モバイルコンピューティングと　15－6
ワイヤレス通信
高度交通システム　3－6
（2000．11．30）

# ITSにおけるセキュアエージェントの検討

田中 俊昭　　清本 晋作　　中尾 康二

（株）ＫＤＤ研究所
〒356-8502　埼玉県上福岡市大原 2-1-15，TEL:0492-78-7406
e-mail: tl-tanaka@kddi.com

あらまし　ITS（高度道路交通システム）では、種々の通信メディアを介して、個々の車両が移動しながら、高速走行、準停止などの走行状態で通信サービスを提供・享受する。このため、様々なサービスを利用者の代理としてネットワーク側で自律的に実行させるエージェント技術の導入を検討している。しかしながら、各種 ITS サービスをエージェント利用環境で安全かつ高信頼に行うためには、エージェント・ホスト間における相互の安全性、利用者に対する認証や認可（アクセス制御）をエージェントが行う代理処理の安全性といった課題がある。従って、本稿では、これらの課題を解決するため、ITS におけるセキュアエージェント基本処理モデルおよび本モデルに従う具体的な代理認証や代理認可の実現メカニズムについて検討する。

キーワード：　ＩＴＳ、移動エージェントセキュリティ、代理認証、認可

# Study on the Secure Agents for Intelligent Transportation Systems

Toshiaki Tanaka　　Shinsaku Kiyomoto　　Kouji Nakao

KDD R & D Laboratories Inc.
2-1-15 Ohara Kamifukuoka-shi Saitama 356-8502, Japan
TEL: +492-78-7406
e-mail: tl-tanaka@kddi.com

**Abstract** ITS (Intelligent Transportation Systems) supports communication services, where each vehicle has access to ITS services via various type of communication media such as DSRC or cellular phone, while moving even at the high speed. For this reason, agents to be delegated the end user's task for ITS services are introduced. In order to realize the secure and high available ITS services on the agent platform, the mechanisms to protect agents against the execution environments vice versa, and to support secure delegation are believed to be main topics focused on. Therefore, we propose the process model of secure agents for ITS services and also show some concrete secure delegation protocols for authentication and authorization on top of the process model.

keywords: ITS, mobile agent security, proxy authentication, authorization

## 1. はじめに

ITS（高度道路交通システム）では、種々の通信メディアを介して、個々の車両が移動しながら、高速走行、準停止などの走行状態で通信サービスを提供・享受する必要がある。特に、高速走行の環境においては、DSRC(狭域無線通信)、セルラ、地上波ディジタルなど様々な通信メディアを跨り、サービスを提供しなければならない。このような環境においては、移動中に通信の途絶や、通信帯域不足などで円滑なサービスが提供できない場合がある。従って、より円滑なサービス提供を実現するには、車両とサービス提供側のネットワーク間の通信を可能な限り削減する技術が重要となる。このため、ネットワーク内に生起させたエージェントが、車両（以下、利用者とする）に代わって様々なサービスをサービス提供者との間で対話的に享受するエージェント技術の導入が検討されている[1]。ここで、利用者と通信を行うエージェントは、利用者の移動に伴い、追随してネットワーク上を移動していく移動エージェントを想定する必要があるが、各種予約サービスや決済等の ITS サービスを移動エージェント利用環境で安全かつ高信頼に行うためには、エージェントのホストに対する安全性、ホストのエージェントに対する安全性、利用者に対する認証や認可（アクセス制御）をエージェントが代理で行う際の安全性といった解決すべき課題がある。従って、本稿では、ITS における移動エージェントのセキュリティ技術に関わる課題を整理し、ITS におけるセキュアエージェント基本処理モデルを提案する。その処理モデルに基づく具体的なセキュリティサービスとして代理認証・代理認可メカニズムについて検討する。

本稿の構成としては、2章においてエージェント通信を想定した ITS の通信モデル、3章においてエージェント通信におけるセキュリティ要件、4章においてエージェントセキュリティ技術の ITS への適用について検討する。さらに5章において具体的な代理認証、代理認可メカニズムついて検討を行ない、6章において今後の課題を述べる。

## 2. エージェント通信を想定したITSの通信モデル

ITS のネットワーク構成イメージを図1に示す。ITS では、ITS-AP センタが提供する各種サービスを ITS サービスセンタが仲介し、路車間通信（DSRC)、移動体通信、放送系通信などの各種通信メディアを経由して、車両内の利用者が各種サービスを受けるモデルが検討されている。各種通信メディアを統合するバックボーンネットワークは、ITS サービスを提供する地域毎にドメインとして管理されることを想定する。この際、通信メディア間やドメイン間のローミングやサービス仲介を円滑に行うことにより、ITS サービスの連続性を実現する手法として以下の要件を満足する移動エージェント(mobile agent)技術を導入する。

- 通信メディアやドメインをまたがるサービスの連続性を実現するには、アプリケーション層において上記の機能を実現する必要がある。
- 予約サービスなどで、利用者の権限や嗜好に基づきサービスを仲介するインテリジェンシが必要となる。
- 仲介処理が利用者の移動に追随して移動する。

具体的には、図1における ITS-AP センタおよびバックボーンネットワーク上の ITS サービスセンタでエージェントを実行させる。ここで、一般性を失うことなく検討を容易にするため、

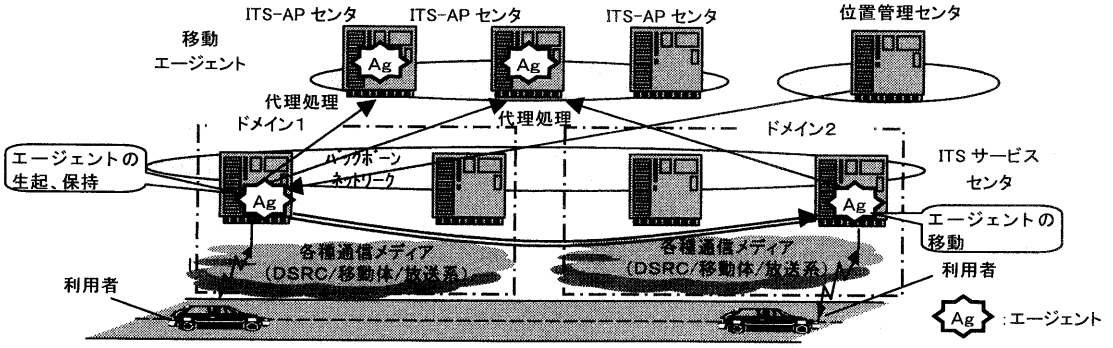


図1 ITS のネットワーク及びエージェント構成図

移動エージェントの通信モデルを以下のように単純化する。すなわち、1つのドメインのみを対処とし、ITS-APセンタのエージェントは、固定エージェント（Stationary Agent）を、ITSサービスセンタ上のエージェントは、利用者の移動に際してエージェントが複数のITSサービスセンタを移動する移動エージェントを想定する。なお、ITS-APセンタとITSサービスセンタ間のエージェントの移動は行なわない。

## 3. エージェント通信におけるセキュリティ要件

### 3.1 動作環境におけるエージェントのモデル

エージェントが動作するホスト上の環境は、ＯＳ、ＶＭ（仮想マシン）、エージェントプラットフォームなどがあるが、図2に示すように、本稿では、これらをまとめて実行環境と呼ぶこととし、1ホスト上で一つのみ存在するという制約を与える。実行環境は、ホスト外部からネットワークなどを経由してエージェントを実行環境上で動作させる、また、実行環境上のエージェントを外部ホストに移動させる、移動エージェントの通信機能を有する。さらに、エージェントが実行環境を介して他のホストや他のホスト上のエージェントとの相互通信可能とする。

ここで、エージェントに対して操作を行なう役割としては、一般的に作成者(creator)、所有者（owner)、実行者（executor）に分類される。作成者とはエージェントプログラムを設計・実装・配布するエンティティ、所有者とはエージェントを起動・終了するエンティティ、実行者とはエージェントを実行するエンティティを表す。図2の例では、作成者が異なる（V,W）エージェントが実行環境により生起され、そのうち1つのエージェントが外部に移動する。また、他の実行環境で生成されたエージェントを受付ける状態を表す。

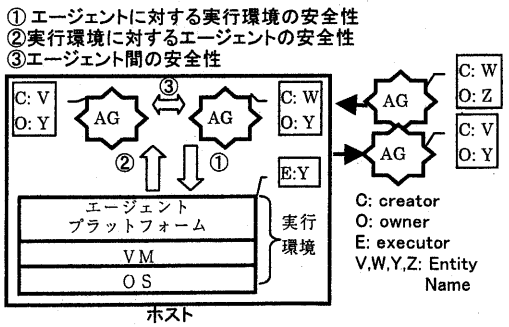


図2 エージェントの動作環境モデル

### 3.2 エージェント動作環境における安全性

ホスト内部のエージェント動作環境の一般的なセキュリティ脅威については、Jansen[2]らによってまとめられており、以下に示すように、①実行環境、②エージェント、および、③エージェント間の観点からの安全性について指摘している。

(1)エージェントに対する実行環境の安全性

エージェントを受け付け、実行させる環境では、受け付けたエージェントが正当なエージェントであることを認証する必要がある。さらに、エージェントが実行環境に対する不正アクセス行わないことを保証する必要がある。また、実行環境のリソースを大量に消費して、実行環境を機能不全にさせるＤＯＳ(Denial of Service)攻撃を防ぐ必要がある。

(2)実行環境に対するエージェントの安全性

移動先の実行環境が正当であることをエージェントが、認証する必要がある。エージェントの内部情報の盗聴や、改竄を防ぐ必要がある。さらに、エージェントのプログラムを不正に実行し、エージェントの当初の目的を達成させないＤＯＳ攻撃を防ぐ必要がある。

(3) エージェント間の安全性

エージェント間で通信を行なう際には、相互にエージェントを認証する必要がある。また、相手エージェントに対する不正アクセス、エージェント通信事実の否認を防ぐ必要がある。

一般的に知られているこれらの脅威に対するセキュリティ対策例を表1に示す。

表1 エージェント実行環境における脅威と対策例

| 脅威 | 保護対策 | | |
|---|---|---|---|
| | 実行環境のエージェントに対する | エージェントの実行環境に対する | エージェントのエージェントに対する |
| なりすまし | 署名付きコード | 検討課題 | 相手認証 |
| 不正アクセス | アクセス制御（認可）<br>署名付きコード | — | アクセス制御（認可） |
| 改竄 | — | 実行履歴管理 | 改竄検知コード |
| 盗聴 | — | 難読化<br>モバイル暗号 | モバイル暗号 |
| DOS | Software Fault Isolation<br>PCC(Proof Carrying Code) | — | — |
| 否認 | — | — | 電子署名 |

### 3.3 エージェント通信環境における安全性

3.2節において、エージェントの実行環境の脅威に対する対策がなされているとの仮定のもとで、エージェントが分散された通信環境で動作する際の安全性について以下に検討する。

(1) 固定エージェント

ネットワーク上で異なるホスト（すなわち、異なる実効環境上）に存在するエージェント間の通信においては、3.2節の動作環境上での脅威に加えて、ネットワーク上に脅威が存在する。すなわち、なりすまし、データの盗聴、改竄、通信否認などである。これらの対策は、相手認証、情報秘匿、メッセージ認証、電子署名などの機能によりそれぞれ実現される。

(2)移動エージェント

移動エージェントは、移動先の情報を得て、正当な実行環境に移動する必要がある。

### 3.4 エージェントが提供するサービスにおける安全性

3.2節および3.3節での安全性の前提のもとに、エージェントサービスが提供される。例えば、商品の購入エージェントの場合､利用者に代わって、購入オーダを作成し、商店に提示する処理が必要となる。これらの処理を安全に行なうために必要となるセキュリティ機能を抽出する。

(1) エージェントの代理性

エージェントが、利用者から依頼された各種の代理処理を安全に行なうためのセキュリティ機能が必要となる。例えば、エージェントでは、利用者とサービス提供者間で保証する相手認証などのセキュリティに関る各種タスクを利用者の代理として遂行する。

(2) エージェントの継続性・一過性

エージェントが生起中は、(1)の各種の代理セキュリティ処理が可能となるが、タスクを終了後は不正な代理処理を防止する必要がある。

## 4. エージェントセキュリティ技術の ITS への適用

### 4.1 ITS 通信モデルに基づくエージェントの要件

3.3 節や 3.4 節の上位レベルでのセキュリティ機能の実現方法は 3.2 節のエージェント実行環境の安全性が保証されるレベルに依存する。特に、移動エージェントの環境においては、耐タンパー性が重要な技術となる。例えば、エージェントが完全な耐タンパー性を保有している場合には、利用者の秘密情報をエージェントが内包できるため、当該エージェントが利用者の振る舞いを完全に模擬できることとなり、利用者そのもののセキュリティ技術に置きかえることが可能となる。しかしながら、移動エージェントによる耐タンパー性は、完全に解決された課題ではないため、各アプリケーションに合致したセキュリティレベルを前提として、3.4 節のエージェントが提供する安全なサービスを構築していく必要がある。従って、3章で述べたエージェントの安全性に基づき、さらに、ITS の特徴を考慮したエージェントセキュリティ機能をについて以下に検討を行なう。

(1) ITS におけるエージェントの移動形態

ITS においては、図1の通信モデルに従い、利用者の代理としてエージェントを生起させた後は、利用者とエージェント間で通信が必要な際には、利用者の車両に最も近傍の ITS サービスセンタにエージェント自身が移動する形態を想定する。

(2) ITS におけるエージェントの内部状態

図1の通信モデルに従い、エージェントは利用者と通信を行なう際に移動する。ここで、エージェントは、利用者との通信途絶中も与えられたタスクを ITS-AP センタと対話的に行なう。従って、エージェントのタスク実行状況に応じて、エージェントの内部状態や内部データが変化し、他の ITS サービスセンタに移動することを想定する。

(3) ITS におけるエージェントのルーティング

通信モデルに位置管理サーバを想定する。エージェントに移動の必要が生じた場合、ITS の位置管理サーバに問い合わせを行ない、起動元である利用者の現在位置からその近傍の ITS サービスセンタの位置情報を得て、エージェントが移動することを想定する。具体的には、ITS サービスセンタ上のエージェントは、次の移動先のアドレスを知るために、エージェントが生起した時点で、位置管理サーバに対して、自身を登録する。また、ITS-AP センタからの利用者（車両）との通信要求を受けて、位置管理サーバは、エージェント名と送付先のアドレスが記載された送付先情報を当該エージェントに送付することを想定する。

(4) ITSにおけるエージェントの耐タンパー性

移動環境で、利用者の代理処理を行なうには、利用者の秘密情報を移動エージェントが持ち回る場合が想定される。従って、移動エージェントが実行環境に対して耐タンパー性を確保する。

### 4.2 ITSで提供するエージェントセキュリティ機能

3章で述べたエージェントのセキュリティ要件のなかでも、ITSに依存しない一般的な対策については本稿の検討対象外とし、特に、4.1節の要件に関わる以下の項目に焦点をあて検討する。

(1)耐タンパー性

耐タンパー技術としては、一般的に

①耐タンパーハードウェア装置を実行環境上に置く[6]

②ソフトウェア難読化(Obfuscator)技術を用いる。

③Tschudinらの Mobile Cryptography[4]技術を用いる。

などの方法が考えられる。①の手法は、耐タンパーハードウェア装置が安全であるとの仮定のもとに実現できるが、ITSにおいては、移動エージェントを想定していること、ITSサービスセンタではトランザクションが比較的大きな環境での利用を想定しており、耐タンパーハードウェア装置に対して、高負荷に対応可能な処理機能が予想されることから、現実的ではない。また、③の手法は、証明可能な安全性が保証され、値を直接知られることなく、エージェントに計算処理を依頼する手法であるが、利用できる処理関数が限定される。従って、本稿では②の難読化技術を前提とする。但し、難読化技術は、暗号学的に証明可能な安全性が保証された方式ではないため、Hohlの時限ブラックボックス[3]の手法を用い、以下の仮定のもとに安全性を保証する。すなわち、

*Time-limited Secure: "一定時間内においては、ソフトウェア耐タンパー性が保証される。一定期間後は、耐タンパー性が消失するが、その秘匿情報は無効となる。"*

(2)エージェント認証

*Time-limited Secure*の仮定を用いたエージェントでの認証技術について検討する。固定エージェントにおいては、常に、'owner' = 'executor'になる。また、同一実行環境内の固定エージェント間の通信においても、所有者(owner)は同一である。上記環境においては、実行環境、実行環境内のエージェントに信頼関係が自明的に成立しており、これらは、セキュリティの観点から同一エンティティとみなすことが可能である。従って、エージェントの秘密情報として、実行環境が保有する秘密情報を用いることが可能である。一方、移動エージェントにおいては、上記の*Time-limited Secure*による安全性のもとに、一時的な秘密情報を保有できる。

従って、エージェントの保有者(owner)と実行者(executor)が同一の場合には、実行環境すなわちホストを認証することによりエージェント認証が成立することになる。すなわち、実行環境が有する秘密情報(例えば、公開鍵暗号の秘密鍵)を用いて、実行環境の相手認証を行なう。その認証が成立している状況で、エージェントの所有者が実行環境と同一であることを確認する。一方、保有者(owner)と実行者(executor)が異なる場合は、移動エージェントと考えられるため、エージェントが内包する一時的な秘密情報を用いて認証を行なう。

(3)エージェント認可方式

一般に、エージェントが不正に実行環境のリソースをアクセスするのを制限するため、プログラムコードレベル(例えば、ファイルへのアクセスやソケットインタフェースの呼び出し関数の許可など)のアクセス制御を行なう。ここで上記のエージェント認証機構から明らかなように、エージェントの保有者と実行者が同一の場合と異なる場合でアクセスできるリソースが異なる、すなわち

*if 'owner' = 'executor' then* エージェントは実行環境上の秘密情報にアクセス可

*else* 実行環境上の秘密情報にアクセス不可

また、アクセス制御のポリシーについては、エージェントが保有するアクセス制御ポリシーと実行環境が保有するアクセス制御ポリシーから、当該実行環境上でのエージェントのアクセス制御ポリシーを決定する手法が考えられているが[7]、移動エージェントの環境においては、さらにエージェントが複数の実行環境を移動していくため、その実行環境上でのアクセス制御ポリシーの委譲方法として、図3に示す

3つの方式が考えられる。

①NoDelegation 方式　エージェントにポリシーが記載されず、実行環境上のアクセス権限のみで、実行される非委譲方式、

②SimpleDelegation 方式　最初に起動されたエージェントのポリシーが変更されなまま、移動エージェント先でも利用する手法

③CascadedDelegation 方式　エージェントのポリシーと実行環境のポリシーを融合した結果を、次の実行環境に対するエージェントのポリシーとする手法。

ここで、ITS におけるエージェントの移動は、図1に示すように、利用者の移動にともない、エージェントが利用者と通信に必要が生じた場合に、利用者の近傍の ITS サービスセンタにエージェントを移動させる形態をとる。すなわち、実行環境である各 ITS サービスセンタが、地域に依存してそのポリシーが変更することは無いと考えられる。すなわち、ITS サービスセンタは、同一のセキュリティポリシーに基づき構成されていると考えられるため、ITS 環境では、SimpleDelegation 方式を採用する。

(4)メッセージ認証方式

4.1 節(2)で述べたように、エージェントは移動する毎に状態が変わることが想定される。従って、エージェント移動元の実行環境がエージェント移動時にエージェントに電子署名を行う署名付きコードを用いる。

(5)エージェントのセキュアルーティング方式

エージェントが生起した時点で、位置管理サーバに対して、自身を登録する際、エージェントが生起した状態では、'owner'='executor'であるため、実行環境が保有する秘密情報（例えば、公開鍵暗号の秘密鍵）を用いて位置管理サーバに改竄検知付きの秘匿された情報を送付する。また、位置管理サーバからエージェントが受け取る位置情報は、エージェント名と送付先のアドレスが記載された送付先情報を当該エージェントに送付する。送付情報は、以下のとおり、

Signed (Server-ID, Agent-ID || Target-Address || DT)

ここで、Sined(x,y)は、メッセージyに対して、エンティティxが署名を施しメッセージyに添付した情報を、Server-ID は、位置管理サーバのＩＤを、Agent-ID はエージェントのＩＤを、Target-ID は、移動先の実行環境のＩＤを、ＤＴは、日付時刻情報を表す。また、署名のための公開鍵などは認証機関などの第三者により保証されていると仮定する。

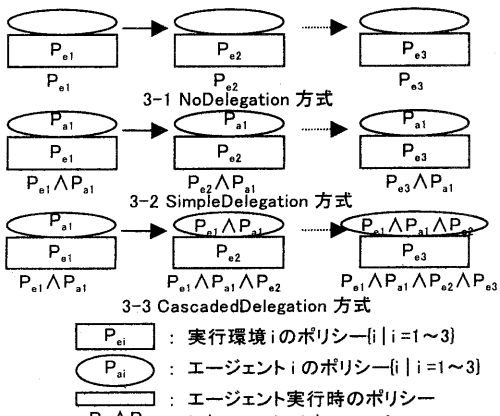


図3　アクセス制御ポリシーの委譲方式

### 4.3 ITS セキュアエージェント基本処理モデル

4.2 節で検討したセキュリティ機能を図1の ITS の通信モデルに適用した結果を図4に示す。以下に全体の流れを記す。なお、ITS-AP センタ(C)上のエージェント(AG2)は、固定エージェントであるので、ITS-AP センタ(C)として扱う。

① 利用者(U)が ITS サービスセンタ(Y)に ITS サービスを要求（予約サービス等）する。

② 利用者(U)からのサービス要求を受けて、実行環境がエージェント(AG1)を生起する。

③ エージェント(AG1)は、位置管理サーバ(P)に対して、自身のエージェントを登録する。この際、利用者の識別子も登録される。

④ エージェント(AG1)は、サービス提供者である ITS-AP センタ(C)にサービス処理要求を行なう。ここで、ITS-AP センタ(C)とエージェント(AG1)間で、例えば、認証処理等を行なう際、所有者と実行者が同一(Y)であるため、永続的な秘密情報を用いる。また、エージェントは必要に応じて、移動先で利用可能な一時的な秘密情報を作成する。

⑤ ITS-AP センタ(C)でサービス処理を終了後、利用者(U)に対して処理応答要求が発生する際、位置管理サーバ(P)がエージェント(AG1)に対して、移動要求を行なう。移動要求には、移動先情報が含まれる。

⑥ エージェント(AG1)は目的の ITS サービスセ

ンタ(Z)に移動する。この環境では、所有者(Y)と実行者(Z)が異なるため、永続的な秘密情報のアクセスは禁止され、エージェントが内包する一時的な秘密情報のみが利用可能となる。

⑦ ITS-AP センタ(C)から移動先のエージェントに対して、処理応答を送信する。

⑧ エージェント(AG1)は要求元の利用者(U)に対してサービス応答を返す（予約完了など)。

⑨ 移動先の ITS サービスセンタ(Z)上の実行環境がエージェント(AG1)を終了させる。

## 5. エージェントを用いたセキュリティサービス

### 5.1 代理認証メカニズム

以下、エージェントによる利用者の代理認証メカニズムについて述べる。本方式の特徴は、

①4章のセキュアエージェント基本処理モデルに従い、利用者(U)は、エージェント(AG1)に代理認証を依頼する。すなわち、エージェントが起動された実行環境においては、永続的な秘密情報を用いて、ITS-AP センタ(C)に対する時限認証情報(Auth)を作成し、以後、エージェント(AG1)が移動する際には、*Time-limited Secure* な時限共有鍵を用いて、ITS サービスセンタ(C)が移動先のエージェント(AG1)を認証する。

②ITS-AP センタ(C)とエージェント(AG1)が秘密裏に鍵を共有する際、エージェント(AG1)は利用者の協力なく鍵を共有できないことを、認証プロトコルを通して、ITS-AP センタ(C)に対して保証する。

③時限情報(s)を利用者(U)が指定することにより、指定寿命をこえて、代理認証が成立しない。

STEP 0: ITS-AP センタ(C)は、乱数 c (c:0<c<q-1) を選択し、Rc=g^c mod p を計算後、Rc をエージェント(AG1)すなわち、ITS サービスセンタ(Y)に送付する。以下、C,AG1,Y,U を用いる。

STEP 1: AG1 は、Rc と自身の公開鍵 Py を U に送付する。

STEP 2: U は、時限情報 (s=Hash(TD || Policy) mod q) を作成、乱数 $r_u$ ($r_u$:0<$r_u$<q-1) を選択、$g^{r_u}$ mod q を作成し、これらから時限認証情報 (Auth=Rc^(su)/Py^$r_u$= g^(csu-y$r_u$)) を作成し、Px, Auth, $g^{r_u}$ mod p,<s>を Y に送付する。

STEP 3: AG1 では、Auth, $g^{r_u}$ mod p と、Y の秘密鍵 y mod q から (Auth)((g^$r_u$)^y)= Rc^(su) mod p を計算し、C との共有鍵となる Kshare = (Rc^su) (Rc^y)を作成する。さらに、m と Kshare をハッシュ関数で処理し、これらの <s>, m, Hash(Kshare, m)を C に送付する。

STEP 4: C では、送られてきた<s>から共通のハッシュ関数で s=Hash(TD || Policy) mod q を計算し、求めた s と、U 及び Y の各々の公開鍵 Pu,Py から共有鍵 K'share = (Py^c)(Pu^(cs)) を求め、自身で得た共通鍵 K'share を用いてハッシュ値 Hash(K'share,m)を求め、送られてきたハッシュ値 Hash(Kshare, m)と比較し、一致していれば、利用者の認証が成立するとともに、以後、Kshare を一時的な認証情報として、<s>、m に含まれるエージェントＩＤと共に保管する。

ここで、p: prime かつ q | p-1、q: prime、g: mod p で位数 q の原始元、利用者(U)の秘密鍵、公開鍵は、各々u (u:0<u<q-1)、Pu = g^u

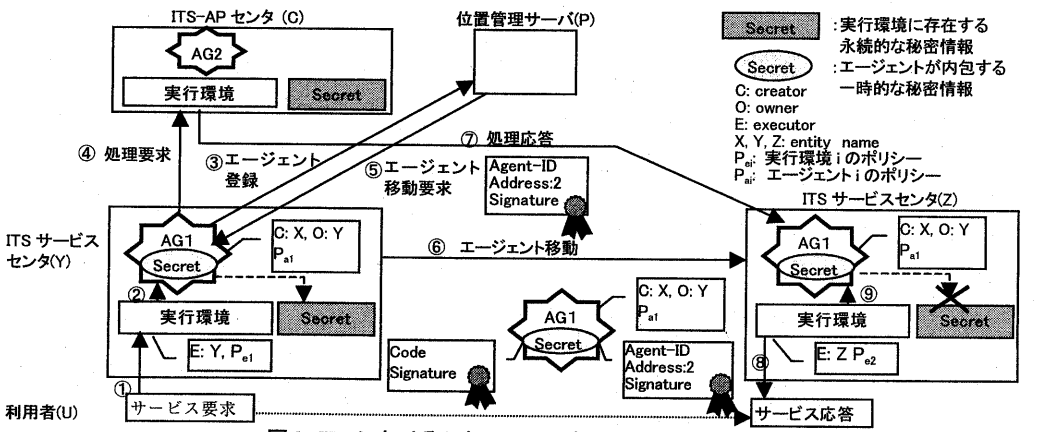


図4 ITS におけるセキュアエージェントアーキテクチャ

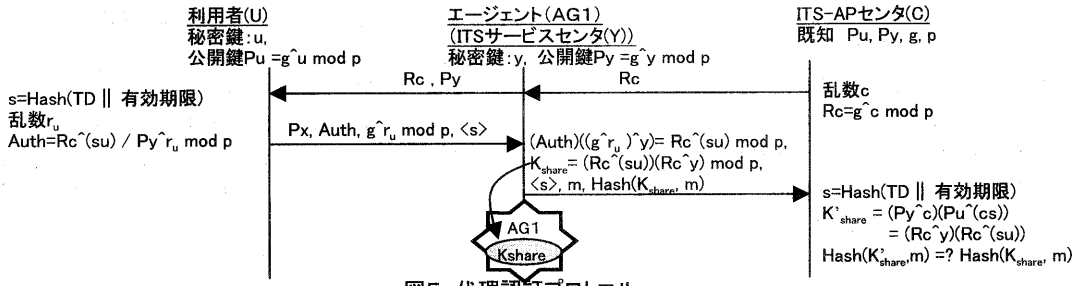


図5 代理認証プロトコル

mod p、ITS サービスセンタ(Y)の秘密鍵、公開鍵は各々y (y:0<y<q-1)、Py = g^y mod p、Hash:一方向性関数、m:任意の乱数とエージェントIDの結合データ、<s>: TD || Policy、||:データの結合、TD:時刻情報、Policy:本認証情報の寿命の記述とする。例えば、代理認証機能が10分間有効である場合は、policy= (*lifetime of the agent is ten minutes*)となる(図5参照)。

STEP 3 において、ITS-AP センタ(C)では、利用者側で作成される一時的な認証情報 s を用いて、共有鍵 Kshare を作成しているため、一定期間のみ認証が成立し、一過性が保証される。エージェントの移動先では、STEP4 での Kshare および s を用いて、エージェントを認証する。

### 5.2. エージェントの代理認可メカニズム

エージェントサービスとしてのアクセス制御については、利用者がエージェントに対して依頼するサービスの集合を Role として定義し、粒度の細かな制御を行う Role-Base 手法を用いる。すなわち、図6に示すような信頼できる第3者機関より発行された Nagaratnam[5]らの代理証明書(Delegation Certificate)を利用者が保有し、ITS サービスセンタに代理要求する際に、当該代理証明書を送付する。ITS サービスセンタでは、代理証明書に含まれる役割証明書(RoleCertificate)の内容と、ITS サービスセンタのローカルなセキュリティポリシーと比較し、許可されたサービスのみが実行可能となる。

```
IMPORTS everything FROM X.509
DelegationCertificate ::= SEQUENCE{
 initiator            Name,    -- 要求元
 role      RoleCertificate,    -- 役割証明書
 validity             Validity, -- 有効期限
 delagationServer Name, -- 失効管理サーバアドレス
 delagationConstraints Constraints, --代理機能上の制約
 signature          OCTETSTRING -- 第3者機関の署名}
RoleCertificate ::= SEQUENCE{
 proxyAuthService       INTEGER, -- 代理認証サービス
 iTSReservationService INTEGER --  ITS 予約サービス}
```

図6 代理証明書の例

## 6. むすび

本稿では、ITS サービスのエージェントへの適用において、動作環境、エージェント通信環境、エージェントが提供するサービスの3つの観点からエージェントセキュリティ機能について検討を行ない、ITS 通信モデルの要件に従うセキュアエージェント基本処理モデルを提案した。さらに、上記基本処理モデル上で提供されるセキュリティサービスとして、代理認証および代理認可について、具体的なメカニズムを提案した。本稿で検討したセキュアエージェント基本処理モデルは、ITS サービスセンタ上に限らず、路側側のネットワークへの適用も可能であると考えられる。また、今後の課題としては、他のセキュリティサービスのメカニズムや時限ブラックボックスの実現手法などがある。最後に、日頃ご指導頂く、㈱ＫＤＤ研究所、小花取締役、浅見副所長、秋葉所長に感謝します。

参考文献

[1] 田中、中尾、清本 "ITS におけるエージェント認証、認可方式の検討" 情報処理学会第61回全国大会 3F-7 Oct. 2000.

[2] Wayne Jansen and Tom Karygiannis "Mobile Agent Security", NIST Special Publication 800-19:, National Institute of Standards and Technology, August 1999.

[3] Fritz Hohl "A Model of Attacks of Malicious Hosts Against Mobile Agents", 4th Workshop on Mobile Object Systems(MOS'98) Secure Internet Mobile Computations, 1998.

[4] Tomas Sander and Christian F. Tschudin "Protecting Mobile Agents Against Malicious Hosts", LNCS on Mobile Agents and Security, spring 1998.

[5] N. Nagaratnam, and D. Lea, "Secure Delegation for Distributed Object Environments," USENIX Conference on OOTS'97, April 1998.

[6] Uwe Georg Wilhelm "A Technical Approach to Privacy based on Moble Agents Protected by Tamper-registant Hardware" 1999.

[7] L. Gong, "Java Security Architecture (JDK1.2)", 1998.