モバイルアドホックネットワークにおけるストリームデータ配信時の認証手法

安藤 和香　(指導教員：小口 正人)

1　はじめに

近年, 固定ネットワークにおけるストリームデータの配信は頻繁に行われているが,MANET(Mobile Adhoc NETwork) においてはまだあまり一般的ではない. サーバが介在せず, 端末同士が情報をやりとりする MANET において, ストリームデータの配信が安全に行える環境があれば, 多岐に渡るアプリケーションに利用可能と考えられる. そこで本研究では MANET におけるストリームデータ配信時の認証を実現する.

2　MANET におけるストリームデータ

サーバに蓄積されたマルチメディアデータをネットワーク経由でダウンロードしながら順次利用する技術をストリーミングという. ストリームデータはデータ量が多く, 動的に内容が変化し, 途切れず流れ続けるという特徴があげられる. そのため, 端末相互間のリンクが不確実な MANET 上で取り扱うのは困難とされている.

また MANET では各ノードは自由にネットワークに参加および離脱することが可能なだけでなく, 他のノードを中継して通信エリアを拡大するマルチホップ通信なども可能である. そのため常に不特定多数のノードがネットワークに存在し, 送信者のなりすましやデータの改ざん, 不正利用等が容易に行えてしまうという問題点があるため,MANET における認証手法を検討する必要がある.

3　公開鍵暗号方式

公開鍵暗号方式ではまず, 互いに関連性のある「秘密鍵」と「公開鍵」というペアの鍵を作成する. 公開鍵から秘密鍵を作ることは不可能であり, 秘密鍵は自分で大切に保管する. 誰でも公開鍵を使ってデータを暗号化することが可能であるが, 公開鍵で暗号化したデータは, その鍵に対応する秘密鍵でのみ復号可能となっている. 公開鍵暗号方式を用いる利点として, 第 1 に「認証」「署名」「検証」の３つの機能が備わっている点があげられる. 第 2 に鍵の管理が容易で安全性が高いという点があげられる. 公開鍵暗号方式を用いれば, 他人が偽造することが不可能なデータ転送を実現できる.

4　提案方式

MANET において, どのノードも安全にストリームデータ配信および受信可能な環境を実現する. 図 1 に示したように後述する JXTA を用いてピアの発見や接続を行い, 認証 Application を起動させる. 相手を正しく認証することができたら, UDP/IP 接続のストリームアプリケーションを呼び出し, ストリームデータの配信と受信を開始する. 本研究では, そのような動作を制御する図 1 の JXTA Application 部分を作成し, システム全体を構築した.

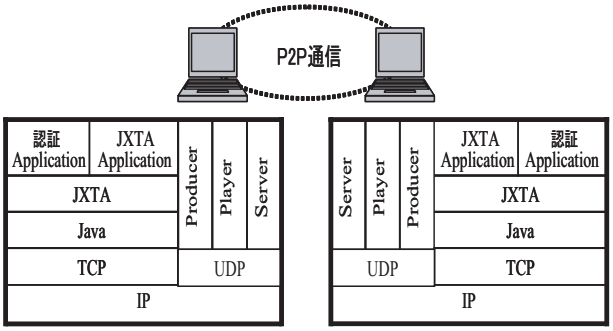

図 1: 提案方式

5　実験環境

本研究では Sun Microsystems 社が開発した P2P 型アプリケーションを容易に構築するためのプラットフォームである JXTA を使用した. ピア同士の自由な通信を実現し, 通信経路を自動的に決定してくれるため, MANET の端末相互間のリンクに有効である.

またストリームには, さまざまなメディアデータに対応できる REALNETWORKS 社の Helix Server を使用した. これは有線, 無線ネットワーク環境のクライアントに対し高品質なメディア配信を可能にするものである. メディアをストリーム配信および受信するために, 同じく REALNETWORKS 社の Real Player,Real Producer を使用した. 本研究の実験環境を図 2 に示す.

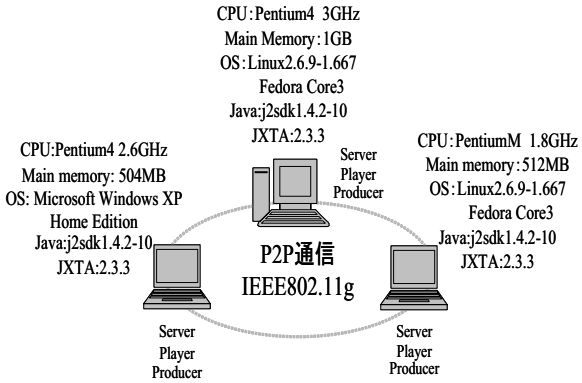

図 2: 実験環境

6　実行結果 1:公開鍵暗号を用いた認証

MANET において公開鍵暗号方式を利用した認証プログラムの実行結果を図 3 に示す. まず送信側でメッセージを秘密鍵で暗号化し, 署名を作成する. 次にメッセージと署名を JXTA の出力パイプから送る. 続いて受信側でメッセージと署名を入力パイプで受け取り, 署名を公開鍵で復号しメッセージを検証する.

図 3: 認証の実行:受信側

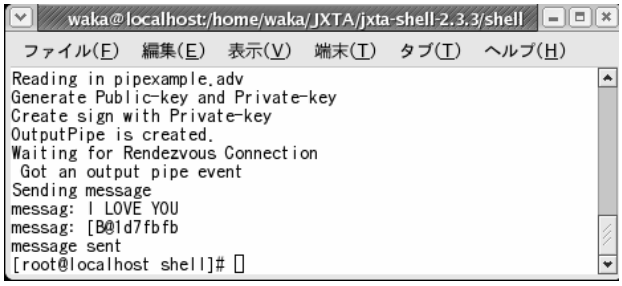

図 4: 認証の実行:送信側

7 実行結果 2:ストリームデータの配信

Real Producer を使用して,Helix Server で配信可能なファイルを作成した. オンデマンド配信ではメディアデータをエンコードし一旦ファイルにしてサーバに格納する. 一方リアルタイム配信ではメディアデータをリアルタイムエンコードしてサーバへ転送する.

図 4 にオンデマンド配信, 図 5 にリアルタイム配信実行時の帯域幅の変化を示す. オンデマンド配信では帯域幅が一定値となっているのに対し, リアルタイム配信においては, メディアデータの変換負荷により帯域幅が変化していることがわかる.

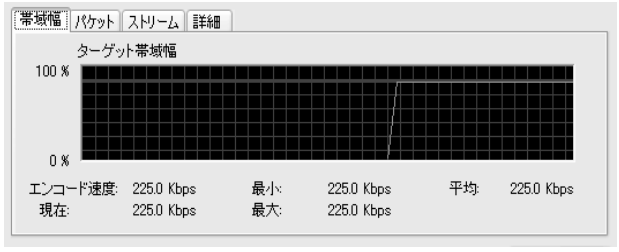

図 5: オンデマンド実行結果

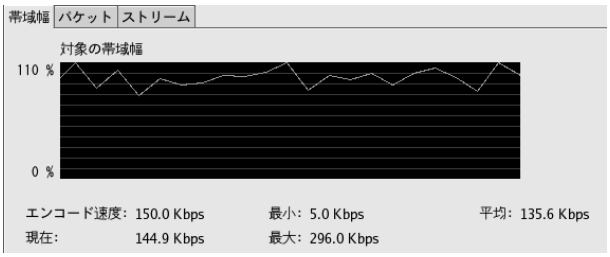

図 6: リアルタイム実行結果

8 実行結果 3:認証後ストリームデータ配信

実行結果 1 と 2 を用いて, 認証 Application の完了後, ストリーム処理が起動する JXTA Application を構築した. 相手を正しく認証したら server か player を選択して起動し, ストリームデータの配信もしくは受信を開始する. 図 6 と図 7 はオンデマンド配信の実行の様子を示している. すべてのノードにおいて server,player ともに選択, 起動可能なこと, またリアルタイム配信も実行可能なことを確認した.

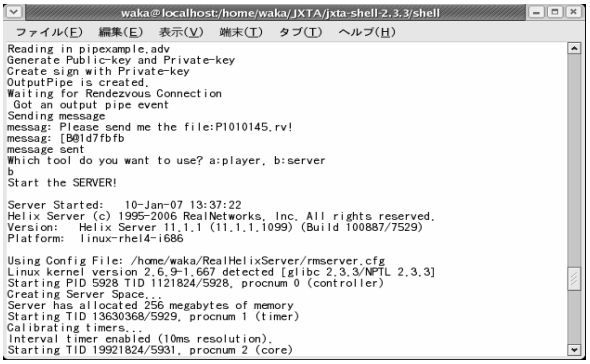

図 7: 認証後 server を起動

図 8: 認証後 player を起動

9 まとめと今後の課題

本研究では,MANET 上のどのノードも安全にストリーム配信および受信することが可能な環境の構築を実現した. オンデマンド配信だけでなく, リアルタイム配信も可能になったことから, さまざまなサービスに応用することができると考えられる. 今後は server と producer の統合, またユーザインタフェースの簡易化を実現したい.

参考文献

[1] Jxta Programmer's Guide
www.jxta.org/docs/

[2] 柴田芳樹, プログラミング言語 Java 第 3 版 , 株式会社ピアソン・エデュケーション

[3] 小原奈緒子, 小口正人:"MANET における公開鍵暗号方式を用いた階層型認証システムの提案と実装" DEWS2006-2Bi8, 2006 年 3 月

[4] 安藤和香, 小口正人:"MANET におけるストリームデータ送受信時の認証手法", 情報処理学会第 69 回全国大会,2007 年 3 月発表予定.