高安全モバイルエージェントシステムアーキテクチャ

正員 岡宅 泰邦* 非会員 奥富 秀俊**
 非会員 吉岡 信和*** 非会員 大岸 伸之**
 非会員 本位田 真一****

A Highly Secure Mobile Agent System Architecture

Yasukuni Okataku*, Member, Hidetoshi Okutomi**, Non-member, Nobukazu Yoshioka***, Non-member, Nobuyuki Ohgishi**, Non-member, Shinichi Honiden****, Non-member

We propose a system architecture for mobile agents to improve their security in the environments of insecure networks and non-sophisticated terminals such as PDAs. As mobile agents freely migrate onto their favorite terminals through insecure networks or terminals, it is not appropriate for them to store some secret information for authentication and encryption/decryption. We introduce one and more secure nodes(OASIS NODE) for securely generating and verifying authentication codes. The each agent's data are encrypted by a pseudo-chaos cipher mechanism which doesn't need any floating processing co-processor. We've constructed a prototype system on a Java mobile agent framework, "Bee-gent" which implements the proposed authentication and cipher mechanisms, and evaluated their performances and their applicability to business fields such as an auction system by mobile agents.

キーワード：認証，暗号，カオス，モバイルエージェント，

Keywords: authentication, encryption, chaos, mobile agent

1. はじめに

ネットワークが高度化することにより，公共無線ネット，アドホックネットワーク，プライベートネットワーク，インターネット，イントラネット等さまざまなネットワーク形態が出現する一方，端末の小型化・高度化により，多様な情報機器を使って時間・場所の制約なく，自由に情報を送受

信できる環境が整いつつある。しかしながら，さまざまなネットワークや機能・能力の異なる端末で統一的に扱える認証技術や暗号化技術が整備されているとは言えない。例えば，PDAのようなセキュリティ機能が必ずしも搭載されていない端末を所持する複数のユーザが，安全でないネットワークを介して共同作業する場合には，MAC(Message Authentication Code)による共通鍵方式⁽¹⁾や公開鍵方式による認証機能の利用を前提にするのは現実的でない。例えば，フリーマーケットのような不特定多数の人々が集まる場で，オークションを行う時に，買い手の端末にセキュリティ機能が搭載されていなければ安全な商取引ができない。特に，売り手のサーバにユーザが接続する場合，安全かどうか不明なサーバや端末等のノードを中継して相互接続される場合，非安全なノードを中継している間にかいざんされている危険が大である。本論文では，ユーザの作業を代行する主体（プログラム）をモバイルエージェント（以後，単にエージェントと記す）と定義する。エージェントは，不変なコード部と可変なデータ部から構成され，サーバ，端末間を移動する機能を有している。前述のオークションの例では，エージェントは，主催者サーバとの認証作業，コンテンツ配信先の指定や配信先サーバでの認証作業とコンテンツのダウンロード作業を代行する。ここで，エージェ

* 日本テレコム（株）戦略企画本部 情報通信研究所
 〒 104-0032 東京都中央区八丁堀 2-9-1
 Information and Communication Laboratories, JAPAN
 TELECOM CO. LTD.

9-1, Hatchobori 2Chome, Chuo-ku, Tokyo 104-0032

** 東芝情報システム（株）
 〒 210-8540 神奈川県川崎市川崎区日進町 7-1
 TOSHIBA INFORMATION SYSTEMS CORPORATION.
 7-1, Nissin-cho Kawasaki-ku, Kawasaki-city, Kanagawa
 210-8540

*** 国立情報学研究所
 〒 101-8430 東京都千代田区一ツ橋 2-1-2
 National Institute of Informatics.
 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430

**** 国立情報学研究所/東京大学
 〒 101-8430 東京都千代田区一ツ橋 2-1-2
 National Institute of Informatics and The University of
 Tokyo.
 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430

ントが、ネットワーク内のノードを移動しながら、収集した情報をユーザが指定した端末やサーバに搬送する際、搬送先ノードでの認証を共通鍵方式で行う場合、エージェント自身が鍵を保持するのは危険なため、当該ノードに別手段で安全に配送しておく必要がある。また、エージェントが公開鍵を保持している場合も、秘密鍵を当該ノードに別手段で安全に配送しておく必要がある。すなわち、不特定多数のエージェントがネットワーク内を移動中に、かいざんやなりすまし攻撃あるいは、盗聴が想定される状況下において、エージェント間の安全な共同作業環境の構築を実現するシステム構築が必要となる。本論文では、システムアーキテクチャとそのプロトタイプシステム構築について論じる。本システムが対象とする領域は、特別なセキュリティ機能を保持しないエージェントが、ネットワーク上の任意のノード内で、他エージェントと情報交換や情報処理の分担などを行う領域のことである。そのような領域内において、エージェント群が必要時に一時的にグループを形成して、目的とする作業を共同で実行する。作業が完了すれば共同作業環境は自然解消される。

このように、エージェントが稼働する情報機器や移動するネットワークのセキュリティ機能が、統一されていなかったり、不十分な環境下では、エージェント自身が、秘密情報を格納して、ネットワーク上を移動するのは避ける必要がある。また、剰余演算や浮動小数点演算用のコプロセッサを持たない端末も存在するため、そのような端末でも、一定の性能が確保できる暗号方式が望まれる。

以下、2. で基本アイデアを、3. でシステムアーキテクチャについて、4. でプロトタイプシステムについて、5. で考察についてそして、6. で今後の課題について論じる。

2. 基本アイデア

本章では、安全が保障されないネットワークにおいて、複数ユーザのエージェント群による共同作業を安全に実行するための、エージェント間で共通の秘密情報を保持する必要がない認証機構ならびに、浮動小数点演算用コプロセッサ(FPU)が不要な、整数型カオス暗号機構についての基本アイデアを記述する。本論文のシステムでは、信頼できるホスト(以後オアシスノードと記す)による認証方式⁽²⁾を用い、エージェントが、かいざんやなりすまし攻撃を受けていないかをオアシスノードで検証する。また、認証されたエージェント間の安全な情報共有は、FPUが不要な整数型カオス暗号方式⁽³⁾を用いて実現する。

2・1 エージェント間認証 文献(2)では、1つ以上のオアシスノードを導入することにより、ユーザ端末側には認証に必要な情報処理機構が不要な、複数エージェントを効率良く認証する機構が示されている。本認証機構では、乱数や認証コードは盗まれてもかまわない公開情報であるが、オアシスノードが保持する定数表を公開しない方式であり、不特定多数のエージェントによる一時的な共同作業における認証処理に適している。ユーザが4人の場合の、

図1 認証の基本アイデア

Fig. 1. A basic idea of authentication.

基本アイデアの動作概要を図1に示す。4エージェントがオアシスノード1から認証コードを発行してもらい、共同作業を開始されるまでは、各エージェントは、自身の作業を好きな場所で行っている。共同作業が必要になれば、適当なオアシスノード2に参集して認証コードを検証してもらい。オアシスノードが正と判定すれば、共同作業開始が許可される。エージェント自身は、秘密鍵等の秘守データを保持する必要はなく、オアシスノードから付与される認証コードを自データ部に格納するだけである。したがって、エージェントが秘密情報を保持することなく、ネットワークを移動することができる。

2・2 整数型カオス暗号 本システムで使用する暗号方式は、カオスの単純生成構造を有し、複雑な振る舞いを呈することができる性質を有する擬似乱数式ストリーム暗号の一種である。カオスの幾何学構造が、整数演算範囲へ拡大しても保たれるという性質を利用しており、そのため、整数演算とビット演算のみで設計できるという特徴がある。カオス生成には、浮動小数点演算を用いるのが通常であるが、本方式では、浮動小数点演算が不要のため、FPU類を搭載しない機器への実装が見込める。そのため、小型情報端末やICカード等の小型機器が広く使用される、ユビキタス環境に適した暗号方式と考えられる。

3. システムアーキテクチャ

本章では、エージェントが動作するプラットフォーム、エージェント間の認証および、暗号機構のシステムアーキテクチャについて記述する。

3・1 エージェントフレームワーク Bee-gent 本システムを実装するにあたって、移動機能をサポートしたマルチエージェントフレームワークであるBee-gentを用いた⁽⁴⁾。Bee-gentは、アプリケーションやデータベースなどの基本要素、それらをネットワークを介して他のエージェントと接続するためのエージェントラッパー (Agent Wrap-

図 2 Bee-gent の全体フレームワーク

Fig. 2. Overview of Bee-gent Framework.

per) および、基本要素間の協調的な処理を実現する仲介エージェント (Mediation Agent) から構成される (図 2)。

仲介エージェントは、分散システムが一貫した動作を行うために、エージェントラッパー間の調整機能、他のエージェントとの通信機能、および、ネットワークを介して他のラッパーに移動する機能を持っている。また、エージェントラッパーは、他のエージェントとの通信機能や、アプリケーションやデータベースと接続するための機能を持っている。これらのエージェントは、IP (Interaction Protocol) と呼ばれる他のエージェントとの相互作用プロセスで記述され、Java の状態クラスで実装される。

各エージェントは、相互に非同期通信を行い、条件付きでメッセージの取り出しが可能となっている。また、メッセージは、Speech-Act theory⁽⁵⁾ に基づいた KQML⁽⁶⁾ をベースに開発された ACL⁽⁷⁾ を採用している。このメッセージの表現形式は、XML 形式で記述されており、プロトコルとして HTTP を用い、エージェント間メッセージ通信を行う。このように、Bee-gent では、システムをその抽象レベルにあわせて 3 階層に分けて記述・実装し、システム全体の一貫性の調整を ACL を用いて行うことで、従来の分散システムより柔軟で拡張性の高い分散システムの構築が可能にしている。

アプリケーション、エージェントラッパー、仲介エージェントを図 2 のように構成することで、仲介エージェントからは、ホストのリソースや処理は隠され、アプリケーション間の調整ロジックのみに集中した記述が可能になっている。

3.2 認証機構 認証コードの安全な発行と、それを用いた認証を行う本認証機構を実現する上で、文献 (2) では、3 つの制約を前提としている。制約 1: 共同作業を主催するエージェントは、共同作業開始の宣言時にオアシスノードに参集している必要がある。制約 2: オアシスノードは 1 つ以上存在し、認証コード生成機能と認証コードの検証機能を有するものとする。制約 3: 認証コード生成後は、各エージェントは任意のノードで作業をしてもかまわないが、認証する必要がある場合は、オアシスノードに移動して認証コードを検証してもらう。制約 2 と制約 3 は、オアシスノードにおいて、認証コードの発行と検証を行う上

図 3 擬似カオスストリーム暗号システム

Fig. 3. A Pseudo Chaos Stream Cipher System.

で自然な前提である。一方、制約 1 は、共同作業宣言時に、ユーザ端末内のエージェントが、オアシスノードに安全に移動できていることを保証するための前提条件である。

主催エージェントは、他の (N-1) 個のエージェントの名前と IP アドレスを予め知っており、オアシスノードは、エージェント群の認証に必要な情報を送信できるものとする。以下、オアシスノード内の乱数と認証コードの生成機能および、認証コードの検証機能を総称して認証管理機能 (MFA; Management Function for Authentication) と記す。MFA は、非公開の定数表が外部に流出したり、かいざんされたりしない機能を有することが前提である。さらに、MFA は、エージェントが保持するデータの暗号と復号に必要な鍵の生成と配送機能を有する必要がある。

3.3 暗号機構 整数演算型カオス暗号の構造を図 3 に示す。本カオス暗号は、浮動小数点演算を用いず、整数演算、ビット演算のみで構成する。このような極端な有限ビットでの演算では、隣接カオス軌道の縮退が顕著化し、短周期性を招く。そこで、ある一定量のカオス乱数生成毎に、パラメータ可変関数を用いて、カオスパラメータを変動させる構造としている。また、パラメータ可変関数でのパラメータ変動パターンは、暗号鍵固有となるように与え、カオス軌道の暗号鍵固有性を維持できる方式としている。本方式は、1-byte のカオス算出毎に、1 回のカオスパラメータ変動を与える (dt=1 と記す) ことを基本としている。しかし、dt=1 での処理では、高速性を期待できないため、n-byte のカオス算出毎に 1 回のカオスパラメータ変動を与え

る($dt=n$ と記す)ことにより,高速化を図る。ただし,あまり大きすぎる n を与えることは,長期間に渡ってカオス軌道が,同一パラメータに滞在することを意味するため,短周期性の観点などから乱数性が維持できなくなる危険がある。したがって, n と短周期性のトレードオフを考慮するために,乱数検定の結果を見ながら最適化を図ることとなる。

3.4 鍵配送 オアシスノードで認証コードの検証にパスしたエージェントは,共同作業遂行のためにデータを安全に共有する必要がある。オアシスノード内部で共同作業する場合は,各エージェントは,自身のデータを暗号化してオアシスノードに移動する必要があるし,各エージェントがオアシスノード以外の端末で情報共有したい場合でも,その端末にデータを暗号化して移動する必要がある。本論文の整数型カオス暗号機構では,共通鍵方式を採用しているため,エージェントに対して,共通鍵を安全に配送する必要がある。本提案方式は,オアシスノードのMFAに全信頼を置くモデルであり,MFAは暗号鍵が必要な都度,乱数を生成し,それを共通鍵としてダミーエージェント群で配送する方法を採る。ただし,ネットワーク上での非公開情報の盗聴を避けるために,複数のダミーエージェントを組み合わせる方式を用いる(図4)。まず,エージェント A_i が,オアシスノードのMFAに鍵配送を要求する。MFAは, A_i 用の乱数 r_{Ai} と複数のダミーエージェント($=Dumma(i,j)$)を生成する($j=1, \dots, J$)。ダミーエージェントの生成個数は,毎回乱数により変動するようにしておき,盗聴者に個数を知られないようにする。MFAは, $Dummy=Dumma(i,1) \text{ XOR } \dots \text{ XOR } Dumma(i,J) \text{ XOR } r_{Ai}$ を計算し,($Dummy, A_i$ の認証コード)を格納した A_i と J 個のダミーエージェントを配信先端末に送信する。受信側ユーザ端末では, J 個のダミーエージェントと排他的論理を繰り返して,オアシスノードと A_i の共通鍵($=r_{Ai}$)を算出する。ダミーエージェント群の移動経路が複数存在する場合は,できるだけ

異なる経路を使用し,待ち伏せ攻撃の危険を少なくする。

4. プロトタイプシステム

本認証機構によるエージェント間認証と,コンテンツ等のデータを安全に授受するための整数型カオス暗号機構を搭載するプロトタイプシステムをPC上に構築した。ビジネス領域での実用可能性を評価するために,オークションのプロトタイプシステムを作成し,認証と暗号の機能を評価した。落札したコンテンツの配送に先立ち,3.4で記したダミーエージェントを用いた共通鍵配送を行う。

4.1 オークションシステム 不特定多数者が参加するオークションシステムへの適用可能性を,本プロトタイプシステムにて評価した。オークションシステムでは,販売者のサーバがオアシスノードとなり,購入者の端末に入札エージェントが配置される。購入者の保持する入札エージェントが,入札期間中,販売者のサイトに参集し,各エージェントが保持する入札戦略である,最低入札額,最高入札額および,上げ幅を用いて競争入札を行い,商品の購入者および購入価格を決定する。ここで,入札戦略は,エージェントのコードの一部として保持され,かいざんに対処する。入札戦略の変更が必要になった場合,再度,オアシスノードから,認証コードの発行を受ける。オークションシステムにおけるエージェントの動作例を図5に示す。ここで,同図中のAgentは,図2の仲介エージェントのことである。まず,入札開始前に,購入者の入札エージェントは,販売者のサーバに移動して,認証コードを配布された後,オアシスノードの入札条件と共に,認証コードをデータ部に格納して,自端末に戻る。そして,入札条件を確認して,入札戦略の変更がなければ,入札開始のため,販売者サーバに移動する。サーバに参集した各エージェントは,販売者サーバのMFAによる認証コードの検証を受ける。検証にパスしてはじめて,入札に参加できることとなる。動作例では,購入者2の認証コードが,かいざんされた例である。そのため,同図中のステップ3)で,入札開始前の販売者サーバによる検証で拒否され,入札できないことになる。一方,入札に参加できたエージェント中,最高金額を提示した購入者3が落札することになる。商品を落札した購入者3のエージェントは,自身が希望する端末へのダウンロードを,サーバに要求する。プロトタイプシステムでは,図5中のステップ6)で,コンテンツを分割して暗号化を行い,購入者端末に配信する。購入者端末では,ステップ5)の落札成功メッセージとともに,ダミーエージェントにより配送された共通鍵を用いて,受信したコンテンツを分割ブロック単位で復号化し,Windows Media Playerを用いて表示する。したがって,ダウンロードの途中から,コンテンツを表示できるため,ストリーム配信が可能である。落札者の端末が,暗号関数を持っていないときは,コンテンツ配信時に,暗号関数プログラムを送信する。サーバは,配送済みの鍵でプログラムを暗号化して端末に送信する。

図4 ダミーエージェント群による鍵配送

Fig. 4. Key distribution by dummy agents.

図 5 オークションシステム例

Fig. 5. An example of auction system.

4・2 システム評価 本節では、本認証プロトコルと整数型カオス暗号関数を、Bee-gent 上に MFA として実装したプロトタイプシステムの機能、性能について評価する。

4・2・1 認証処理負荷 本認証プロトコルでは、 N エージェントのハッシュ値 $H_i (i=1, 2, \dots, N)$ を計算するのに $3N$ 回のハッシュ計算と $(N-1)$ 個の認証コードの計算が必要である。本プロトタイプシステムでは、ハッシュ関数 h と関数 f_n に MD5⁽⁸⁾ を使用した。1.0GHz の Pentium III と RAM512MB 搭載の PC(Linux) で行った $(4N-1)$ 回のハッシュ値の処理時間を表 1 に示す⁽²⁾。乱数発生、排他的論理和ならびに、認証コード比較の処理時間は、ハッシュ計算負荷に比べ無視できる。

4・2・2 整数型カオス暗号の統計的乱数検定結果とセキュリティ強度について 本暗号方式は、擬似乱数式ストリーム暗号に属するため、擬似乱数生成部(コア部)が生成する乱数の性能に大きく依存する。表 2 に、(A) 等頻度検定、(B) ポーカー検定、(C) 連検定についての統計的乱数検定結果を記した。今回は、ランダムに選んだ 2E16 個の鍵について、各鍵から生成される 2E16、および 2E20 の長さの乱数を被検定データとして用いた。表 2 中の数値は、統計検定の慣例である 5 パーセントおよび 1 パーセント棄却率で棄却された試行の、全試行(2E16)に対する割合(パーセント表示)を意味する。各検定手法とも、それぞれが属する棄却率で棄却された試行の割合は、棄却率にほぼ等しい結果を得ている。また、今回の実験範囲では、 $dt=128$ の場合でも特に目立った乱数性能の劣化は見られなかった。すなわち、本暗号コア部が生成する乱数列は、理論分布に即しており、良好な結果といえる。

本論文において、セキュリティ強度に関する理論検証については論じていないが現在、検証作業に取り組んでいるところであり、別途結果を報告する。これまで提案されているカオス暗号システムについては、そのセキュリティが低いことが報告されている^{(9)~(11)}がその一方、改善方法も提案されている⁽¹²⁾。また、制御理論的アプローチから、カオ

表 1 MD5(JavaJDK1.4.0) ハッシュ計算時間

Table 1. Processing time of hashed value in case of MD5(JavaJDK1.4.0).

N	10	25	50	75	100	150
Size	sec	sec	sec	sec	sec	sec
1KB	0.004	0.009	0.018	0.027	0.036	0.054
5KB	0.009	0.023	0.046	0.069	0.092	0.138
10KB	0.018	0.046	0.092	0.139	0.185	0.278
50KB	0.087	0.222	0.446	0.670	0.894	1.342
100KB	0.175	0.445	0.894	1.343	1.792	2.690
500KB	0.805	2.044	4.109	6.174	8.239	12.369
1000KB	1.689	4.287	8.617	12.947	17.277	25.937

表 2 統計的乱数検定結果

Table 2. Statistical results of randomness tests.

		dt=1	dt=1	dt=128	dt=128
test type	rejection prob.	2E16	2E20	2E16	2E20
frequency	5 per.	5.03	5.27	4.98	4.88
frequency	1 per.	0.93	1.22	1.20	0.71
poker	5 per.	5.00	5.18	5.00	4.76
poker	1 per.	1.25	1.15	1.05	0.98
runs up	5 per.	4.88	4.79	5.35	4.83
runs up	1 per.	1.22	1.03	1.05	1.12
runs down	5 per.	5.32	4.83	5.25	5.25
runs down	1 per.	1.22	1.03	1.22	1.00

ス暗号システム(暗号器と復号器)を設計する手法の研究がなされており、最近、不正者による復号器の設計の困難性についてのシミュレーション実験報告がなされている⁽¹³⁾。文献⁽¹³⁾では、オブザーバー型カオス暗号システムを設計して、不正者が送信信号(暗号文)を同期化する適応復号器の頑健性についてシミュレーションしており、暗号システムの伝達関数の相対次数を 2 以上にすることで適応同期化を困難にできることが報告されている。

4・2・3 整数型カオス暗号の処理性能について 本暗号方式の性能を表 3 に示す。 dt 値を 128 にすることにより、CPU が 2.0GHz の Pentium4 で 517Mbit/sec の暗号と復号時の処理性能が得られている。これは、ストリーム処理性能としては、実用的な性能値を示している。なお、これら暗号処理方式の高速化手法による性能向上と、乱数性能に関する詳細な評価ならびに、他の検定方式を含めた検定結果については、現在実験を行っており、理論検証と合わせて別途結果を報告する。

本論文では、整数型カオス暗号方式の持つ優位性(コプロセッサ不要、高い処理性能、長い鍵サイズが可能)を示すためにプロトタイプシステムに組み込んだが、セキュリティ強度が既知のブロック暗号方式を用いることも可能である。ブロック暗号方式もコプロセッサは不要である。2つのブロック暗号方式(DES と AES)と本提案手法との処理性能比較を表 4 に示す(650MHz の PentiumIII/Windows98/VC++6.0)⁽¹⁴⁾。両ブロック暗号方式と比べて、本整数型カオス暗号方式の方が、優れた処理性能を示していることがわかる。

表 3 整数型カオス暗号の処理性能

Table 3. Performance result of the proposed pseudo chaos cipher system.

CPU/OS/Compiler	performance	
	dt=1	dt=128
AMD Athlon 1GHz/Linux/gcc	201 Mbps	440 Mbps
Intel Pentium4 2GHz/Linux/intel icc	323 Mbps	517 Mbps

表 4 ブロック暗号方式との比較

Table 4. Comparison of Block cipher systems and pseudo-chaos cipher system.

	DES	AES	pseudo-chaos
Key length	168 bit	128 bit	512 bit
Cipher type	Block	Block	Stream
Speed [Mbps]	48.6	130.9	289.0
by PentiumIII 650MHz/ Windows98 /VC++6.0			

5. 考 察

ネットワークの安全が保証されない環境下において、アドホックな共同作業環境の構築に適した認証方式と、低機能端末でもデータの暗号・復号が可能な整数型カオス暗号方式を実装したプロトタイプシステムを、エージェントフレームワーク上に構築した。求められる認証の負荷や暗号処理性能は、適用領域により異なるが、オークションやコンテンツ配信システムへの適用は充分可能と考えられる。本章では、エージェント認証時の制約 1 の緩和方式と、なりすまし対策ならびに、オークションシステムを含めた他の応用領域の可能性と課題について考察する。

5.1 制約 1 の緩和方式について オアシスノードとユーザ端末間が非安全なネットワークで接続されている場合、制約 1 を前提にするのは現実的でない。本論文では、制約 1 を緩和するために、3.4 の鍵配送で提案したダミーエージェント生成機能を、ユーザ端末側に保持させる方式を考察する。ユーザ端末内のエージェントが、共同作業宣言時にオアシスノードに移動する際、発生乱数個数分のダミーエージェントを生成し、エージェント全体の排他的論理和を実行する。そして、ダミーエージェントの任意の 1 つが、排他的論理和結果をデータとして格納し、全ダミーエージェントが、オアシスノードに移動する。オアシスノードに移動したダミーエージェント群により、元のエージェントに復元される方式である。全てのダミーエージェントが盗まれない限り、制約 1 の条件を充たすことができる。オアシスノードに移動したダミーエージェントは、復元後破棄される。ダミーエージェント群の移動経路が複数存在する場合は、鍵配送時と同様、できるだけ異なる経路を使用する。

緩和された制約条件の下で、エージェント間で共通の秘密鍵を必要としないエージェント間の認証プロトコルを以下に記す。 認証プロトコル

[ステップ 1] ユーザ端末の主催エージェントである

appA1 が、オアシスノードの MFA に対して、(N-1) 個のエージェント (appA2,..., appAN) との共同作業を主催することを宣言すると共に、appA2,..., appAN に、当該オアシスノードへの参集を要請する。参集を要請された各エージェント appAi は、自端末で発生させた乱数個分のダミーエージェントを生成し、自身をそれらダミーエージェントで排他的論理和を算出する。そして、ダミーエージェントの 1 つに、算出値をデータとして格納し、appAi とダミーエージェントは、オアシスノードに移動する。

[ステップ 2] MFA は、移動したダミーエージェント群から、元の appAi を復元する。そして、MFA は、N 個の乱数 (r1, r2, ..., rN) を生成し、自身が管理する非公開定数表から定数 (const) を選択して各乱数 ri との排他的論理和 $rri (= \{ri \text{ XOR } \text{const}\}, i=1,2,..., N)$ を計算する。MFA は、エージェントのコード部の完全性保証となりすまし防止のために、各エージェント (appAi) のコード部 (codeAi) と非公開数の rri とエージェント識別子 (idAi) を用いて、ハッシュ値 Hi (i = 1, ..., N) を以下のように算出する。

$Hi = h(hi1 \parallel hi2), hi1 = h(\text{codeAi} \parallel rri), hi2 = h(idAi)$.

MFA は、上記ハッシュ値をもとに、主催エージェント以外のエージェントに 1 つずつ配布する認証コード、 $\text{Cond}_2 = \text{fn}(H1 \parallel H2), \text{Cond}_3 = \text{fn}(H1 \parallel H3), \dots, \text{Cond}_N = \text{fn}(H1 \parallel H_N)$ を算出する。ここで、h はハッシュ関数を、Cond_K は appA1 と appAK 間の認証コードを表し、H1 H2 は H1 と H2 の結合を表す。上記 fn は、非公開定数に対応して MFA が選択するハッシュ関数が非周期関数あるいは、分散の大きな乱数発生関数である。MFA は、appA1 の内部に {r1, fn の種別名} を、各 appAi の内部に {ri, Cond_i} を、各エージェントのデータの一部として格納する。

[ステップ 3] 各エージェントは、参集時刻あるいは主催エージェントからの呼び出しがあるまでは、オアシスノードの外部で自由に作業を行うことができる。参集時刻あるいは主催エージェントからの呼び出しに従って、各エージェントは指定されたオアシスノードに参集する。

[ステップ 4] 指定オアシスノードの MFA は、appA1 と共同作業相手 appAi が格納している乱数 {r1, ri} と fn に対応する非公開定数を用いてハッシュ値 H1 と Hi を、ステップ 2 と同様に計算する。MFA は fn を用いて $\text{fn}(H1 \parallel Hi)$ を算出し、appAi が持ち込んだ Cond_i 値と比較する。 $\text{fn}(H1 \parallel Hi)$ の計算結果が、ステップ 2 の Cond_i 値と一致した場合、両エージェント (appA1 と appAi) の認証が成功したことになる。不一致ならば、オアシスノードは、両エージェントの共同作業を許可しない。

5.2 なりすまし対策 エージェントのコード部には、エージェント識別子 ($\text{idAi} = \{\text{NameOfAi} \parallel \text{IPAddress}\}$) が埋め込まれているので、ネットワーク上で第三者 (=X) が、認証コードを格納したエージェントを盗み、idAi をかきざんし、X になりすますしても、オアシスノード参集時の検証処理で検出できる。一方、X が何もかきざんせずにそのまま、オークションサイトのサーバ (=オアシスノード

ド)に送りつけて、サービス使用不能攻撃(DOS)をかけることも想定される。この場合、エージェントが、ステップ2で、オアシスノードから認証コードの発行を受けた時に、コードに記述されたIPaddressの端末に、発行された旨のメッセージを送信するようにコーディングしておくことで、第3者のDOSを検出することができる。すなわち、appAiが、認証コードの再発行依頼をしていないのに、当該メッセージがユーザ端末に送信される場合は、不正な使用が行われていると見なすことができるので、処理中断等による対処が可能となる。あるいは、Xが同一オークション中に、同名エージェントによる入札を複数回発生させた場合は、オアシスノードが、DOS攻撃と判定し当該エージェントの入札処理を中断することも可能である。

5・3 オークションシステム プロトタイプシステムでは、入札者のなりすまし検出や、入札戦略プログラムのかいざん検出に適用した。入札エージェントが、販売者のサイト(オアシスノード)にエントリーするために参集して、認証コードを発行してもらい、自端末に戻る。初期入札金額や上限金額を記した戦略(=プログラム)を保持した各入札エージェントがオアシスノードに参集し、各入札戦略にしたがって、入札状況をモニターしながら、入札金額を決定する。オアシスノードへの参集時に、各エージェントの認証条件が検証され、不正と判定されれば、入札に入ることができない。最終的に、落札できたかどうかは、オアシスノードから通知される。

5・4 コンテンツ配信 一人の購入者によるコンテンツ購入やレンタルの場合だけでなく、多数の購入者への同時販売やレンタルにも対応可能である。複数の配布者の予約(認証コード)がかいざんされていないことを同時に検証することが可能である。また、配信先サーバ(オアシスノード)は、駅や空港、ホテルなど多種多様な場所に設置が可能である。ただし、実用化には、課金システムの構築と不正2次配布の防止策が必要である。

5・5 共同購入や複数取引先との受発注業務などの一括管理 共同購入者や発注元、発注先のID等の変更されないデータに対するかいざんチェックが可能である。データ部のセキュリティは、提案するカオス暗号方式などの暗号技術で確保する。

5・6 パケット認証 ルータやスイッチ機器をオアシスノードとして実装し、送信元に一番近い機器で、パケット毎に認証コードを生成・配布して、受信側機器で当該パケットのかいざんチェックを行うことができる。認証コード生成に使用するエージェントのコード部に相当するのが、各パケット情報(=ヘッダー+ペイロード)である。初期条件として、各オアシスノード共通の秘密定数表と認証条件計算アルゴリズムを事前に配布しておく。秘密定数表の更新は、初期配布した定数表の一部を鍵として暗号化して、各オアシスノードにマルチキャストすることで実現できる。

6. 終りに

ネットワークの安全性が保証されない環境下において、アドホックな共同作業環境の構築に適した認証方式と、低機能端末でもデータの暗号・復号が可能な整数型カオス暗号方式を実装したプロトタイプシステムをエージェントフレームワークであるBee-gent上に構築し、各方式の基本機能、性能について報告した。また、制約1を緩和するダミーエージェントを用いた方法を提案した。今後、整数型カオス暗号方式の理論検証作業を進めるとともに、さらなる処理速度の高速化に取り組む。

謝辞

本研究の一部は、総務省「戦略的情報通信研究開発推進制度(研究主体育成型研究開発)」の支援(委託研究課題名:自由でかつ安全なコンテンツ流通を実現するためのエージェントフレームワークの研究開発)を受けて実施したもので、ここに記して謝意を表す。

(平成15年4月7日受付,平成15年9月12日再受付)

文 献

- (1) C. Kaufman, R. Perlman, and M. Speciner: "Network Security: Private Communication in a Public World", Prentice Hall (1995)
- (2) Y. Okataku, N. Yoshioka, and S. Honiden: "An Authentication Architecture For Collaboration Among Agents in Ad Hoc Networks", *The transaction of the institute of electronics, information and communication engineers (B)*, Vol.J86-B, No.3, pp.419-427 (2003-3) (in Japanese)
岡宅泰邦・吉岡信和・本位田真一:「アドホックネットワークにおけるエージェント間共同作業のための認証機構」, 信学論(B), J86-B, 3, pp.419-427 (2003-3)
- (3) 奥富秀俊:「整数演算型カオス写像を用いた疑似乱数発生手法と暗号応用について」, CSS2001 (2001)
- (4) Bee-gent, <http://www.toshiba.co.jp/beegent/>
- (5) J.L. Austin (ed.): "How to do Things With Words", Oxford University Press (1962)
- (6) T. Finin, et al: "An Overview of KQML: A Knowledge Query and Manipulation Language", Dep. of Computer Science, Stanford University (1993)
- (7) Agent Communication Language, <http://www.fipa.org/specs/fipa00061/>
- (8) R. Rivest: "The MD5 Message-Digest Algorithm", RFC1321, April (1992)
- (9) K.M. Short: "Steps toward unmasking secure communications", *Int., J. Bifurcation and Chaos*, 4-4, pp.959-977 (1994)
- (10) K.M. Short: "Unmasking a modulated chaotic communications scheme", *Int., J. Bifurcation and Chaos*, 6-2, pp.367-375 (1996)
- (11) C. Zhou and T. Chen: "Extracting information masked by chaos and contaminated with noise: Some considerations on the security of communication approaches using chaos", *Phys. Lett. A*, 234, pp.429-435 (1997)
- (12) T. Yang, C.W. Wu, and L.O. Chua: "Cryptography based on chaotic systems", *IEEE Trans. CAS-I*, 44, pp.469-472 (1997)
- (13) T. Matsuo, H. Suemitsu, and K. Nakano: "A Design of a Nonlinear Decrypter for Chaotic Secure Communications", *IEEJ Trans. EIS*, Vol.123, No.9, pp.1587-1596 (2003-9) (in Japanese)
松尾孝美・末光治雄・中野和司:「カオス暗号通信における非線形複合器の設計」, 電学論C, 123, 9, pp.1587-1596 (2003-9)
- (14) CRYPTREC Report 2001 (2001)

岡 宅 泰 邦（正員）1973 年関西学院大・理・物理卒。1975

年京大大学院工学研究科原子核工学専攻修士課程了。(株)東芝, ケイアンドアイソフトウェア開発(株)を経て, 2001 より日本テレコム株式会社戦略企画本部情報通信研究所に勤務, 現在に至る。セキュリティ, ネットワーク QoS, エージェント技術の研究開発に従事。電子情報通信学会, システム制御情報学会, 日本シミュレーション学会各会員。

大 岸 伸 之（非会員）1987 年信州大・理・物理卒。同年より

日本ビジネスオートメーション(株)(現, 東芝情報システム(株))に勤務, 現在に至る。ネットワークセキュリティ技術の研究開発に従事。情報処理学会会員。

奥 富 秀 俊（非会員）1995 年日大・理工・物理卒。1997 日

大大学院理工学研究科量子理工学専攻修了。同年より東芝情報システム(株)に勤務, 現在に至る。暗号化セキュリティに関する研究開発に従事。情報処理学会会員。

本位田 真 一（非会員）1976 年早大・理工・電気卒。1978 同

大大学院理工学研究科電気工学専攻修士課程了。(株)東芝を経て, 2000 より文部科学省国立情報学研究所教授, 現在に至る。2001 より東大大学院情報理工学系研究科教授併任。エージェント技術, オブジェクト指向技術, ソフトウェア工学の研究に従事。工学博士。1986 年度情報処理学会論文賞受賞。IEEE, ACM, 日本ソフトウェア科学会, 情報処理学会各会員。

吉 岡 信 和（非会員）1993 年富山大・工・電子情報卒。1995

年北陸先端科学技術大学院大情報科学研究科博士前期課程了。1998 同博士後期課程了。博士(情報科学)。同年(株)東芝入社。エージェント技術の研究開発に従事。2002 より文部科学省国立情報学研究所産業連携研究員, 現在に至る。情報処理学会, ソフトウェア科学会各会員。