

インスタントメッセンジャを用いた ネットワークファイルシステムの実現

金子 直 矢[†] 新城 靖^{††} 佐藤 聡^{††}
中井 央^{†††} 板野 肯 三^{††}

1. はじめに

今日、さまざまな方法でネットワーク上のリモートホストのファイルへのアクセスが行われている。その例としてここではクライアントサーバモデルに基づいてサーバを用いる方法と、インスタントメッセンジャ(以下 IM)を用いる方法の2つを挙げる。

前者として NFS(Network File System), WebDAV がある。これらをインターネット上で実現する場合、サーバを公開することにより起こりうる、ネットワークを介した攻撃についての対策を講じる必要がある。後者はユーザによる公開サーバが不要であること、ユーザの身元確認を IM が行っていること、また一般に IM ではその通信路が暗号化されていることなどの利点がある。その一方で、専用のインタフェースを通したファイルの転送という方式でのアクセスに限定される。このため、ファイルブラウザ等を用いた通信相手のユーザによる任意の選択・操作ができないという欠点がある。

そこで、本研究では以上にあげた双方の利点を持つネットワークファイルシステムを設計・実装する。つまり、本研究のシステムでは前者における利点である任意のアプリケーションを用いての通信相手のファイルへのアクセスを可能にする。また IM を用いることにより、ユーザによる公開サーバを必要とせず、またアクセス制御を行う際に通信相手の身元確認ができる。さらに暗号化された通信路を利用する IM を用いることでこのアクセスをセキュアに行うことが可能となる。

2. 実 装

本研究では Windows ホストにおいて、IM として Skype を用い、そのネットワーク上で RPC を行う仕組みを実装する。また Windows で使用可能な

FUSE(Filesystem in USER space) として Dokan¹⁾を用いる。これによりユーザアプリケーションに対するインタフェースを提供する。これらをそれぞれ図 1にある SkypeRPC, SkypeFSServer, SkypeFSClient モジュールとして作成し、SkypeFS と呼ぶシステムの実装を行っている。これに加えて Skype の機能を用いたアクセス制御を可能にする仕組みを追加する。

2.1 Skype を利用した RPC

Skype は、独自のネットワークを構成しその上で各ホストがチャットメッセージや映像・音声データを送受信している²⁾。これらの Skype のネットワークやプログラムの動作を外部プログラムから操作可能とする Skype API が公開されている³⁾。この API は COM(Component Object Model) コンポーネントである Skype4COM, あるいは Java などのラッパクラスからも利用可能である。

本研究では Skype4COM を通して Skype AP2AP (Application to Application) 通信のための API を利用する。これを用いることにより通信を行うそれぞれのホスト上の Skype 間でのデータの送受信などの操作が可能になる。

この API を用いて RPC による通信を可能にする仕組みを実装する。これを SkypeRPC モジュール(図 1)と呼ぶ。本研究で実装する RPC の通信はメソッド名とデータのセットからなるメッセージを送受信することで行われる。1回の通信は必ず一方のホストの要求メッセージともう一方のホストからの応答メッセージの対からなり、前者のホストでは要求メッセージを送った後、応答が返ってくるまでは動作がブロックされる。

要求を受け取るホストでは、あらかじめ手続き呼び出しが発生した際のコールバックメソッドを登録しておく。RPC を行うホストは呼び出す手続き名をメソッド名、手続きに与える引数をデータとしてメッセージを作成し送信する。これを受け取ったホストは、指定された名前でのコールバックメソッドを呼び出し、処理を

[†] 筑波大学第三学群情報学類

^{††} 筑波大学システム情報工学研究科

^{†††} 筑波大学図書館情報メディア研究科

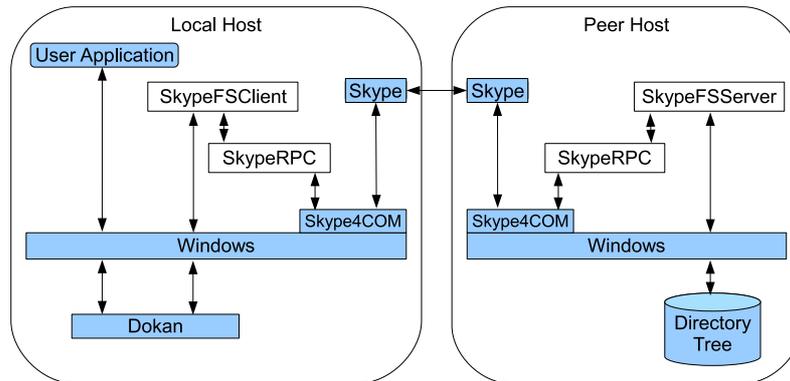


図 1 SkypeFS の構成

行う。この結果をメッセージのデータとしてメッセージを作成し送り返すことにより RPC を実現している。

2.2 Dokan によるファイルシステム構築

本研究ではリモートファイルシステムにアクセスするためのインタフェースとして、Dokan を用いて作成したファイルシステムを使用する。これにより Windows で動作する任意のアプリケーションからの通信相手のファイルシステムへのアクセスが可能となる。

Dokan 上のファイルあるいはディレクトリに対する操作（イベント）が発生する場合、Windows から Dokan を通して操作に対応した特定のコールバックメソッド（例:open,read）がそれぞれに必要な引数とともに呼び出される。

本研究では、このコールバックメソッドの処理として、ユーザが行った操作をリモートホストに対して、2.1 節で述べた SkypeRPC を用いて転送する。このときコールバックメソッドの名前をメソッド名、引数をデータとしてメッセージを作成する。このメッセージを受けとったホストはメッセージ中のメソッド名に従って登録されたコールバックメソッドを呼び出すことにより実際にローカルのファイルシステムを操作する。リモートホストはこの処理の結果をデータとして応答メッセージを作成しこれを送信する。この実装により Dokan を用いたリモートホストへの操作を実現する。

2.3 Skype の機能を用いたアクセス制御

Skype の機能を用いたアクセス制御としてそれぞれ、ユーザ名、コンタクトリスト、グループチャットを用いる方法を提供する。Skype の AP2AP 通信では通信を行っている相手のユーザ名を取得可能である。これにより通信を許可するユーザのリストを作成することでアクセス制御が可能となる。

このユーザ単位でのアクセス制御に加えて、本研究ではコンタクトリスト、グループチャット単位でのアク

セス制御を行う仕組みを実装する。コンタクトリスト、グループチャットに所属しているユーザの名前も取得可能であるため、複数ユーザに対しての設定が容易になる。加えて招待制であるグループチャットの機能を用いることにより、一時的なグループを作成し、これに所属するユーザにのみアクセスを許可することも可能である。

ここでは Skype API を通してユーザ、コンタクトリスト、現在開かれているチャットなどの情報を取得し、これに基づいたアクセス制御を可能にする機能を追加する。

3. おわりに

本稿ではインスタントメッセージングを用いたネットワークファイルシステムを提案した。この方法ではインスタントメッセージングのネットワーク上で通信を行うことによりファイルアクセスをセキュアに行うことができる。今後は実装を完了させ、機能と性能の評価を行う。

参考文献

- 1) Hiroki Asakawa: “Dokan Ruby Binding 0.1.4,” Dec. 2007. <http://dokan-dev.net/> accessed 2008-10-28
- 2) S. A. Baset, H. Schulzrinne: “An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol,” *IEEE Infocom'06*, Apr. 2006.
- 3) Skype API: “Skype Public API 3.6 Reference Guide,” Oct. 2007. <https://developer.skype.com/Docs/ApiDoc> accessed 2008-10-30