ICカードを利用したピア・ツー・ピアによるコミュニケーションプラットフォームの提案

小宅 宏明, 菅原 陽子, 宮地 玲奈, 岡田 謙一 慶應義塾大学院 理工学研究科

近年，常時接続環境の整備に伴って，非対面同期型協調作業を行う機会が増してきている．本稿ではピア・ツー・ピア型のネットワークを用いて協調作業のためのグループを構築し，またグループ構築に必要な情報をICカードによって携帯することによって，ユーザの移動に対応した安全なコミュニケーションプラットフォームを実現する方法を提案した．

A Proposal for Peer-to-Peer Based Communication Platform Using IC Card

Hiroaki Ohya, Yoko Sugawara, Reina Miyaji, Ken-ichi Okada
Faculty of Science and Technology,Keio University

In recent years, we have increasing opportunities of non-face-to-face syncronous collaboration with spread of regular connection environment. In this paper, we propose a method to construct a new platform of communication that is non-terminal-dependent and secure by constructing collaboration groups using peer-to-peer network and store nessessary infomation in IC card.

1. はじめに

近年，常時接続環境の整備に伴い，遠隔地で会議やゲームなどの非対面同期型協調作業を行う機会が増えてきた．ネットワークに接続された端末がどこにでもある状況においては「いつでも」「すぐに」他のユーザと協調作業を行えるようなシステムが求められている．

また，ユーザが利用する端末について考えてみると，同一のユーザでも自宅や職場，出先など様々な場所においてその場にある端末を利用することが多い．利用する端末は毎回異なるかも知れず，その端末が属している環境 (IPアドレスやドメイン，ファイヤーウォールの有無など) もそれに伴って変わってゆく．

協調作業のためのシステムにおいてもこの点を考慮し，ユーザがどの端末を使用しても同じように利用できることが望ましい．

以上のことから，非対面同期型コミュニケーションプラットフォームに求められる条件は次のようなものになると思われる．

- アドホックにグループを構築し，協調作業を行えること．
- 端末と端末の属している環境に依存せず，相互運用可能であること．
- 安全であること．

本稿ではこれらの条件を満たすコミュニケーションプラットフォームを構築する方法を提案する．

2. 非対面同期型プラットフォーム

非対面同期型コラボレーションを行ためのプラットフォームとしては，ICQやIRCなどのクライアント/サーバ型のプラットフォームとMicrosoft社のNetMeetingなどのピア・ツー・ピア型のプラットフォームが存在する．

2.1 クライアント/サーバ型

クライアント/サーバ型のアーキテクチャはリソースがサーバ上で一括管理でき，リソースの同期をとる必要がないため，大規模な協調作業に適している．しかし，クライアントとサーバの両方でアプリケーションに対応していなければならず，またサーバ上にユーザアカウントがサーバ上に必要であるた

め，アドホックな協調作業には向いていない．

2.2 ピア・ツー・ピア型

ピア・ツー・ピア (以下 P2P) とは，対等な機能を持った端末同士が相互に接続されたネットワーク形態を指す．P2P では各端末がクライアント/サーバシステムにおけるクライアントとサーバの両方の機能を持っているため，専用のサーバを必要と必要とせず，ネットワーク構成の動的な変化に対応できる．従って，P2P はアドホックなグループ構築に適していると言える．

P2P によるコラボレーションプラットフォームとしては，Microsoft 社の NetMeeting[1] が有名である．NetMeeting は Windows 上で動作し，NetMeeting 上では音声チャット，ファイル共有，実行中のアプリケーションの共有などを行うことができる．NetMeeting は Windows 上でコラボレーションを行うには有用であるが，IP アドレスで相手を指定して通信を行うため，通信を行うために前もって相手の IP アドレスを知っている必要がある他，異なるファイヤーウォール内にあるピア同士では通信ができないという問題がある．

2.2.1 ピア・ツー・ピアにおけるセキュリティ

P2P による通信に関しては，以前からセキュリティの脆弱性が指摘されてきた．これは，P2P ではクライアント/サーバ型のシステムのような中央集権的な管理が不可能であり，また P2P における通信が無関係のピアを経由してルーティングされるために盗聴と改竄の危険性を持っているからである．

P2P で通信を行う際に満たされるべきセキュリティの条件として，以下の 5 つが挙げられる．

認証 (Authentication) あるユーザが実際にそのユーザ自身が宣言した人物であることを保証する．

許可 (Authorization) 送り手が，あるメッセージの送り手としての認可を受けていることを保証する．

機密性 (Confidentiality) 通信内容が，それを知る権限のない個人に対して明かされてしまわないことを保証する．

データの完全性 (Data Integrity) 通信の内容が，予期せぬ理由によってであれ，あるいは，意図的な仕方によってであれ，移動中に変更を加えられはしなかったことを保証する．

P2P でセキュアな通信を開始する際の手順を説明する．過去にセキュアな通信を行ったことのあるピア同士の場合，安全な通信を開始するまでの手順は次のようになる．

(1) ピア A はピア B に接続し，その身元を公表する．
(2) ピア B はピア A が自分自身を認証することを要求する．
(3) ピア A はピア B に対して自分自身を認証する．
(4) ピア A は，ピア B が自分自身を認証することを要求する．
(5) ピア A はピア B に対して自分自身を認証する．
(6) ピア B はピア A に特権を割り当てて，ピア A が特定のリソースにアクセスするのを許可する．
(7) 2 つのピアは両者間のチャネル接続を暗号化するための準備ができる．

初めて通信を行うピア同士の認証は，双方のピアにとって信用できる第三のピアを経由することによって行われる．認証を行いたいピアをピア A とピア B，ピア A とピア B にとって信用できる第三のピアをピア C とすると，安全な通信を開始する手順は次のようになる．

(1) ピア A がピア C とセキュアな通信を開始する．ピア C は，ピア B を認証するために必要な情報 (ピア A の公開鍵など) をピア A に与える．
(2) ピア B はピア C とセキュアな通信を開始し，同様の操作を実行する．
(3) ピア A とピア B はピア C から取得した情報を元に相互認証を行う．
(4) 2 つのピアは両者間のチャネル接続を暗号化するための準備ができる．

ユーザが特定の端末を利用することが想定されていない場合には，認証に上記の方法を用いることはできない．なぜならば，端末はそのユーザが初めて使用するものであり，過去の通信に依存するような前提を設けることはできないからである．つまり，認証を必要としているピア同士は過去にセキュアな通信を行ったことはなく，また双方のピアにとって信用できる第三のピアも存在しない．この場合，何らかの別な形で安全性を保証する手段が必要になる．

3. システムの概要

本稿ではピア・ツー・ピアネットワーク上においてICカードを利用することにより，利用する端末に依存せず，かつ安全なコミュニケーションを可能にする機構を提案する．

ユーザはICカードを携帯し，移動先でその場にある端末を利用して他のユーザと協調作業を行う．ユーザは次のように本システムを利用する．

(1) ユーザはICカードを端末のICカードリーダライタにセットすると，アクセス可能なユーザの一覧が表示される．
(2) ユーザが一覧の中からコミュニケーションを行いたい相手を選択する．
(3) 本システムが選択されたユーザをネットワーク上で検索し，発見されれば相手に接続要求を出す．
(4) 相手によって接続要求が許可されると，相手側のシステムと相互認証を行う．
(5) 認証が成功すればグループを構築し，グループ内で協調作業が可能になる．

3.1 システム構成

本システムは次の3つの部分から構成される．

ベース部分　P2Pによる通信を行う．

サービス部分　他のユーザの発見と認証，通信の暗号化を行う．

アプリケーション部分　ユーザが協調作業を行うアプリケーション．

アプリケーション部分はサービス部分，およびベース部分とは独立して機能する．例えばユーザが利用するアプリケーションはグループ構築の方法とは無関係に動作する．

3.2 他のユーザの検索

ユーザが利用する端末は常に同一とは限らないため，IPアドレスではなく各ユーザに固有のIDでユーザを識別する．

検索の方法としては，P2Pネットワーク内で検索を行うことによって限られた範囲内で他のユーザを検索することは可能であるが，この方法では検索に時間がかかる上，必ずしも探しているユーザが発見できるとは限らない．

そこで本提案では，Rendezvous Pointという特殊なピアを用いることにより，コミュニケーションを行いたいユーザを検索する．Rendezvous Pointとは一種の「待ち合わせ場所」であり，ユーザ一人につき一つのRendezvous Pointが存在する．Rendezvous Pointは本システムで利用しているP2Pプラットフォーム”JXTA”で提供されている機能である．例えば，ユーザAがユーザBを探す場合は，ユーザAはユーザBのRendezvous Pointに接続し，そのRendezvous Point上でユーザBを検索する．この時ユーザBが自分のRendezvous Pointに接続していなければ，ユーザAはユーザBを発見できずに終わることになる．

Rendezvous Pointとなるピアはファイヤーウォールの内部にあってはならず，また常に同じIPアドレスをもっている必要がある．これらの条件を満たすRendezvous Pointを利用することで，コミュニケーションをとりたいユーザの利用する端末のIPアドレスを知らなくても，そのユーザを検索することが可能となる．

3.3 情報の携帯

各ユーザは自分のIDとRendezvous Point，それから認証のために必要な情報を持っている．他のユーザとコミュニケーションを開始するには，そのユーザに関するこれらの情報が必要になる．ユーザが移動先でその場にある端末を用いてコミュニケーションを行うには，これらの情報を携帯する必要があり，セキュリティに関する情報は外部から読めないようになっていなければならない．

本提案においては，ICカードによって情報を携帯する．ICカードの情報は外部から読み込めないようになっているため，セキュリティに関する情報が外部に洩れる危険性はない．

3.4 セキュリティ

本提案では認証情報として非対称鍵暗号と対称鍵暗号を使う2種類のモデルを用いる．両者はなるべく違いが少なくなるように考慮されており，どちらの方法を用いても安全に非対面同期型の協調作業を行うことができる．両者は排他的なものではなく，協調作業を行う相手に応じて同一のシステム上で併用することが可能である．例えば，日頃公開鍵サーバを利用している相手には非対称鍵暗号による方法を利用し，そうでない相手には対称鍵暗号による方法を利用する，といった使い分けが考えられる．

ここでの安全性の定義については，2.2.1で挙げたようなセキュリティ上の条件を満たすこととする

が，このうちの「許可」については本システム上で動作するアプリケーションによって管理するものとする．

3.4.1 対称鍵を用いる場合

対称鍵を用いる場合には非対称鍵を用いる場合と比べて，より簡単にシステムの構築を行うことができるが，共通鍵を安全に共有するための仕組みが必要になる．最も安全な方法としては，鍵を共有するユーザが物理的に集まって同一の端末から鍵を供給する方法が考えられる．2.2.1 の条件を満たすために次のような方法を用いる．

認証 チャレンジ・レスポンス方式

$Response = h(Challange||SecretKey)$ ☆ ☆☆

チャレンジ・レスポンス方式を用いることにより，対称鍵をカードから端末に読み込むことなく認証を行うことができる．

機密性 対称鍵暗号を用いてセッションキーを共有し，セッションキーを用いてエンドツーエンドで共通鍵暗号による暗号化を行う．セッションキーの生成はカード内で行い，通信内容の暗号化はカード内で生成されたセッションキーを用いて端末上で行う．

データの完全性 メッセージを MAC(Message Authentication Code) に付加する．

$MAC = h(Message||SecretKey)$

3.4.2 非対称鍵を用いる場合

非対称鍵を用いる場合，鍵の管理を公開鍵サーバで一元的に行うことが可能となる他，ユーザが他のユーザ (又はそのメッセージ) を認証するための秘密情報を持たなくて済むという利点がある反面，公開鍵サーバを設けなければならないため，システムの構築に手間がかかるという欠点がある．認証や暗号化のプロトコルは，対称鍵を用いる場合となるべく近い方法を用いている．こちらの場合では，2.2.1 の条件を満たすために次のような方法を用いる．

認証 チャレンジ・レスポンス方式

$Response = Challange \| UserID +$ 電子署名 ☆☆☆

機密性 非対称鍵暗号を用いてセッションキーを共有し，セッションキーを用いてエンドツーエンドで共通鍵暗号による暗号化を行う．

データの完全性 メッセージに電子署名 (メッセージのハッシュ値を秘密鍵で暗号化したもの) を付加する．

☆ $h()$ はハッシュ関数
☆☆ $||$ は連結を意味する
☆☆☆ $+$ は電子署名を付加するという意味

4. 実　　装

アクセス可能なユーザのリストを IC カードに格納し，図 1 のように一覧表示するようにした．一覧からコンタクトをとりたいユーザを選択し，“Select” ボタンを押すことでそのユーザに接続を申し込むことができる． アクセスを受けたユーザ側に

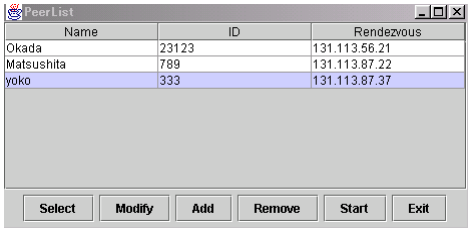

図 1 ユーザリスト実装画面

は接続を許可するかどうかを尋ねるダイアログが表示され，接続を許可すると双方向の認証が実行される．

4.1 実装環境

提案したシステムを Java 言語を用いて実装した．P2P のプラットフォームとして JXTA[4] を採用した．P2P プラットフォームは 3.1 におけるベース部分に相当する．本システムは JXTA 上のサービスとして実装されている．従ってアプリケーションは JXTA 上で動作するアプリケーションが本システム上でそのまま利用できる．カードリーダには同じく Gemplus 社の GCR401 を利用し，カードリーダにアクセスする手段として OCF(OpenCard Frameworks) を用いた．セキュリティに関しては，暗号化するアルゴリズムとして RC4 を用い，一方向性ハッシュ関数として SHA-1 を用いた．

4.1.1 Java Card

本稿では IC カードのプラットフォームとして Sun Microsystems の提供する Java Card[5] を利用した．Java Card とは，Java の実行環境を実装した IC カードであり，以下のような特徴がある．

- 異なるメーカーのカード間でもアプリケーションの互換性が保たれる．
- 必要に応じてアプリケーションの追加, 削除ができる．

- アプリケーションを複数搭載できる．
- 通常の Java 開発環境を利用できる．

Java Card 上で実行されるアプリケーションはアプレットと呼ばれ，プログラムとして機能するだけでなく，データを内部に格納することができる．

今回の実装では Gemplus 社の Java Card，GemXpressoRad211IS と同じく Gemplus 社の IC カードリーダライタ，GCR410 を利用した．

4.2 アプリケーション

アプリケーションとして，オークションアプリケーション (図 2) とチャットアプリケーション (図 3) を実装した．

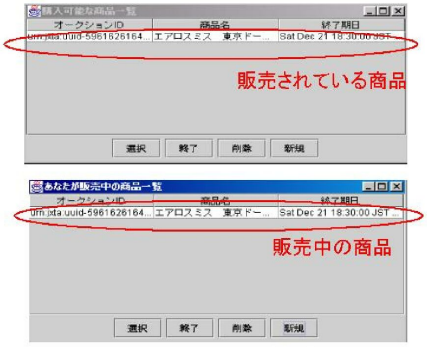

図 2 オークションアプリケーション実装画面

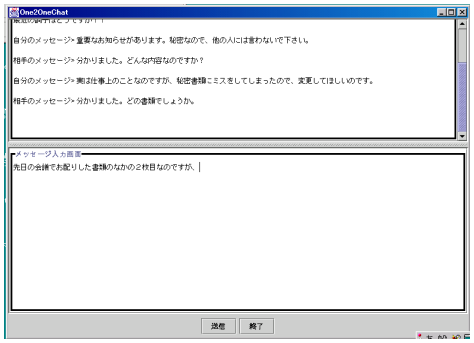

図 3 チャットアプリケーション実装画面

5. 評価と考察

5.1 ユーザの検索

異なるネットワーク内に存在する 2 つの端末を用いて，一方のユーザから他方のユーザの検索を行った．ファイヤウォールの有無と接続する Rendezvous Point の選び方を変えることによって，ユーザを発見できた回数と発見できた場合に要した時間を調べた．

Rendezvous Point への接続方法は次の三通りの方法を試した．

実験 1 2 つの端末から各々異なる Rendezvous Point に接続する．

実験 2 実験を行った時点において JXTA で用意されていた 4 つの Rendezvous Point 全てに接続する．

実験 3 同一の Rendezvous Point に接続する．

本提案で採用したのは実験 3 である．また，検索には 10 分間の制限時間を設け，制限時間以内にユーザを発見できなかった場合には見つからなかったものとした．検索は各々10 回ずつ行った．

表 1 ファイヤーウォール内のピアからファイヤーウォール外のピアを検索した場合

	発見できた回数	平均発見時間
実験 1	0 回	-
実験 2	6 回	4 分 36 秒
実験 3	10 回	32 秒

表 2 ファイヤーウォール内のピアから別のファイヤーウォール内のピアを検索した場合

	発見できた回数	平均発見時間
実験 1	0 回	-
実験 2	0 回	-
実験 3	10 回	24 秒

どちらの場合においても本提案で採用した実験 3 において最も発見できた割合が高く，しかも短時間で発見できたことがわかる．実験 2 の結果が 1 と 2 で違いが出ているのはファイヤーウォールの影響ではなく，偶然の結果であると考えられる．いずれにせよ，ユーザが必ず発見できなければコミュニケーションプラットフォームとしては信頼性に欠ける言える．

6. 今後の課題

6.1 「許可」のコントロール

今回の提案では 2.2.1 の「許可」についてはアプリケーションの側で対処することにしているが，ユーザの立場に付加的な情報に基づいて許可を制御

する仕組みがあると利便性が高まると思われる．

6.2 認証が食い違った場合の対処

クライアント/サーバシステムの場合，認証はサーバを基準として行われる．しかし，P2Pでは全てのピアが対等な立場にあることから，認証は基準を設けず，相互認証によって行われる．もしP2Pにおいて3人以上のグループを構築する際に，認証の結果が食い違ったとすると，どの結果を優先するのかは明らかでない．例えばAとBおよびBとCの相互認証が成功したが，AとCの相互認証が失敗した場合，グループ構築を行うかどうかを決定する必要がある．

この場合，失敗した結果を優先してグループを構築しないのが最も安全な方法であるが，ピア同士の関係は完全に対等であるという前提を置かずに，グループを構築しようとしているピアの中に基準となるピアを設けるという方法もある．

7. ま と め

本稿ではピア・ツー・ピア型のネットワークを用いて協調作業のためのグループを構築し，またグループ構築に必要な情報をICカードによって携帯することによって，ユーザの移動に対応した安全なコミュニケーションプラットフォームを実現する方法を提案した．この提案によって，ユーザはその場にあるPC端末を利用して他のユーザを検索し，アドホックにグループを構築できるようになった．

参 考 文 献

1) http://www.microsoft.com/japan/windows/netmeeting/default.htm, 25,Nov,2001.
2) http://www-6.ibm.com/jp/developerworks/java/011214/j$_j$ $-p2ptrust.html, 25, Nov,$ 2001.
3) 山崎重一郎，岩尾忠重，塩内正利，和田祐二，岡田誠，荒木啓次郎
P2P型エージェントプラットフォームにおける信用ドメインの構築について
マルチメディア，分散，協調とモバイル 2001 pp681-686.
4) http://www.jxta.org/, 25,Nov,2001.
5) http://www.java.sun.com/products/javacard/, 25,Nov,2001.
secutiry book 佐々木良一，吉浦裕，手塚悟，三島久典
インターネット時代の情報セキュリティ
共立出版株式会社