帯域・記憶領域共有による管理者不在のセキュア分散ファイル管理

1. 背景

2005 年 4 月より個人情報保護法が施行され，企業の情報管理が厳しく問われるようになったが，依然として重要情報の流出が後を絶たないのが現状である．

サーバ型情報共有では，データ及び管理権限が 1 箇所に集中しているため，サーバの管理者権限が奪取されるとすべての情報が流出してしまうという危険性がある．また，管理者はあらゆる権限を持っているため，自身が不正を行った場合の痕跡を消すことができるため，問題発生時の情報の管理責任が明確ではない．つまり，セキュリティが管理者の能力に依存している．

P2P 型情報共有では，特別なサーバを必要とせず，また絶対的な管理者が存在しないためコストの面では優れている．しかし多くは匿名もしくは不特定多数での情報共有を目的としている．管理者のない状況では限られたメンバーでの共有が困難である．さらに，情報発信者の特定がほぼ不可能であり，やはり問題発生時の情報の管理責任が明確ではない．

2. 目的

本プロジェクトでは，専門の管理者を雇えないような中小規模のユーザでも導入可能なレベルのコストで，安全に情報を共有する分散ファイル管理システムの開発を目的とする．まず，P2P 型システムにして特別な管理者を設けずコストを下げる．保存データはシステムによって分割し分散保存することによって,特定箇所への攻撃耐性を高める．各ファイルはユーザ毎の公開鍵・秘密鍵で暗号化することによって安全性を高めるとともに，その所有及び管理責任の所在を明確化する．さらに，限られたメンバーだけで柔軟な共有を可能にする．

3. 開発の内容

3.1. システム概要

図 1 に，開発したシステム JIGFS の概要を示す．JIGFS は，保存ファイルの副次的な情報を保存する仮想サーバと，これを利用しかつ保存先となるクライアント PC 群で構成される．JIGFS の動作は次のように行われる．

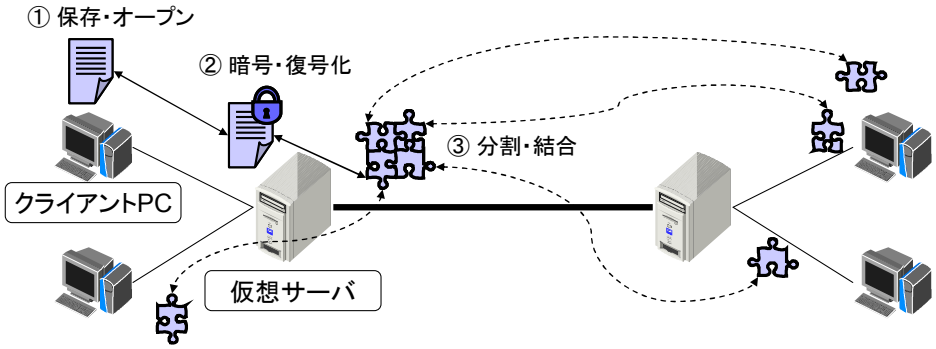


図 1：JIGFS の概要

1. クライアント PC 上にある専用ツールでファイルを保存する
2. 各ユーザの公開鍵ベースでファイルが暗号化され，仮想サーバ上へ転送される
3. 仮想サーバ上でファイルが分割され，ネットワークを越えて分散保存される
4. 保存したファイルへアクセスする際は，ファイル収集・結合・復号という逆の順序をたどる

分割の際は冗長性を持たせているため，復元の際にすべての断片が揃わなくても全体を復元可能である．また，保存先は通信帯域や PC 同士の起動時間の相性などを総合して決定しているため，障害耐性や運用上のパフォーマンスも考慮されている．

3.2. クライアント

3.2.1. Windows 版

図 2 に Windows 版クライアントを示す．画面左側には仮想サーバが提供するファイルシステムのディレクトリツリーが表示される．上部にはツールバーがあり，左から「上のディレクトリへ移動」「新規ファイル作成」「新規ディレクトリ作成」「ファイルアップロード（保存）」「リスト更新」となっている．カレントディレクトリ部へファイルをドラッグアンドドロップすれば，システムによってファイルが自動的に暗号化・分散保存される．ファイルアイコンをダブルクリックすれば，断片を収集・結合して復元が行われる．

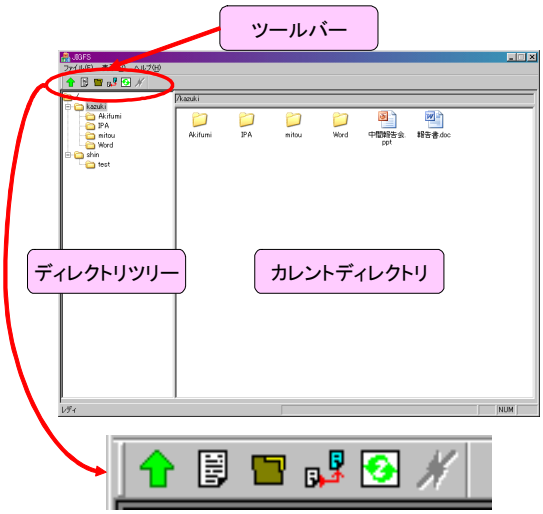
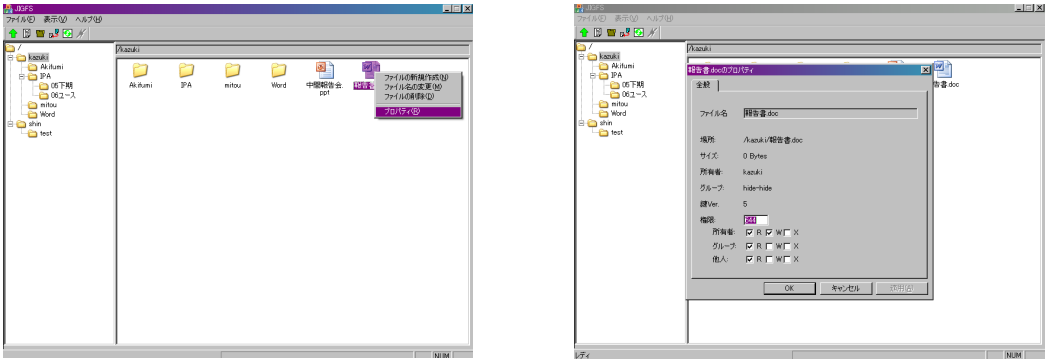


図 2：Windows 版クライアント

Windows 版クライアントにおける共有の様子を図 3 に示す．カレントディレクトリ部分のファイルを右クリックし，「プロパティ」を選択すると（図 3(a)），該当ファイルに対する様々な情報を表示できる．この中で「権限」フィールドを 8 進数の UNIX 形式，もしくはチェックボックス部分を変更すれば，当該ファイルのアクセス権限を変更可能である（図 3(b)）．また，ファイルの過去状態へのアクセス管理を実現するため，「鍵 Ver.」という鍵の更新回数が表示されている．各ユーザは，自身が保持しているバージョン以前の鍵で暗号化されたファイルにはアクセスすることができないようになっている．



(a) 右クリック時　　(b) アクセス権限設定

図 3：Windows 版ファイル共有

3.2.2. Linux 版

図 4 に Linux 版クライアントを示す．Linux 版は FUSE を用いてファイルシステムと透過型に実装した．このため，シェルからは JIGFS が通常のディレクトリと同じように見える．画面におけるディレクトリ “/mnt/jigfs” 以下はすべて，実際には仮想サーバがネットワークで提供している仮想的なファイルシステムである．このため，cd コマンドによるディレクトリ間の移動，ls コマンドによるファイル情報の閲覧，cp コマンドによるファイルコピーなどが，システムの存在を意識することなく可能となっている．

図 4：Linux 版クライアント

Linux 版クライアントにおける共有の様子を図 5 に示す．図 5(1) では Linux 版クライアントから本システム中で共有されているファイルにアクセスをし，less コマンドでその中身を確認しようとしたが失敗している．このファイルは実際には，ユーザが共有グループに参加する以前に暗号化されたファイルである．よって現在のシードから復号に必要な鍵を生成することができず，アクセスすることはできない．アクセスできない理由は，図 5(2) において専用のユーティリティを用いて鍵のバージョンを調べることができる．図 5(3) において，自分の鍵バージョンが不適格であることがわかる．

図 5：Linux 版ファイル共有

また，Linux 版には，JIGFS にアクセス可能なプロセスの種類を限定する機能がある．これにより，近年被害が著しい暴露型ウィルスのような意図しないプロセスからのファイルアクセスを制限できる．

3.3. 仮想サーバ

仮想サーバは，ファイル本体ではなく，ファイル名・サイズ・タイムスタンプといった副次的な情報のみを管理する．すべての情報は暗号化されているため，攻撃をしかけたとしても，どのファイルがどこへ保存されているかを推測することは困難である．また，仮想サーバのデータベースそのものも，本システム上に自動的にバックアップされる．万一サーバに障害が発生したとしても，初回起動時に作成したいくつかの情報を入力するだけで，分散保存されたバックアップファイルを自動的に取得し，直ちにもとの状態に復旧可能である．

仮想サーバは Linux 上で動作するデーモンとして実装されており，通常はバックグラウンド，デバッグモードではフォアグラウンドで動作する．データベースには MySQL を利用した．図 6 に，仮想サーバによって各クライアント PC 上に分散保存されたファイルの断片を示す．各ファイル断片はランダムな名前がつけられている上，サイズも統一されている．このため，断片からどのファイルの構成要素であるかを判断することは出来ない．

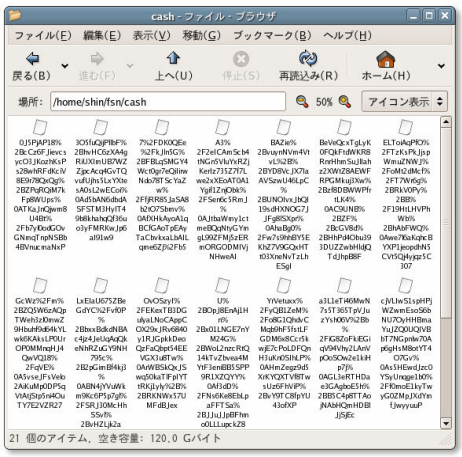


図 6：分散保存されたファイル断片

4. 従来技術との相違

開発成果と従来技術との顕著な差異を以下に列挙する.

1. 徹底的なセキュリティ
   - ◆ 分散保存により情報が一気に流出しない
   - ◆ 意図しないプロセスがファイルにアクセスできない
   - ◆ ファイルの断片サイズから元のファイルを推測できない
2. 管理責任の明確化
   - ◆ 自身のデータに公開鍵で指紋を残すことで,違法データ所有などの不正時において所有事実を否認できない
3. 複数組織間での実用性
   - ◆ 仮想サーバでクライアントに必要な計算・通信処理の負荷を軽減
   - ◆ 仮想サーバの自動バックアップと容易な復元

5. 期待される効果

本プロジェクトで開発した仮想サーバの機能がルータのような小型機に実装されれば,各家庭や管理者を雇用できない中小規模の業者でも安全に情報を共有することが可能になる.

6. 普及及び活用の見通し

動作実験からは,日常良く利用する 2MB 程度までのドキュメントファイルであれば 2 秒以内での書き込みが可能であり，十分実用に耐えうることがわかった．今後はより実際の利用に近い環境で Linux 版を長期間運用し，復元率やパフォーマンスなどの詳細なデータを収集する．その結果を元に，シェアが見込まれる Windows 版を開発して行きたい.

7. 開発者名

井上　亮文（東京工科大学）
石原　礼男（有限会社エムエル）
大津　一樹（東京工科大学）
鹿島　隆行（東京工科大学）
手塚　伸　（東京工科大学）