3F−7　　ITSにおけるエージェント認証・認可方式の検討

田中 俊昭　　清本 晋作　　中尾 康二
(株)KDD 研究所

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。このため、様々なサービスを利用者側の代理としてネットワーク側で自律的に駆動す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、エージェントがサービス利用者に代わって、種々の要求を行う際には、なりすまし、不正アクセスといった脅威を防ぐ代理の認証技術が、また、エージェントがネットワーク上を移動する場合には、実行環境上でエージェントを正しく認証し、与えられた権限に従ったタスクのみを実行するための認可技術の検討が必要となる。従って、本稿では、ITS におけるエージェントの認証・認可方式について検討する。

2. ITSにおけるエージェント通信の適用

ITS の接続構成イメージを図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。この際、各種通信メディアをまたがり、連続的な ITS のサービスの提供・享受を行う技術が必要となる。また、予約サービス等では、ITS サービスセンタが複数の ITS-AP センタと通信を行い、利用者と ITS-AP センタの仲介を行う必要がある。このような通信メディア間のローミングやサービス仲介を円滑に行う実現方法として、本稿では以下の観点を考慮しエージェント技術を導入する。

・通信メディア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。

具体的には、図１における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、ITS-AP センタに対して、エージェントが利用者に代わって各種処理を代行するとともに、車両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してタスクを実行する移動エージェントを想定する。

3. エージェント通信におけるセキュリティ要件

ITS におけるエージェントに関するセキュリティ要件を以下に示す。

(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必要がある。

(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 ITS-AP センタに対して不正な代理処理を防止する必要がある。

(3) エージェントの移動性
エージェントが移動する際、エージェントおよび ITS サービスセンタが互いに不正動作を防ぐため、移動エージェントの認証やアクセス制御等の機能が必要となる。

本稿では、上記(1)～(3)を満足する具体的なメカニズムを検討する。特に(1)では、利用者に代わり、ITS-AP センタからサービスを受ける際のなりすましを防止する代理認証を検討する。

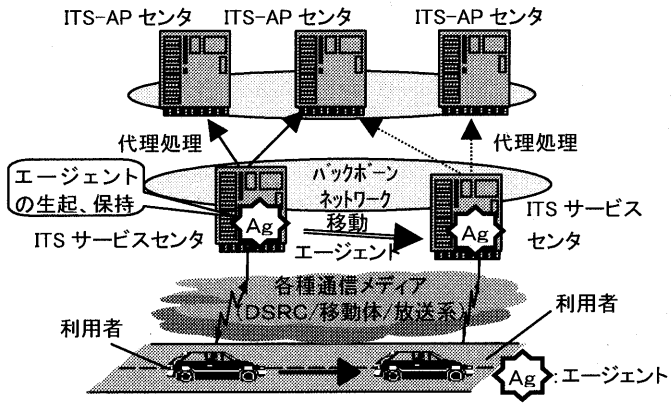

図1 ITS のネットワーク及びエージェント構成図

Study on Agent Authentication and Authorization Mechanisms for the ITS.　Toshiaki Tanaka, Kiyomoto Shinsaku and Kouji Nakao　KDD R & D Labs Inc.
2-1-15 Ohara Kamifukuoka-shi Saitama 356-8502,Japan

4. エージェントの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。なお、ITS サービスセンタとその上で動作するすべてのエージェントは、相互に正当性が保証されているとする。本メカニズムの特徴としては、

①利用者(U)は、エージェント(A)に代理認証を依頼するため、ITS-AP センタに対する一回限りの時限認証情報(Auth)を作成し、エージェントが本認証情報を引継ぎ情報として利用する。

②ITS-AP センタ(C)とエージェントが秘密裏に鍵を共有する際、エージェント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 センタに対して保証する。

③時限情報(s)を利用者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: C: 乱数 u (u:0<u<q-1) を選択、
z←g^u mod p
send C→A:z

STEP 1: A: send A→U:z, Py

STEP 2: U:s←Hash(Timestamp || Policy)mod q
乱数 r (r:0<r<q-1) を選択、
Auth←z^sx/Py^r=g^(usx-yr)
send U→A: Auth, <s>, g^r

STEP 3: A: Auth * (g^r)^y= z^sx
K_share ← z^y * z^sx
send A→C: <s> || m || Hash(K_share, m)

STEP 4: C:s←Hash(Timestamp || Policy)mod q
K_share ← Py^u * Px^(su)
verify Hash(K_share,m)

ここで、p: prime かつ q | p-1、q: prime、g: mod p で位数 q の原始元、利用者の秘密鍵、公開鍵：x (x:0<x<q-1)、Px = g^x mod p、ITS サービスセンタの秘密鍵、公開鍵：y (y:0<y<q-1)、Py = g^y mod p、Hash:一方向性関数、m:任意の電文、<s>: Timestamp || Policy、||：データの結合、Timestamp：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。

STEP 3 において、ITS-AP センタでは、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 K_share を作成しているため、一定期間のみ認証が成立する。

5. エージェントの認可メカニズム

移動エージェントの認証については、移動元の ITS サービスセンタが移動エージェントのプログラムに対してディジタル署名を行う一般的な Authenticode[1]を用いる。すなわち、各 ITS サービスセンタは互いに信頼できると想定し、Authenticode による認証により、移動エージェントの正当性を保証する。

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[2]。すなわち、図２に示すような信頼できる第３者機関より発行された代理証明書(DelegationCertificate)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2 代理証明書の例

6. むすび

本稿では、ITS における円滑なサービス提供を目的として、エージェントを用いた際のセキュリティ機能について検討し、具体的な代理認証、時限エージェントおよび移動エージェントの認可の基本メカニズムについて提案した。今後、移動エージェントのコードレベルの安全性について検討を進める予定である。最後に日頃ご指導頂くＫＤＤ研究所秋葉所長に感謝します。

[1] e.g. Java Security Architecture (JDK1.2),1988.
[2] N. Nagaratnam, and D. Lea, "Secure Delegation for Distributed Object Environments," USENIX Conference on OOTS'97,April 1998.