L-008

秘密分散法を用いた秘匿性と耐障害性を備えたファイルシステムの開発
Development of a File System with Confidentiality and Fault Tolerance Using Secret Sharing Scheme

永沼 祥吾[1)]　滝 雄太郎[1)]　藤田 茂[2)]
Shogo Naganuma　Yutaro Taki　Shigeru Fujita

1　はじめに

インターネットサービスの一つとして，WCS(Web Cloud Storage) がある．ただし，WCS を扱う場合，個人情報漏洩，単一ストレージへの集中管理の課題がある．その課題を解決する 1 つの手法として，秘匿性と耐障害性を同時に保証する秘密分散法がある [1]．しかし，秘密分散法を応用した商用システムは現時点では広く一般には普及していない．この理由として，元の情報よりもデータ長が大きくなるため，頻繁にデータをやり取りする環境では実用的ではない点が考えられる．

そこで，本研究では秘密分散法を適用した WCS が実用に耐えるために，独自のファイルシステムを開発できる FUSE(Filesystem in USErspace)[2] を用いて，秘密分散法を用いたファイルシステムを開発する．本研究では既存のネットワークファイルシステムと比較して性能の劣らないファイルシステムを開発し，秘密分散法による秘匿性と耐障害性を備えたファイルシステムの有効性を示す．

2　秘密分散法

2.1　Shamir の (k,n) 閾値秘密分散法

秘密分散法は Shamir により提案された (k,n) 閾値秘密分散法がある [1]．秘密情報を n 個のデータ (シェア) に分散し，k 個以上のシェアからは秘密情報を復元できるが，k 個未満では復元できないという性質を持つ．ただし，Shamir の (k,n) 閾値秘密分散法は $k-1$ 次の多項式となることから計算負荷が大きい．したがって本研究では Shamir の (k,n) 閾値秘密分散法は用いない．

2.2　XOR による (k,n) 閾値秘密分散法

栗原らによって提案された XOR による秘密分散法 [3] は Shamir の秘密分散法より高速な処理が可能で，$k=3, n=11$ のときに 900 倍以上高速となったことを示した．したがって本研究では，XOR による (k,n) 閾値秘密分散法を用いる．本研究では $k=2$，$n=3$ を適用し，そのアルゴリズムを以下によって分散・復元される．

分散アルゴリズム

1. 秘密情報を $S \in \{0,1\}$ として，S_1, S_2 に均等分割する．
2. 乱数を $R \in \{0,1\}$ として，S_1, S_2 と同サイズの R_1, R_2 を生成する．
3. 部分分散情報 W_{ij} を生成し，分散情報 W_1, W_2, W_3 として次のように結合する．ただし，|| はデータの連結を示す．

$$W_1 = W_{11} \,||\, W_{12}$$
$$W_2 = W_{21} \,||\, W_{22}$$
$$W_3 = W_{31} \,||\, W_{32}$$

復元アルゴリズム

分散情報 W_1, W_2 を取得した場合：

1. 分散情報 W_1, W_2 の持つ部分分散情報 W_{ij} から S_1, S_2 を生成する．

$$W_{11} \oplus W_{21} = S_1$$
$$W_{12} \oplus W_{22} = S_2$$

2. 生成された S_1, S_2 を結合することで秘密情報 S を得られる．

$$S = S_1 \,||\, S_2$$

部分分散情報 W_{ij}

部分分散情報 W_{ij} は以下のようにおく．

$$W_{11} = R_1$$
$$W_{12} = R_2 \oplus S_2$$
$$W_{21} = R_1 \oplus S_1$$
$$W_{22} = R_2$$
$$W_{31} = R_1 \oplus S_2$$
$$W_{32} = R_2 \oplus S_1$$

3　提案手法

3.1　FUSE

FUSE はユーザ空間上のプログラムが，ファイルシステムを Linux カーネルにエクスポートするためのインタフェースである [2]．FUSE は FUSE カーネルモジュールと FUSE ライブラリ (libfuse) で構成されており，その仕組みを図 1 に示す．ファイルを読み書きするプロセスが発生すると，標準ライブラリ (libc) からシステムコールが生成されて，それをカーネルの VFS(Virtual File System) に渡す．VFS は様々なファイルシステム上で統一された基本的な機能を提供しており，FUSE カーネルモジュールはその機能を受け取ることができる．FUSE カーネルモジュールは，そのシステムコールの実際の処理を独自のプログラムに依頼する．FUSE はその依頼に対して独自の処理を付加することが可能であり，その処理結果を FUSE カーネルモジュールへ返す．FUSE カーネルモジュールは VFS を通してユーザに返すことで処

1)　千葉工業大学 大学院 情報科学研究科
Graduate School of Information and Computer Science, Chiba Institute of Technology
2)　千葉工業大学 情報科学部
Department of Computer Science, Faculty of Information and Computer Science, Chiba Institute of Technology

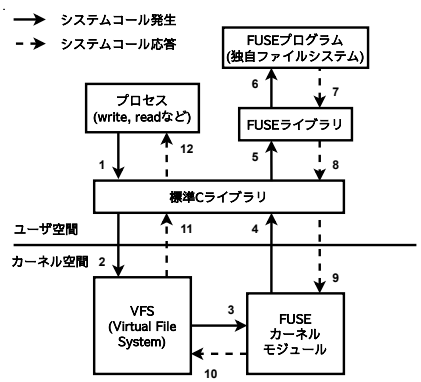

図 1　FUSE の仕組み

理が完了される．

このように FUSE カーネルモジュールは，ユーザ空間の FUSE ライブラリを介して，ユーザが独自のファイルシステムを実装できるようにしている．FUSE は Linux，FreeBSD，Android，macOS などの様々な OS と互換性がある．また，独自のファイルシステムの実装には，カーネルコードを修正することなく用意された関数のみで実装できることから容易にファイルシステムを開発できるという利点がある．

3.2　分散/復元手順

本システムの分散/復元手順について，図 2 に示す．本システムを動作させるにはマウントポイントを決定する．そのマウントポイントにファイル (秘密情報) を mv コマンドなどで配置することで分散される．分散時にはディレクトリ内にシェアと復元用ファイルが生成される．復元時にはディレクトリ内の復元用ファイルを指定して読み込むことで秘密情報が復元される．その提案手法の分散・復元手順について以下に示す．

分散手順

1. 任意のディレクトリをマウントし，秘密情報となるファイルを配置する．
2. 配置時にファイルが分散され，ディレクトリ内にシェアと復元用ファイルが生成される．

復元手順

1. ディレクトリ内に復元に必要数のシェアを用意する．
2. ディレクトリ内にある復元用ファイルを読み込む (read システムコール発生)．
3. read システムコールを受け取ると各シェアを結合して元のファイル (秘密情報) を復元する．

4　評価実験

本研究で開発したファイルシステムは分散ファイルシステムである NFS，Samba，GlusterFS と write/read 性能を比較する．各ファイルシステムはサーバとファイル共有するときに使われる．提案システムはファイルを分散保存することを目的とするため，各ファイルシステムと同様の使い方を想定している．

測定には IOzone を使用し，ファイルサイズを 1〜512MB の範囲で変化させて 10 回測定する．その結果を図 3，4 に示す．提案システムの write 性能は NFS の 1.2 倍，GlusterFS の 2.2 倍速く，Samba の 1.5 倍遅くなった．また，read 性能は，GlusterFS の 1.2 倍速く，NFS の 7.0 倍，Samba の 3.1 倍遅くなった．

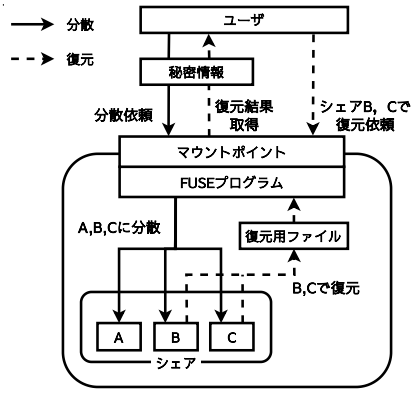

図 2　提案手法における分散/復元手順

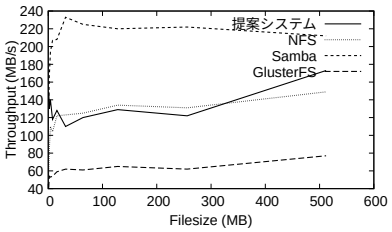

図 3　各ファイルシステムの write 性能

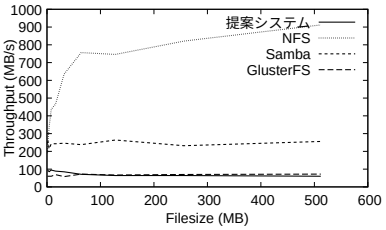

図 4　各ファイルシステムの read 性能

5　おわりに

本研究では，XOR による秘密分散法を用いて開発したファイルシステムを NFS，Samba，GlusterFS と比較実験を行い，write/read 性能を評価した．その結果，write 性能は NFS の 1.2 倍，GlusterFS の 2.2 倍速く，read 性能は GlusterFS の 1.2 倍速くなり，既存のファイルシステムに劣らない性能となった．ただし，本実験では既存のファイルシステムをローカル環境下で実験を行った．また，現在の提案システムはネットワーク上に展開することができないことから write/read 性能が実用範囲内に収まったと考えられる．今後，ネットワークを用いた実験を行う予定である．

参考文献

[1] Adi Shamir, “How to share a secret,” *Commun. ACM*, vol. 22, p. 612–613, nov 1979.

[2] “libfuse.” https://github.com/libfuse/libfuse.

[3] Jun Kurihara, Shinsaku Kiyomoto, Kazuhide Fukushima, and Toshiaki Tanaka, “On a fast (k, n)-threshold secret sharing scheme,” *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 91, pp. 2365–2378, sep 2008.