

# MANET におけるストリームデータ送受信時の認証手法

安藤 和香†

小口 正人†

†お茶の水女子大学 理学部情報科学科

## 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送受信は頻繁に行われているが, MANET (Mobile Ad-hoc Network) においてはまだまだ一般的ではない。サーバが存在せず, 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送受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 MANET におけるストリームデータ送受信時の認証を実現する。

## 2. MANET におけるストリームデータ

サーバにある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, 離脱することが可能だけでなく, 他のノードを中継して通信エリアを拡大することも可能である。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っ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を不可能にするデータ転送を実現できる。

A study on authentication method for transmission and reception of stream data on Mobile Ad-hoc Networks

† Waka Ando, Masato Oguchi

Department of Information Sciences, Ochanomizu University (†)

## 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, 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。

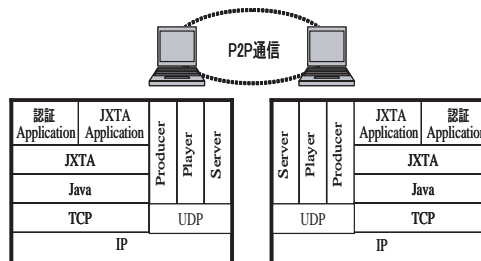


図 1: 提案方式

## 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よう 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高品質なメディア配信を可能にするために構築されたものである。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。

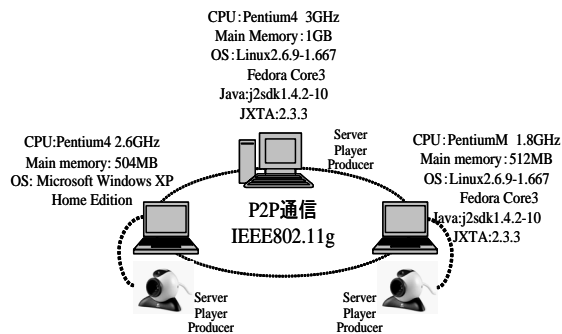
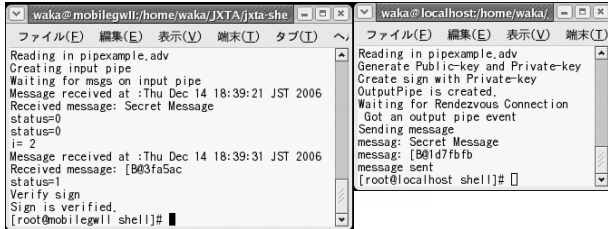


図 2: 実験環境

## 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。



```
waka@mobilegwll/home/waka/JXTA/jxta-she...
Reading in pipexample.adv
Creating input pipe
Waiting for msgs on input pipe
Message received at :Thu Dec 14 18:39:21 JST 2006
Received message: Secret Message
status=0
status=0
i= 2
Message received at :Thu Dec 14 18:39:31 JST 2006
Received message: [083fa5ac
status=1
Verify sign
Sign is verified.
[root@mobilegwll shell]#
```

図 3: MANET における認証の実行

## 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。これに対して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。

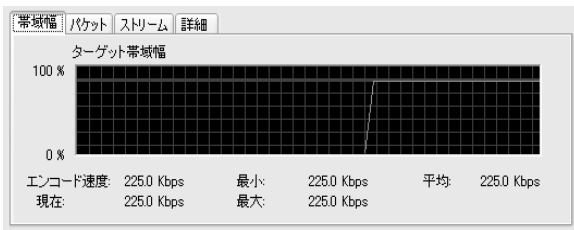


図 4: オンデマンド配信実行結果

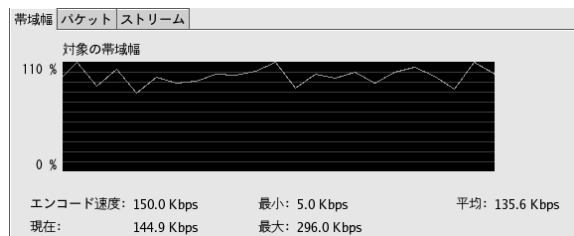
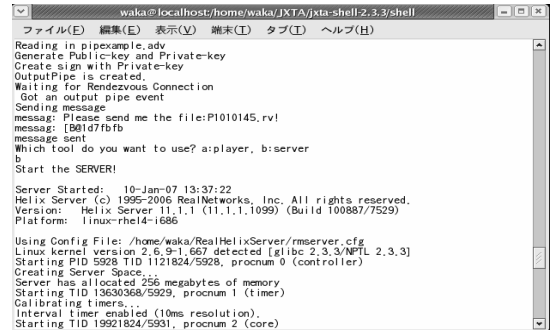


図 5: リアルタイム配信実行結果

## 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,JXTA Application が起動する環境を構築した。相

手を正しく認証したら server か player を選択して起動し, オンデマンド配信もしくは受信を開始する。すべてのノードにおいて server,player とともに選択, 起動可能なことを確認した。



```
waka@localhost/home/waka/JXTA/jxta-shell-2.3.3/shell
Reading in pipexample.adv
Generate Public-key and Private-key
Create sign with Private-key
OutputPipe is created.
Waiting for Rendezvous Connection
Got an output pipe event
Sending message
message: Please send me the file:P1010145.rv!
message: [081d7fbfb
message sent
Which tool do you want to use? a:player, b:server
Start the SERVER!
Server Started: 10-Jan-07 13:37:22
Helix Server (c) 1995-2006 RealNetworks, Inc. All rights reserved.
Version: Helix Server 11.1.1 (11.1.1.1099) (Build 100887/7529)
Platform: linux-rhel4-i686
Using Config File: /home/waka/RealHelixServer/mserver.cfg
Linux kernel version 2.6.9-1.667 detected [glibc 2.3.3/NFPL 2.3.3]
Starting PID 5928 TID 1121824/5928, procnum 0 (controller)
Creating Server Space...
Server has allocated 256 megabytes of memory
Starting TID 13630368/5929, procnum 1 (timer)
Calibrating timers.
Interval timer enabled (10ms resolution).
Starting TID 19921824/5931, procnum 2 (core)
```

図 6: 認証後 server を起動

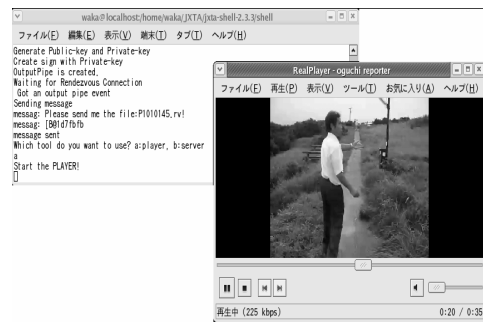


図 7: 認証後 player を起動

## 9. 今後の課題

本研究においては,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。今後は提案方式を発展させ,producer も使用することで,リアルタイム配信も可能にする。また,ユーザインタフェースを簡易化し,さらに新たなサービスを考え実装する。

## 参考文献

- [1] Linux によるストリーミング入門  
<http://web-old.turbolinux.co.jp/world/library/features/>
- [2] Jxta Programmer's Guide  
[www.jxta.org/docs/](http://www.jxta.org/docs/)
- [3] 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
- [4] 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