ファイル盗難リスク分散のためのP2Pアプリケーションの開発*

永井 健太郎† 山崎 航† 平石 広典† 溝口 文雄†
東京理科大学 理工学部 経営工学科‡

1 はじめに

近年, インターネットの普及などによりネットワークに接続されている端末は飛躍的に増加している. ネットワークの普及によりファイル共有などが容易になった一方でネットワークに接続されている端末は第三者によってアクセスされやすく, その結果として端末内のファイルが外部へ漏洩する危険性が高まっている. この原因としてはネットワークセキュリティの問題に加え, ファイル管理の問題があると考えられる. 現在ファイルは一つのデータ塊として単一の場所に配置されている. これはファイル共有の観点からすると非常に有用であるが, 逆に侵入者や悪意を持った内部犯行者からみてもファイルを取得しやすい状況だといえる. ファイルの漏洩を防ぐという観点からするとこれは非常に不利である.

また, ファイル漏洩への対策としてはファイルは暗号化するという手段がよくとられる. これによってファイルを取得されても鍵がわからなければ中身を閲覧できない. 従来はこれでも問題はなかったが, 近年の暗号解読に特化されたハードウェア [1] の出現や膨大な計算力を持つ分散コンピューティングネットワーク [2] の台頭により, すべての鍵を網羅的に試し, 暗号解読を試みる総当たり攻撃が無視できないものとなってきている.

そこで本研究ではファイルを容易に漏洩するのを防ぎ, また, 漏洩したとしても総当たり攻撃の適用を困難にするための手法を提案し, 実際にアプリケーションの開発を行なった. このアプリケーションによりファイル漏洩の危険性を低減できることを示す.

*Development of P2P application to achieve risk distribution against file disclosure

†Kentaro Nagai, Wataru Yamazaki, Hironori Hiraishi, Fumio Mizoguchi

‡Industrial Admin., Facl. of Science, Tokyo Univ. of Science

2 設計および実装

本研究では前節で述べたような問題の解決策のひとつとして, ファイルの分割分散配置をする方法を提案する. ここではアプリケーションの設計および得られる効果について説明していく. なお, 開発はすべて Java 言語で行なった.

2.1 ファイルの分割/暗号化

ファイル漏洩リスク低減のための本研究の特徴の 1 つがファイルの分割である. ファイルはすべて一定のサイズ (8kb) に分割し, ファイルサイズという特徴を隠蔽する. さらに, 他の端末に分散配置しても中身を確認できないように, 分割されたファイルはそれぞれを暗号化される. この暗号化の際, 分割されたファイルは前後の分割されたファイルと関連を持たせるようにし, 分割されたファイルの 1 つを正しく復号するには前後の分割ファイルも必要となる. したがって, 分割ファイルを総当たり攻撃をすることは可能だが, 正しい順序で正しい分割ファイルを対象に行なわなければ解読はできない. また, 分割されたファイルを取得することができたとしても, その分割ファイルがどのファイルの断片なのか, またどの順序で結合すると元のファイルを復元できるのかは推測できない. 暗号アルゴリズムには DES を利用している.

2.2 分割ファイルの名前付け

本研究では 1 つのファイルから複数の分割ファイルが生成される. この分割ファイルはネットワークを使って他者の端末に配置されるので, 分割ファイルの名前から元のファイルが推測されないよう, 名前を無個性化する必要がある. 具体的には名前は分割ファイルのデータから計算されるハッシュ値を元に ID を生成しこれを名前として利用する.

また, 分割したファイルの ID を衝突させないためには, 精度の高いハッシュ関数を利用する必要があるが, 現状では MD5(64bit) を利用している. 実際のファイル名は分割ファイルのどこにも現れず, 侵入者などにはどのようなファイルが分散配置されているかはわからない.

2.3 分割ファイルの分散配置

分割, 暗号化されたファイルはネットワークを介して他の端末に分散配置される. ファイルを分散配置させるには各端末にファイルを格納, 取得できるインターフェイス, つまりサーバとクライアントの機能が必要である. この点は Peer-to-Peer(P2P) 型アプリケーションとして実装することで実現した. これによりファイルが一箇所に集中することを回避できる分散配置が可能になった. P2P アプリケーションは TCP/IP コネクションで仮想的なネットワークを構成する. 本研究での分割ファイルの分散の手段としてこの仮想的なネットワークを利用した.

2.4 分割ファイルの収集

元のファイルを復元するには分散配置したすべての必要な分割ファイルを収集しなければならない. その収集の手順をしめす (図 1). まず, ファイルを復元するためには分散時に指定した格納名とパスワードを知っている必要がある (1). アプリケーションが元のファイルを構成するのに必要である最初の分割ファイルの ID を格納名とパスワードからわりだし (2), それを P2P ネットワーク内から検索 (3), 取得する (4). そして, 取得した分割ファイルを実際にパスワードを用いて復号する (5). 復号したファイルには次の分割ファイルの ID が記されているので, この ID を元にアプリケーションが次の分割ファイルをネットワークから取得する. この操作を終端に辿りつくまで繰り返す (6). 集めた全てのファイルを結合して, 1 つのファイルにし (7), ユーザにファイルを渡す (8).

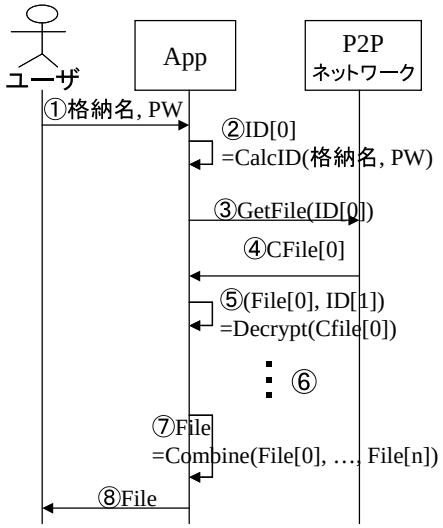

図 1: ファイルの収集

3 まとめ

本研究ではファイル漏洩リスクの低減するためのファイルを分割分散配置するアプリケーションの開発を行った. ファイルを分割した上で P2P ネットワークを利用して分散して配置し, また P2P ネットワーク内に散らばった分割ファイルから元のファイルを復元する仕組みを持っている点が, CODA[3] のような分散型ファイルストレージシステムや, Freenet[4] のようなファイル共有アプリケーションとの違いであると考える. このアプリケーションを利用することで, ファイルを容易に分割分散配置を実現できるようになり, 結果として, ファイルの漏洩リスクの低減する容易な手段を提供できることを示した.

参考文献

[1] M. J. Wiener, “Efficient DES Key Search”, *IEEE Computer Society Press*, pp.31-79., 1996

[2] distributed.net, http://www.distributed.net/

[3] J. J. Kistler and M. Satyanarayanan, “Disconnected Operation in the Coda File System”, *Thirteenth ACM Symposium on Operating Systems Principles*, 1992

[4] Freenet, http://freenet.sourceforge.net/