書き込み時に情報の暗号化を読み込み時に復号化を自動的におこなうシステムである。このシステムは、書き込み情報の暗号化により情報の漏洩を防止する。また、自動化によってユーザによる情報の手動暗号化による誤りを防ぐことができる。さらに暗号／復号鍵の適切な管理により、移動計算機が盗難や紛失の可能性が高い場所でも移動計算機を使用をすることを可能にする。

1 動機

移動計算機の小型化、高性能化によって、利用者は移動計算機上に様々な情報を保持するようになってきた。その利便性から、利用者は、貴重な情報を保持した移動計算機を携帯し使用するため、盗難や紛失の可能性が増加してきている。そこに保持されている情報が増加するほど、盗難や紛失の問題は移動計算機自体の損失よりもむしろそこに保持されている情報の漏洩と喪失にある。特に漏洩は二次的に損害をもたらすことになってしまう。

漏洩を防ぐための最も単純な方法としては、ユーザ自身が必要に応じて情報を暗号プログラムなどで暗号／復号化をおこなうことである。しかし、この方法は利便性に欠け、暗号化をし忘れて平文を保存してしまうことが考えられる。

そこで本研究では、これらの問題を防ぐ方法として移動計算機内の情報を自動的に暗号化して保存するこ

とにした。さらに必要に応じてその情報を復号化するファイルシステムを提案し、そのプロトタイプを作成し評価する。

2 目的

提案するシステムは、移動計算機が盗難や紛失の可能性が高い場所で使用されることを前提としている。このため、以下の二つの経路からの情報漏洩防止を目的としたシステム構築をおこなわなければならない。また、この他にハイパネーション領域の情報の保護も考慮する必要がある。

1. ディスクの紛失や盗難による情報の漏洩
2. 放置端末からの情報の漏洩

本システムは 1) を防止するために情報を暗号化してディスクに保存する。また、自動的に情報の暗号化をおこなうことにより、ユーザの手動による情報の暗号化による誤りを防止する。

2) は復号化鍵情報をタイマにより制限することにより防止する。

3 関連研究

関連研究として以下をあげることができる。

- **SFS**[1]: このシステムの利点は移動計算機での使用を考慮しているため、タイマによる暗号／復号鍵管理をおこなっている。しかし書き込み時にタイムアウトしている場合に平文がディスクに書き込まれてしまうという欠点がある。また、mmap システムコールに未対応であり、暗号化された情報がマッピングされてしまうためにバイナリ情報を扱うことができない。
- **CFS**[2]: このシステムは NFS をベースとして暗号／復号化をおこなっている。そのため移植性が

MOCFS: Protecting File System against data leakage from a mobile computer

* Mobile computer Oriented Cryptographic File System

[†] Satoru Isogai, Yoshiyuki Nakamura, Yoshikatsu Tada

[‡] Graduate School of Information Systems, The University of Electro-Communications.

表 1: ファイルシステムとの比較				
	SFS	CFS	TCFS	MOCFS
移植性	×	○	×	○
mmap 対応	×	○	○	○
鍵管理	△	×	×	○

高く、全てのシステムコールを通常のファイルシステムと同様に使用できる。しかし、ファイルシステムがマウントされてしまうとアンマウントされるまで永続的に使用できてしまうという欠点がある。

- TCFS[3]**：このシステムは上記の CFS を拡張したもので CFS に比べて透過性にすぐれているが、Linux カーネルに統合されているために移植性が低い。

本研究で開発をおこなっている MOCFS は NFS をベースとし、そこに独自の鍵管理機構を付加することにより移動計算機環境での情報の保護をおこなう。表 1 に SFS、CFS、TCFS および本研究で実装する MOCFS との比較を示す。

4 実装

本システムは、NFS 互換サーバ、鍵管理機構の二つから構成される。NFS 互換サーバは、情報の暗号／復号化を実際におこなう部分である。鍵管理機構はユーザから与えられた鍵の認証と管理をおこなう部分である。

図 1 に本システムの構成を示す。

4.1 NFS 互換サーバ

NFS 互換サーバは、アプリケーションが情報の書き込みを要求すると、鍵管理機構から得た鍵情報をもとに情報の暗号化をおこない、暗号化情報をディスクに保存する。

同様に、アプリケーションが情報の読み込みを要求すると、ディスクに保存された暗号化情報を鍵情報をもとに復号化をおこないアプリケーションに復号化情報を渡す。

Unix カーネルとの通信を NFS インタフェイスを介しておこなうことにより、MOCFS はユーザレベルで実装が可能となる。

4.2 鍵管理機構

鍵管理機構は、ユーザからの鍵登録とタイマによる復号化のための鍵情報の破棄をおこなう。これにより NFS 互換サーバが常に暗号化情報を復号化して使用できることを不可能し、移動計算機が盗難や紛失の可能性が高い場所での移動計算機の使用を可能にする。

また、ハイパネーション時に暗号／復号化鍵情報を破棄することにより安全性を確保する。

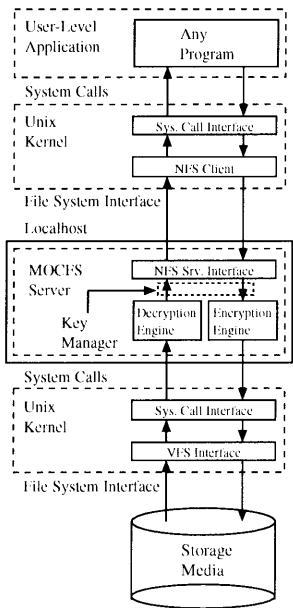


図 1: 本システムの構成

5 まとめ

現在、NFS 互換サーバの実装と、鍵管理機構の実装をおこなっている。今後の予定として NFS 互換サーバの性能評価をおこなう。

参考文献

- [1] 斉藤 紀, 松永良太郎, 暗号ファイルシステムの試作, 1997, <http://www.netlaputa.ne.jp/%7Eeossan/freebsd/securefs/index.html>
- [2] Blaze, M., A Cryptographic File System for Unix, *ACM Conference on Computer and Communications Security*, pp. 9–16, Nov. 1993. <http://www.crypto.com/papers/cfs.ps>
- [3] Luigi, C., Aniello, S., and Luigi, M., Transparent Cryptographic File System, <http://tcfs.dia.unisa.it/>