6G−3

エージェントのマルチホップにおけるハッシュを用いた高速認証*

江藤 秀一　宇田 隆哉　岩田 善行　重野 寛　松下 温 †
慶應義塾大学理工学部‡

1 はじめに

ネットワークの急速な普及により，情報サービスの内容は大きな広がりをみせている．しかし，現状のような支援ツールのサポートのみでは，ネットワークを介してアクセス可能な膨大な量の情報やサービスの活用が困難である．このような問題に対して，エージェント技術が有効であると考えられる．エージェントとは，人間の代理人としてネットワーク上で機能するプログラムの総称で，人間に代わって必要な情報やサービスを探し回ったり，状況に合わせて行動を柔軟に修正するなどの機能を有することから，ネットワーク活用の有効な手段として期待できる．

現在，エージェントを用いた様々なプラットフォームが提案されているが，マルチホップにおける認証方法は確立されていない．そこで本研究では，エージェント通信においてハッシュを用いた低コストで高セキュリティな認証法を提案する [2][3].

2 エージェントによる通信システム

2.1 モバイルエージェントの利点と問題点

モバイルエージェント [4] は，プログラムコードを保持したままホスト間を移動できる（マルチホップという）ため，ホストを次々と経由して処理をする際にトラフィックの軽減を図れるという利点を持つ．また，エージェントがインテリジェントである場合には，各ホストにおいて必要な処理のみを自動的に行うことができ，エージェント固有の実装を各ホストにおいて予め実装する必要がなく，セキュリティ面の向上を図れる．さらに，その処理中ユーザは通信回線を切断しておくことができ，通信コストを削減できることも利点の一つである．

しかし，問題点としてエージェント通信路の安全性・オーナシップ・実行権限・アクセス権利・ライフサイクル・コントロールなどが挙げられる．エージェントが移動する通信路が安全でない場合，公開ネットワーク上で運営されるサービスは非常に危険である．単に暗号化により安全性を確保することはシステムの負荷やネットワークのトラフィックを増大させることにもなり，モバイル機器との連携や大規模なシステムでは，従来方式での安全性の確保は困難であると言える．

また，オーナーシップとは各エージェントの持ち主や出身となるドメインを特定することであり，オーナシップの管理は実行権限やアクセス管理の基盤となる．オーナシップの特定は各々のエージェントの認証によって得られる．しかし，モバイルエージェントの認証は非常に難しい性質を持っている．

2.2 マルチホップ時の認証

図1において，マルチホップ時の認証について概略を説明する．

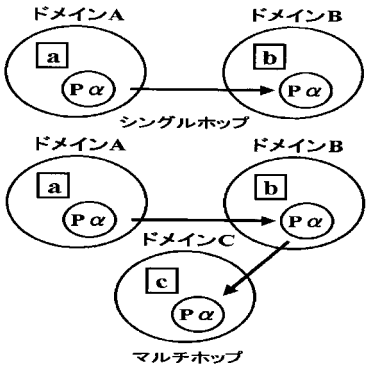

図 1: エージェントの認証

通常，モバイルエージェントにおける認証は，ドメインの認証に基づくものである．図1の例では，ドメインBは，ドメインAとドメイン間での認証を行うため，ドメインAから発信されたエージェントを認証できる．しかし，マルチホップの場合，ドメインAから発信されたエージェントPαは，ドメインBを経てドメインCへ到着する．ドメインCは，ドメインBを認証することは出来るが，通信を行わないドメインA

*Flexible and Inexpensive Onestop Service System with Agents
†Shuichi Eto, Ryuya Uda, Yoshiyuki Iwata, Hiroshi Shigeno, Yutaka Matsushita
‡Faculty of Science and Technology, Keio University

を認証することは出来ない．そこで，ドメインBから来たエージェントが，本当にドメインAから発信されたものであるかどうかは確認することができない．

本研究では，このようなエージェント通信路の安全性や認証に着眼しそれを解決する方法として差分方式を提案する．

3　差分方式による高速な認証方法

3.1　差分方式の概要

本研究では，エージェントがマルチホップする際，ハッシュを多段に用いることで認証を行う方式を提案する．本提案は，公開鍵暗号方式を使わずに高速な認証を行うことが可能であるため，負荷の高い状態においては非常に有効であると考えられる．本提案では，この差分方式によってエージェントの処理を行うことを前提としている．図2に，今回提案する差分方式の簡単な例を示す．

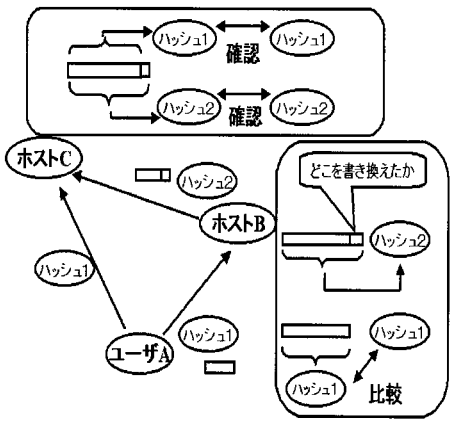

図 2: 差分方式によるエージェントの認証

図2では，ユーザAは，あるデータをホストB，ホストCの順で処理するように要求している．そこで，ユーザAは，自分自身が発信するエージェントのハッシュをブロードキャストでホストBとホストCに送信する．ホストBは，ユーザAから受け取ったデータ部は破壊せずに，エージェントの後部に，どのデータをどのように処理したかのプログラム部を追加し，その全体のハッシュをホストCに送信する．ホストCは，ユーザAとホストBからそれぞれのハッシュを受け取るため，同様のハッシュを計算することにより，受け取ったエージェントのどこに不正があったかを判断できる．

ここで，エージェントが4者間をマルチホップする場合を，図3に示す．

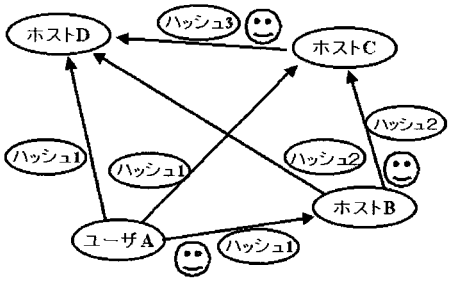

図 3: エージェントのマルチホップ

このように，エージェントが複数のホストをマルチホップしていく場合にはハッシュが多段となるが，差分方式によって不正ホストの発見が可能となり，システムの早期リカバリにも繋がる．モバイルエージェントが処理をする場合には，本来，エージェント自体が内包するデータ部を更新するが，今回の差分方式では，受信データ部を更新せずに処理内容をログとして付加することでハッシュをとることを可能にしている．この差分方式が本研究の最大の特徴である．

4　まとめ

本研究では，エージェントの通信路の安全性や認証の問題を解決する方法として，差分方式を提案した．この方式を，エージェントを利用した大規模ネットワークに利用することによって，低コストで高いセキュリティを維持できる．また，本提案は，複数のホストで連続的に処理を行う必要のあるサービスにおいて非常に有効である．

参考文献

[1] http://www.cgc.co.jp/text/security6.html

[2] 佐々木 良一, “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，2000

[3] http://www.ibm.co.jp/developerworks/java/000915/j_jw-0428-security.html

[4] Prithviraj Dasgupta，Nitya Narasimhan，Louise E. Moser，P.M. Melliar-Smith，“MAgNET:Mobile Agents for Networked Electronic Trading”，IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING VOL.11，NO.4，JULY/AUGUST 1999,pp.509-525