Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística

Especificação de Segurança na Comunicação de Agentes do Sistema Multi-Agentes DEEPSIA

Francisco Gomes Milagres

Prof. Dr. Edson dos Santos Moreira
Orientador

Junho de 2002

Monografia apresentada ao Departamento de Ciências de Computação e Estatística do ICMC-USP, como parte dos requisitos para o Exame final da disciplina SCE-191 – Projeto de Graduação I.

Resumo

O objetivo do DEEPSIA (*Dynamic On-line Internet Purchasing System Based on Intelligent Agents*) é desenvolver uma infra-estrutura computacional que auxilie empresas como compradoras em processos de *e-procurement* no comércio eletrônico, utilizando um sistema multi-agentes. Por ser um sistema que utiliza a Internet e lida com informações críticas, um alto nível de segurança é necessário. O objetivo deste trabalho é estudar a atual arquitetura do Sistema DEEPSIA, avaliar as potenciais ameaças sobre seu Sistema Multi-Agentes e propor uma especificação que proporcione um nível mínimo de segurança a este sistema.

Agradecimentos

Gostaria de utilizar este espaço para agradecer às pessoas que direta ou indiretamente permitiram a realização deste trabalho.

Primeiramente agradeço a Deus pela força no desenvolvimento deste trabalho, bem como em todos os meus momentos difíceis durante toda a minha graduação em São Carlos, no ICMC – USP.

Agradeço imensamente a Andrea por estar sempre a meu lado.

Obrigado a meus pais que, mesmo a 1300 quilômetros de distância, sempre me ajudam a prosseguir e nunca me deixaram desistir de meus sonhos.

Agradeço a meu orientador, Prof. Edson Moreira, pelo incentivo e pelo conhecimento compartilhado desde o meu segundo semestre de graduação.

Obrigado aos amigos do Laboratório Intermídia pela ajuda, troca de conhecimentos e festas compartilhadas desde a minha entrada no grupo.

Obrigado às instituições de pesquisa FAPESP e CNPq que fomentaram ou ainda fomentam os meus trabalhos no grupo.

Índice

1 Introdução

O projeto DEEPSIA (*Dynamic on-linE IntErnet Purchasing System based on Intelligent Agents*)[1] (**DEEPSIA – IST-1999-20483**), que está em desenvolvimento no âmbito do programa europeu de pesquisa denominado *Information Society Technologies* (IST) e tem duração programada de 18 meses, foi iniciado em Janeiro de 2001. O consórcio DEEPSIA é formado por empresas e institutos de pesquisa de vários países europeus, dentre eles: empresa *ComArch* (Polônia), *Universidade Nova de Lisboa* – UNINOVA (Portugal), *Université Libre de Bruxelles* (Bélgica), *University of Sunderland* (Inglaterra), empresa *Zeus* (Grécia) e empresa *Atlante* (Espanha). O Brasil é o colaborador externo à União Européia do projeto DEEPSIA, sendo representado pelo ICMC – Instituto de Ciências Matemáticas e de Computação [2] e pelo NUMA – Núcleo de Manufatura Avançada da Escola de Engenharia de São Carlos [3] e fomentado pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) (**DEEPSIA – CNPq-68.0236/01-2**). A participação do Brasil está ligada a uma parceria direta com a UNINOVA e se encerrará oficialmente em Março de 2003.

O objetivo do DEEPSIA é desenvolver uma infra-estrutura computacional que auxilie a atuação das Pequenas e Médias Empresas (PMEs) como compradoras no comércio eletrônico em processos de *e-procurement* [4]. Esta infra-estrutura deve estar disponível para a entidade (pessoa ou departamento) responsável pelas compras e deve ser adaptável aos requisitos particulares da empresa em questão. Os modelos tradicionais de comércio eletrônico têm como característica focarem-se nas PMEs como fornecedores, normalmente dentro de centros comerciais ou mercados virtuais. No entanto, as PMEs são também compradoras de bens e serviços.

A infra-estrutura computacional em desenvolvimento pelo DEEPSIA visa tornar o processo de compra mais eficiente em termos de tempo e custo, fornecendo uma interface amigável baseada em um catálogo de compras personalizado que será automaticamente atualizado com a informação sobre os produtos, informação essa obtida dos portais eletrônicos existentes na Internet ou diretamente de bancos de dados das empresas parceiras

---

[1] DEEPSIA *Consortium web site:* http://www.deepsia.com

[2] Instituto de Ciências Matemáticas e de Computação (ICMC): http://www.icmc.usp.br

[3] Núcleo de Manufatura Avançada da Escola de Engenharia de São Carlos (NUMA): http://www.numa.org.br

[4] *e-procurement:* no *bussiness-to-bussiness,* é a cotação para compra e venda de produtos ou serviços pela Internet. (Fonte: http://searchhp.techtarget.com)

do sistema. Este catálogo personalizado fornecerá ao comprador um conjunto de ofertas apropriadas em termos de qualidade, diversidade e aplicabilidade.

O processo de busca e processamento de informação no sistema DEEPSIA é suportado por um conjunto de agentes inteligentes que analisam informação sobre produtos na *web* e processam seus resultados, com o objetivo de obter a informação referente aos produtos vendidos como, por exemplo, custo, descrição, etc. (Garção, 2001).

No âmbito do projeto DEEPSIA, a Universidade de São Paulo (USP) contribui em duas áreas distintas: a área sócio-econômica e a de desenvolvimento da infra-estrutura tecnológica do DEEPSIA.

2 Objetivos e Resultados Esperados

O trabalho em desenvolvimento na área sócio-econômica pela USP tem em especial consideração os aspectos legais, éticos e comerciais considerados pelas PMEs brasileiras, através do estudo das mudanças promovidas com a utilização do sistema DEEPSIA por empresas locais. Desta maneira o resultado global do DEEPSIA deixa de estar limitado unicamente à União Européia, passando a incorporar características da realidade comercial vivida no Brasil.

Na área de desenvolvimento da infra-estrutura tecnológica do DEEPSIA, que é onde está inserido especificamente este trabalho, a equipe da USP trabalha em diversos setores:

- *Data Mining* [5] nas informações do catálogo, visando a descoberta de informações que não sejam implícitas aos usuários;
- Categorização de textos: será utilizada para uma correta classificação dos produtos pelo *Miner Agent* segundo uma ontologia [6] definida;
- *Browsing* na ontologia: visando facilitar a interação do usuário com a ontologia dos produtos, através de mecanismos como busca e inserção de *bookmarks* [7];
- Extração de produtos: Novas funções estão sendo desenvolvidas no *Miner Agent* para melhorar a extração de informações sobre produtos em páginas de venda;

O objetivo deste trabalho é estudar a atual arquitetura do Sistema DEEPSIA, avaliar as potenciais ameaças sobre seu Sistema Multi-Agentes e propor uma especificação que proporcione um nível mínimo de segurança a este sistema.

Dentre os resultados esperados após a implementação da especificação aqui proposta, é possível destacar: (Thirunavukkarasu *et al.*, 1995) (Mayfield, 1995)

- Capacidade de autenticação: Os agentes do sistema de Multi-Agentes DEEPSIA devem ser capazes de provar as suas identidades e de verificarem a identidade de outros agentes que são envolvidos nas comunicações;

---

[5] *data mining:* classe de aplicações de banco de dados que buscam por padrões implícitos em grupos de dados. (Fonte: http://www.webopedia.com)

[6] ontologia, do inglês *ontology:* é uma especificação de uma conceitualização. É composta de um vocabulário de termos básicos e uma especificação precisa para o que estes termos significam. (Fontes: http://www.ontology.com e http://www-ksl.stanford.edu/kst/what-is-an-ontology.html). A ontologia utilizada será a UNSPC da ECCMA (http://www.eccma.org)

[7] *bookmark:* s. marcador ou endereço que identifica um documento ou ponto específico neste. (Fonte: http://www.webopedia.com)

- Capacidade de preservação de integridade da mensagem: Os agentes devem ser capazes de detectar alteração intencional ou não nas suas mensagens e devem prover métodos eficientes para que a mensagem em trânsito seja preservada;
- Capacidade de proteção de privacidade: A arquitetura de segurança do sistema de Multi-Agentes DEEPSIA deve prover mecanismos para que os agentes troquem informações confidenciais [8];

A seguir será apresentada a organização deste trabalho:

No capítulo 3 será feita uma descrição da arquitetura do sistema DEEPSIA, destacando o funcionamento de seus agentes e da forma de comunicação entre eles. O capítulo 4 apresentará o cenário atual de comércio eletrônico no Brasil e contextualiza a necessidade de segurança para que seja possível o desenvolvimento desta *nova economia.*

No capítulo 5 será apresentado um estudo das principais vulnerabilidades e ameaças contra sistemas de agentes e a especificação de requisitos desejados para a definição da arquitetura proposta neste trabalho. O capítulo 6 apresenta o estudo da linguagem de comunicação que é usada pelos agentes, a KQML (*Knowledge Query and Manipulation Language*). O capítulo 7 apresenta a arquitetura de segurança proposta, a sua modelagem e as limitações conhecidas, bem como as considerações sobre a continuidade deste trabalho. No capítulo 8 são listadas as referências citadas neste texto.

Estão incluídos também três apêndices a este trabalho; o apêndice A centraliza em uma lista os *links* de *sites* citados, o apêndice B apresenta das abreviaturas utilizadas e no apêndice C é apresentado um breve histórico do grupo de pesquisas Intermídia.

[8] Referências a *codificar/decodificar* e *encriptar/decriptar* devem ser consideradas como sinônimos neste texto.

3 O Sistema DEEPSIA

Um dos mais importantes tipos de serviços na *web* atualmente relaciona-se ao desenvolvimento de sistemas on-line aplicados a negócios, ou *e-business*. A atual onda de globalização da economia encontrou na Internet um meio apropriado para a divulgação e comercialização de bens e serviços. Nichos de mercado, antes inacessíveis por limitações geográficas, agora podem ser alcançados pela vasta abrangência da "rede mundial de computadores".

A fim de solucionar o problema da divulgação de negócios, muitos portais ou *market places* começaram a surgir. Tais portais estabelecem associações com *sites* de venda, mediante contrato e possibilitam a realização pelos usuários da Internet da busca por produtos oferecidos naquele domínio restrito. Pelo fato de divulgarem os seus produtos, torna se um bom negócio para os fornecedores de bens e serviços.

Porém, tal solução não satisfaz todas as necessidades do usuário, pois a principal intenção de uma pessoa ou empresa, ao buscar por ofertas de bens e serviços é avaliar qual negócio oferecido melhor lhe favorece em uma relação custo X benefício. A limitação do domínio na busca reduz as opções de escolha, constituindo a desvantagem do comprador em relação ao fornecedor nesse tipo de negócio proposto pelos *market places*.

Sistemas como o DEEPSIA visam apresentar uma solução mais adequada ao comprador, através de processos de classificação de páginas *Web* em um catálogo, que representa uma ontologia de produtos à venda. A correta extração de informação de produtos à venda depende de métodos otimizados de recuperação e classificação de conteúdos das páginas na Internet.

No sistema DEEPSIA, a busca e classificação de produtos em páginas da Web são feitas por três agentes distintos e com papéis bem definidos. O *Web Crawler Agent* busca páginas com produtos a venda, o *Miner Agent* extrai as informações sobre os produtos nas páginas e o *Dynamic Catalogue* armazena as informações sobre os produtos.

A figura 1 ilustra como ocorre a integração entre os principais agentes do DEEPSIA, que formam o Sistema Multi-Agentes e que se integram com os catálogos das empresas parceiras no projeto:

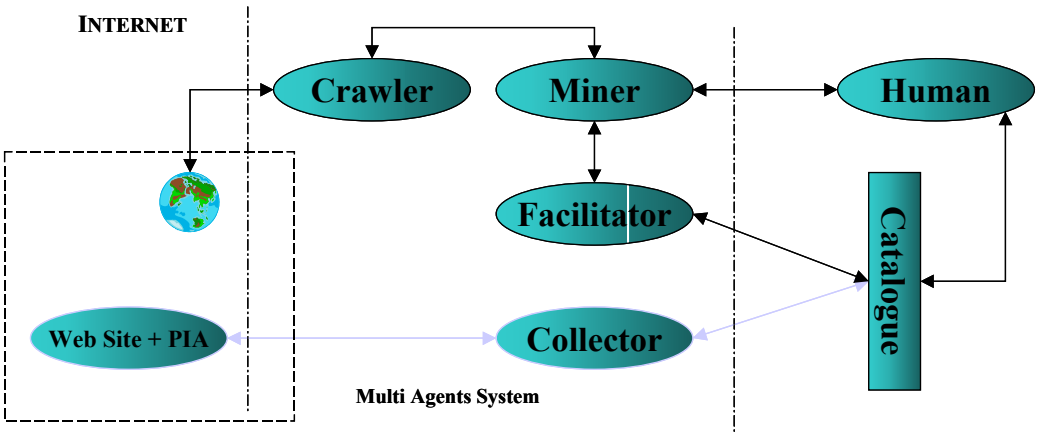

Figura 1: Os agentes que compõe a arquitetura DEEPSIA (Garção, 2002)

- *Portal Interface Agent* (PIA): Cria uma interface privilegiada entre os portais da *Web* e o *Multi-Agent System* (MAS). O agente atuará como um facilitador ao acesso a dados contidos nos *sites* e bancos de dados de fornecedores. Sua utilização pelos fornecedores (que é opcional), pode garantir que as informações sobre seus produtos serão acessadas pelo MAS, independente dos resultados das buscas efetuadas pelo *Web Crawler Agent.* A empresa Atlante detém a responsabilidade de seu desenvolvimento;
- *Dynamic Catalogue* (DC): Consistirá da interface do usuário e será responsável pela manutenção e apresentação dos dados coletados pelos agentes contatados, com base nas preferências do usuário fornecidas. Além de tais dados, o sistema possibilitará o acesso a informações de *sites* visitados, e a configuração da ontologia (representando o perfil individual da expectativa de compra) pelo usuário. A empresa ComArch detém a responsabilidade de seu desenvolvimento;
- *Multi-Agent System* (MAS): Sistema autônomo para coleta de dados e um processo semi-automático de atualização do catálogo, composto de um conjunto de agentes, com tarefas específicas. A responsabilidade de seu desenvolvimento é da UNINOVA. Os módulos internos deste sistema, que caracterizam os tipos especialistas de agentes, são descritos como:

    - *Web Crawler Agent* (WCA): Busca por páginas Web com dados de interesse pelo usuário, baseando-se em um processo recursivo a partir de uma "semente", sendo previamente treinado em um processo *off-line*. Executa a primeira seleção e classificação das páginas, separando em *páginas de venda* e *páginas que não são de vendas*, para posterior processamento do *Miner;*

- *Miner Agent* (MA): De posse da página, o *Miner Agent* se encarrega de obter informações sobre o tipo de produto que está sendo vendido na página e também de classificar este produto segundo uma ontologia;
- *Human Agent* (HA*):* Integrará a interface do usuário para validação sobre recursos e *sites* pelo *Facilitator Agent*, e possibilitará também a adição de *sites* pelo usuário ao *Miner*. É o único agente a estabelecer contato direto com o usuário, os demais serão executados através da interface com o catálogo.
- *Facilitator Agent* (FA): Delimitará a interface entre o catálogo e o conhecimento obtido pelo *Miner* e/ou *Collector Agent*. Possibilitará a configuração da periodicidade da produção e atualização de dados para o catálogo.

- *Collector Agent* (CA): Será a interface direta com o banco de dados que mantém os *sites* inscritos pelo PIA. Receberá mensagens do catálogo com solicitações por atualização e manterá a relação de usuários on-line, possibilitando também o acesso a informação de fornecedores inscritos.

No capítulo seguinte serão apresentados um breve cenário atual do comércio eletrônico e a necessidade de segurança da informação para a economia digital.

4 A Necessidade de Segurança

Desde a origem da Internet, o uso de redes de comunicação vem permitindo a efetivação de negócios e a troca de informação em velocidade e eficiência nunca antes imaginadas. Por meio destas redes, sejam elas públicas ou particulares, são também efetuadas transações entre parceiros de diferentes continentes e filiais de empresas que são localizadas a grandes distâncias.

De acordo com estudos de três grandes grupos de pesquisa, o Brasil é um dos mercados mais emergentes do mundo e as tendências para o *e-commerce* em longo prazo são animadoras para qualquer investidor. Segundo o IDC, devem ser investidos US$ 13,8 bilhões no Brasil até 2004. O *Yankee Group* [9] prevê um aporte de US$ 22,8 bi, enquanto o *Forrester Research* [10] tem uma previsão ainda mais otimista: US$ 59,4 bilhões.[11]

Com o uso cada vez mais intenso de novas tecnologias para esta comunicação no mercado global, surgem também novas ameaças contra o mais valioso patrimônio das organizações: a informação. A preocupação crescente pela segurança da informação foi redobrada nos últimos meses a partir de novas ameaças que não podiam sequer ser imaginadas, como a que se abateu sobre o *World Trade Center* (WTC) e o Pentágono em 11 de Setembro de 2001, quando muitas empresas perderam, além de seus funcionários, todas as suas bases de dados e sistemas de informação devido à um ataque terrorista. [12]

Grande parte destas empresas não possuía planos de contingência de seus negócios e não possuíam um planejamento de segurança visando assegurar o patrimônio digital da corporação em caso de catástrofes. A tecnologia, que neste caso não foi usada pelas empresas para assegurar a manutenção dos seus negócios, foi usada para assegurar a privacidade da comunicação dos terroristas nos planos de ataque através do uso de criptografia em mensagens eletrônicas.

Um dia após a tragédia que destruiu o WTC, os bancos *Deutsche Bank* e *Morgan Stanley*, que estavam instalados nos prédios, voltaram a operar quase normalmente. De acordo com informações da *Globo News* [13], o *Morgan Stanley* ocupava 25 andares de um dos prédios. Executivos afirmaram que as informações armazenadas nos computadores não foram

---

[9] *Yankee Group:* http://www.yankeegroup.com

[10] *Forrester Research:* http://www.forrester.com

[11] Fonte: Magazine Negócios Exame, Maio de 2001.

[12] Fonte: *Módulo Security Solutions*, http://www.modulo.com.br

[13] *Globo News:* http://globonews.globo.com

perdidas, graças ao plano de continuidade de negócios contra possíveis danos do *bug* do milênio - investimento feito em 1999. Os dados dos clientes estavam guardados em outro ponto da cidade.

É neste contexto cada vez mais notório de paranóia em segurança da informação que se mostra a necessidade de serem assegurados alguns fatores para que um sistema – seja ele de comércio eletrônico, de um banco ou de uma agência nacional de segurança – tenha um nível mínimo de segurança específico: sigilo, integridade de dados, disponibilidade, consistência, controle e auditoria (Spafford, 1996).

No capítulo seguinte serão apresentados um estudo de segurança no Sistema Multi-Agentes do DEEPSIA, uma análise das potenciais vulnerabilidades que possam existir em sistemas de agentes como é o sistema em questão e a especificação de requisitos necessários para que o modelo da arquitetura a ser definida.

5 Recomendações de Segurança

As atividades de *e-procurement* com que o sistema Multi-Agentes do DEEPSIA trata são de caráter estritamente confidencial. O conhecimento pela concorrência de quais os produtos que uma empresa está buscando no mercado pode fornecer-lhe informações valiosas sobre as tendências de mercado ou mesmo de um concorrente em especial. Assim, é de todo o interesse que a informação e mesmo do ato de busca da informação sejam protegidos de acessos alheios à empresa. (Pimentão, 2002).

De acordo com Murray (2000), um ponto de vulnerabilidade está no nível da interface com o usuário, considerando que as atuais características das aplicações *web* podem ser protegidas com protocolos pouco confiáveis, como o SSL (*Secure Sockets Layer*), por exemplo. No entanto, a principal falha de segurança está num nível inferior, na forma como as mensagens circulam em claro, em KQML (*Knowledge Query and Manipulation Language*) [14], entre os componentes móveis do sistema.

De acordo com as necessidades de cada módulo do sistema e de cada interface, são necessários níveis específicos de segurança, com as seguintes características, segundo Spafford (1996):

- Sigilo: proteção contra leitura ou cópia não autorizada;
- Integridade dos dados: proteção contra alterações de informações de formas não autorizadas;
- Disponibilidade: proteção dos serviços para que eles não se degradem ou fiquem indisponíveis sem autorização;
- Consistência: garantia de que o sistema se comporte da maneira com se é esperado;
- Controle: trata de regular o acesso ao sistema somente a entes autorizados;
- Auditoria: da mesma forma que o acesso não autorizado deve ser monitorado, o acesso regular deve sofrer auditagem, evitando atos maliciosos ou erros.

A seguir, serão apresentados os tipos de ataques que comprometem a segurança e que, de alguma forma, atingem o nível de segurança desejado para um sistema de multi-agentes que lidam com informação crítica, especificamente neste caso, dos agentes do sistema DEEPSIA.

---

[14] *What is KQML*: http://www.cs.umbc.edu/kqml/whats-kqml.html

5.1 Tipos de Ataques e Ameaças

Para apresentar os principais tipos de ataques e ameaças aos sistemas de agentes móveis, um modelo será utilizado (figura 2) Este modelo é bastante simples e nele o sistema de agentes é composto apenas pelos dois elementos do sistema multi-agentes: os agentes móveis e servidores de agentes. (Uto, 2001)

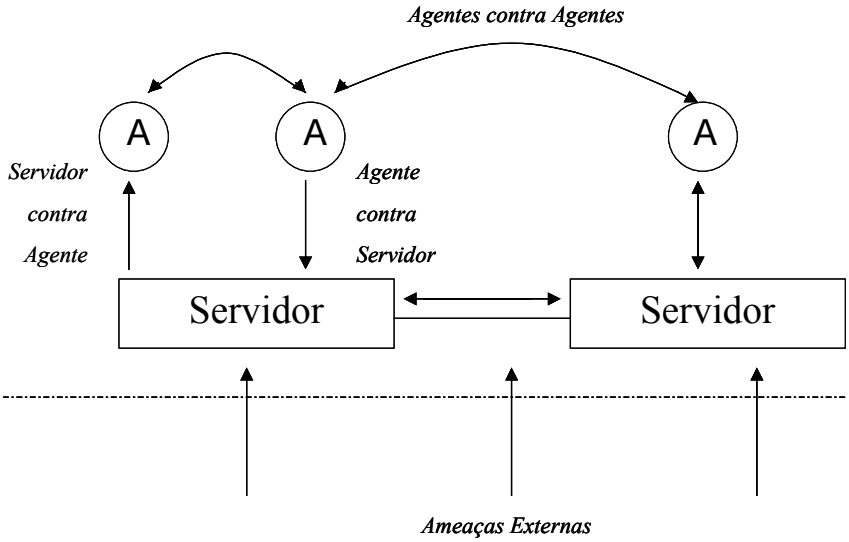

Figura 2: Os principais tipos de ataques e ameaças ao sistema multi-agentes

Os agentes realizam alguma tarefa específica em nome de um usuário e podem transportar dados e instruções de execução. Para tal, eles podem migrar de forma autônoma para os servidores que ofereçam os recursos necessários à sua execução. A figura 2 também destaca os módulos externos ao sistema e que podem ser alvos de ataque.

A representação de um ataque, neste diagrama, é feita através de setas unidirecionais. A cada uma das setas foi atribuído um número que indica umas das classes de ataques e ameaças possíveis, de acordo com a classificação no NIST [15] (*National Institute of Standards and Technology*), por Jansen (1999).

- Ameaças do servidor contra o agente
- Ameaças do agente contra o servidor
- Ameaças de um agente contra outro agente
- Ameaças externas contra o sistema de agentes

Nas subseções seguintes será detalhada cada uma das classes de ataques.

[15] NIST (*National Institute of Standards and Technology*): http://www.nist.gov

5.1.1 Ameaças de Servidor Contra Agente

Um agente em execução sempre está exposto aos ataques de qualquer natureza contra o seu servidor. Caso nenhum mecanismo de proteção seja utilizado pelo agente, tanto o seu código em execução como os seus dados podem ser completamente acessados pelo servidor, que pode, então, alterá-lo por um ataque ou uso indevido.

Como geralmente parte do estado de um agente precisa ser modificado para armazenar resultados intermediários de suas instruções, acredita-se, segundo Farmer *et al.* (1996), não ser possível garantir que um agente não será alterado maliciosamente. Essa classe de ameaças compreende os seguintes problemas:

- Personificação - um servidor pode personificar outro servidor, fazendo-se passar como uma terceira parte confiável, para extrair informações de agentes que tinham como destino o servidor original;

- Negação de serviço – quando servidor recebe um agente, espera-se que este seja executado corretamente e no tempo determinado. Um servidor malicioso pode recusar uma solicitação feita por um agente, completar a tarefa em um tempo inaceitável ou finalizar o agente abruptamente;

- *Eavesdropping* [16] – o atacante pode interceptar o canal de comunicação estabelecido entre dois agentes móveis ou obter acesso aos servidores onde os agentes são executados, obtendo acesso a toda informação não cifrada contida nos agentes visitantes. Assim, um servidor malicioso pode obter informações valiosas que podem, por exemplo, serem usadas como espionagem industrial por um concorrente;

- Alteração – um agente pode ter seus dados ou suas instruções alteradas por algum servidor malicioso. Assim, mecanismos que permitam a verificação da integridade dos dados dos agentes devem ser providos aos servidores confiáveis. De acordo com Farmer *et al.* (1996), ainda não há uma solução definitiva para que um agente identifique alguma alteração não autorizada em seu estado.

5.1.2 Ameaças de Agente Contra Servidor

Nesta classe de ataques, os agentes procuram explorar as vulnerabilidades dos servidores:

---

[16] Do inglês *eavesdrop*, v. espreitar, escutar às escondidas, espiar. (Fonte: Dicionário Melhoramentos, http://www.uol.com.br/dicionario)

- Personificação – neste tipo de ataque, um agente tenta se passar por outro para utilizar recursos aos quais não tenha acesso autorizado ou com objetivo de não ser responsabilizado por ações específicas;

- Negação de serviço – esta situação ocorre quando os agentes consomem excessivamente os recursos de um servidor, podendo, em alguns casos, tornar um serviço ou o próprio servidor indisponível;

- Acesso não autorizado – o acesso não autorizado a recursos de servidores pode comprometer o funcionamento e a segurança de um sistema multi-agentes, incluindo os agentes e servidores. Neste caso, por exemplo, um agente de busca externo ao sistema poderia realizar uma busca em categorias de produtos que não foi autorizado previamente, obtendo informações confidenciais de bancos de dados de concorrentes.

5.1.3 Ameaças de Agente Contra Agente

Esta categoria compreende os ataques em que um agente malicioso tenta explorar vulnerabilidades de outros agentes e estão baseados nos serviços de comunicação entre os agentes e os servidores.

- Personificação – um agente pode tentar se passar por outro para obter informações confidenciais de um terceiro, como resultados de uma busca no DEEPSIA ou para prejudicar o funcionamento de uma busca em execução por um concorrente. Pode-se realizar este ataque também com o intuito de corromper a autenticação de um agente válido, impedindo novas consultas válidas;

- Negação de Serviço – agentes hostis podem realizar um ataque de negação de serviço contra outros agentes, enviando-lhes mensagens desnecessárias ou corrompidas. Independentemente de se processar ou descartar as mensagens inválidas, o tempo de processamento será desperdiçado pela vítima;

- Repúdio – este ataque é caracterizado quando um agente que participa de uma transação ou comunicação nega ter realizado as ações executadas. Nestes casos, uma certificação dos agentes nos servidores ou entre eles é necessária;

- Acesso não autorizado – sem um controle de acesso a recursos dos agentes, um agente hostil pode manipular dados ou as instruções de outro agente em execução em um servidor comum.

5.1.4 Outras Ameaças Contra Sistema de Agentes

Nesta classe de ataques, são descritas formas alternativas de exploração das vulnerabilidades do sistema multi-agentes. São exploradas as falhas nos protocolos de comunicação dos agentes, no caso sistema multi-agentes do DEEPSIA, KQML, ou nos protocolos da rede de comunicação, no caso, a Internet.

- Personificação – um agente se fazer passar por outro agente para obter acesso a um recurso ou a um serviço oferecido por um servidor, que também pode assumir outra identidade para enganar agentes e outros servidores;

- Acesso não autorizado – um agente ou alguma entidade externa pode tentar acessar recursos de forma não autorizada, obtendo acessos privilegiados. Com isso, seria possível destruir dados do servidor, comprometendo seu funcionamento;

- Negação de serviço – é possível realizar estes ataques acessando remotamente os serviços dos servidores ou direcionando-os aos sistemas operacionais e aos protocolos de comunicação;

- *Eavesdropping* – um atacante pode "escutar" o canal de comunicação entre servidores para obter informações de agentes em trânsito ou mensagens trocadas entre os agentes;

- Repetição – nesta forma de ataque, uma entidade maliciosa intercepta um agente ou uma mensagem, realiza a cópia e então, clona ou retransmite a mensage.

5.2 Especificação de requisitos

Os requisitos funcionais básicos para a definição de um modelo de comunicação segura no sistema DEEPSIA usando KQML são baseados na análise de outros modelos de segurança como: PEM (*Privacy Enhanced Mail*)[17], CORBA (*Common Object Request Broker Architecture*) [18], DCE (*Distributed Computing Environment*) [19], além de outros modelos para comunicação segura por KQML previamente definidos. (Thirunavukkarasu *et al.*, 1995) (Mayfield, 1995) Dentre as características básicas que a arquitetura deve possuir, pode-se listar:

---

[17] *Privacy Enhancement for Internet Electronic Mail:* http://www.ietf.org/rfc/rfc1421.txt

[18] *CORBA Security Service*: http://www.omg.org/technology/documents/formal/omg_security.htm#corba

[19] *Security in the Distributed Computing Environment* : http://www-3.ibm.com/software/network/dce/library/redbooks/sg244949/4949c112.htm

- Autenticação: Os agentes devem ser capazes de provar as suas identidades e de verificarem a identidade de outros agentes;
- Preservação de integridade da mensagem: Os agentes devem ser capazes de detectar alteração intencional ou não nas suas mensagens;
- Proteção de privacidade: A arquitetura deve prover mecanismos para que os agentes troquem informações confidenciais;

Outras características de implementação desejadas deste modelo são:

- Independência da camada de transporte: A arquitetura de segurança não deve depender de questões de implementação da camada de transporte da rede pela qual as mensagens serão enviadas. Este fator permite que sejam usadas redes de comunicação heterogêneas e que o modelo possa ser usado aplicado em novos protocolos de transporte que no futuro venham a ser utilizados;
- Autenticação por agentes sem capacidades de criptografia: Um agente que não possua capacidades de codificar e decodificar uma mensagem deve ser capaz de verificar a autenticidade de quem o enviou uma mensagem;
- Suporte a grande variedade de sistemas criptográficos: Os agentes devem ser capazes de usar diversos algoritmos criptográficos, no entanto, dois agentes, ao se comunicarem, devem estabelecer um único algoritmo para a comunicação. A arquitetura definida também não deve ser dependente de um conjunto fixo de algoritmos criptográficos.

Há outras características desejadas para a arquitetura em questão e que não estão sendo consideradas nesta implementação, no entanto, são sugestões para implementações futuras e mais avançadas deste modelo:

- Não repúdio das mensagens: Um agente deve ser responsável pelas mensagens que ele envia ou recebe, de modo que não seja possível negar o envio ou recebimento destas;
- Prevenção contra captura ou duplicação de mensagens: Um agente hostil não deve ser capaz de extrair somente uma informação de autenticação de uma mensagem ou de duplicar uma mensagem em trânsito, o que pode permitir um ataque de personificação ou repetição de uma mensagem válida, por exemplo.

No capítulo seguinte será apresentada a estrutura básica da linguagem e protocolo de comunicação entre agentes que é usada no DEEPSIA, a KQML.

6 A Linguagem KQML

KQML (*Knowledge Query and Manipulation Language*) é uma linguagem e protocolo de comunicação, de alto nível, por troca de mensagens e que foi desenvolvida dentro do "*Knowledge Sharing Effort*" patrocinado pelo DARPA, em 1993 (Finin *et al.*, 1993). KQML é independente do mecanismo de transporte (TCP/IP, SMTP, etc), independente da linguagem conteúdo (KIF [20] ou SQL [21], por exemplo), e independente da ontologia assumida pelo conteúdo. (Finin *et al.*, 1997)

Conceitualmente, uma mensagem KQML é organizada em três partes: conteúdo, comunicação e mensagem. O conteúdo é a parte que transporta a mensagem em si, podendo ser transportada com qualquer linguagem de representação, seja em texto ou em modo binário. Toda implementação de KQML ignora a porção de conteúdo da mensagem, exceto a parte que determina onde a mensagem termina.

A parte denominada mensagem é a parte central de um pacote KQML. Esta parte determina os tipos de interação que um agente pode ter com outro. A função principal desta parte é identificar o protocolo de rede que vai ser usado para entregar a mensagem e a ação de comunicação ou *performative* [22] com a qual o remetente está enviando na mensagem. Ou seja, a parte de mensagem define que operação deve ser executada com o conteúdo transportado.

Como o conteúdo não é interpretado, a parte denominada mensagem define outras opções, tais como: a linguagem em que está representado o conteúdo, a ontologia assumida, entre outras. Estas opções permitem as implementações de KQML analisar, rotear e entregar corretamente mensagens cujo conteúdo é inacessível.

A figura 3 mostra um exemplo de mensagem KQML (Thirunavukkarasu *et al.*, 1995).

---

[20] *Knowledge Interchange Format* (KIF): http://logic.stanford.edu/kif/kif.html

[21] SQL.org: http://www.sql.org

[22] *performative:* uma mensagem em linguagem KQML. (Finin *et al.*, 1997)

```
(ask-one
  :sender joe
  :content (PRICE IBM ?price)
  :receiver stock-server
  :reply-with ibm-stock
  :language LPROLOG
  :ontology NYSE-TICKS)
```

(a)

```
(tell
  :sender stock-server
  :content (PRICE IBM 14)
  :receiver joe
  :in-reply-to ibm-stock
  :language LPROLOG
  :ontology NYSE-TICKS)
```

Figura 2: Exemplos de mensagens em KQML: (a) uma consulta do agente *joe* sobre o preço das ações da IBM e (b) a resposta do servidor

A tabela 1 apresenta os tipos de *performatives* ou ações disponíveis em KQML para os agentes. (Silva, 2001)

Consulta simples	`evaluate, ask-if, ask-in, ask-one, ask-all, ...`
Consulta composta	`stream-in, stream-all, ...`
Resposta	`reply, sorry, ...`
Afirmação genérica	`tell, achieve, cancel, untell, unachieve, ...`
Gerador	`stanby, ready, next, rest, discard, generator, ...`
Anúncio e definição de capacidades	`advertise, subscribe, monitor, import, export, ...`
Rede	`register, unregister, forward, broadcast, ...`

Tabela 1: Categorias básicas de performatives de KQML

Embora KQML tenha um conjunto de *performatives* reservadas, este não é restrito e pode ser expandido de acordo com a necessidade. Um agente que use KQML, por exemplo, pode escolher apenas algumas *performatives* para interpretar e uma comunidade de agentes pode escolher utilizar *performatives* adicionais, caso isto seja previamente definido. No entanto, se uma *performative* é reservada, ela deve ser implementada da forma padrão.

Como um conjunto de *performatives* pode ser definido de acordo com a necessidade, neste caso, a especificação de um conjunto que possa garantir a segurança dos agentes e das mensagens que neles são trafegadas é o objetivo principal.

O capítulo seguinte apresenta a arquitetura de segurança do sistema de Multi-Agentes do DEEPSIA e as adições sugeridas a KQML para que a implementação deste modelo seja possível.

7 A Arquitetura de Segurança

A arquitetura proposta é baseada em algumas técnicas de criptografia de dados (Schneier, 1996) e está em conformidade com a característica assíncrona da linguagem KQML, ou seja, uma mensagem codificada pode ser automaticamente verificada e não deve haver necessidade de um mecanismo de desafio-resposta para verificar a mensagem após seu envio, segundo Thirunavukkarasu *et al.* (1995).

Esta proposta apresenta um modelo básico de segurança, que inclui características de autenticação das partes envolvidas na comunicação, integridade e privacidade das mensagens. Funções adicionais como suporte contra não repúdio do emissor da mensagem, proteção contra ataques de *replay* [23] e freqüente troca de chaves de criptografia para evitar ataques de cifra (*ciphertext-only attacks*) [24] não são implementadas neste modelo, no entanto, são características desejadas para um modelo assíncrono e avançado de troca de mensagens.

7.1 Conceitos de criptografia

A seguir serão explicadas as técnicas de criptografia que serão utilizadas por esta arquitetura (Schneier, 1996) e os novos *performatives* e parâmetros que devem ser introduzidos em KQML para permitir que os requisitos necessários sejam concluídos. (Thirunavukkarasu *et al.,* 1995).

Chaves criptográficas: um agente que implementa a arquitetura de segurança proposta deve ter uma chave principal K_a (*master key*) que pode ser usada para comunicação com outros agentes. Esta chave pode ser baseada em um sistema criptográfico simétrico ou assimétrico (usando chaves públicas e privadas, por exemplo) (Diffie *et al.*, 1976). Se uma chave simétrica for utilizada, a sugestão é que seja usada também, além da chave principal, uma chave específica para cada agente $K_{a1,a2}$, de modo a prover um nível maior de privacidade e de autenticação.

Se mais de dois agentes compartilham uma única chave principal, um destes dois pode se mascarar como o seu par ou escutar as comunicações entre outros agentes que fazem uso da mesma chave criptográfica, ou seja, se uma chave principal é compartilhada, o nível de

---

[23] *replay attack*: ataque no qual uma transmissão de dados válida é maliciosamente repetida, geralmente por um adversário, que a intercepta e retransmite, possivelmente como parte de um ataque de personificação. (Fonte: http://www.linuxsecurity.com/dictionary/)

[24] *ciphertext-only attack*: técnica de análise de código critografado na qual o analista tenta determinar a chave utilizada para codificação, com basicamente o conhecimento dos dados interceptados. (Fonte: http://www.linuxsecurity.com/dictionary/)

segurança é diretamente proporcional ao grau de confiança entre os agentes que a compartilham. Se um agente não compartilha sua chave principal com outro agente, ele pode usar esta chave ou pode recorrer a um servidor de autenticação que gerará a sua nova chave.

Os agentes devem usar chaves diferentes em cada sentido de suas mensagens, ou seja, se uma chave $K_{a1,a2}$ é criada por *a1* e é usada para que uma mensagem seja enviada para *a2*, uma chave $K_{a2,a1}$ deve ser criada para que a resposta de *a2* para *a1* seja devolvida. Caso seja usado um método de criptografia assimétrico, um par de chaves é suficiente, pois um agente pode usar a chave pública de seu par para codificar a mensagem e também usar sua chave pessoal para assinar a mensagem e provar sua identidade.

Assinatura da mensagem: cada mensagem segura usando esta arquitetura deve possuir uma assinatura (ou *digest*) associada a ela. A assinatura é calculada usando uma função *hash* [25], como por exemplo, MD5 ou SHA [26]. Esta função calcula uma impressão digital da mensagem, de forma semelhante a uma seqüência de verificação de dados (*checksum*).

O agente que envia a mensagem então codifica esta assinatura usando a chave criptográfica atual e a anexa à mensagem e o agente que a recebe usa esta assinatura para verificar a identidade de quem a enviou e a integridade da mensagem.

Alguns outros conceitos como chave criptográfica de sessão e identificação da mensagem não são considerados nesta arquitetura por não serem necessários para as funções básicas de segurança definidas (autenticação, privacidade e integridade das mensagens).

7.2 Adições ao KQML

Com o objetivo de implementar esta arquitetura de segurança, são propostos alguns novos *performatives* para a linguagem KQML, alguns novos parâmetros e também algumas modificações para a ontologia padrão dos agentes. (Thirunavukkarasu *et al.,* 1995)

7.2.1 Considerações sobre a ontologia dos agentes

É considerado que agentes que se comunicam por KQML usam uma ontologia básica que os provê um conjunto de classes, atributos e relacionamentos para comunicação eficiente. (Finin *et al.*, 1997) Assumindo esta ontologia, esta arquitetura introduz uma nova subclasse de agentes chamada *authenticator* e um novo relacionamento chamado `key/5`, que descreva a

[25] Função *hash* é uma tal que recebe um valor de entrada e retorna um valor de saída, de alta precisão de tamanho constate para qualquer tamanho de entrada, deve ser sem colisão, sem função inversa e de baixa complexidade computacional. (Fonte: http://www.rsasecurity.com/rsalabs/faq/)

chave (*key*) de comunicação que será utilizada por cada agente.

```
(key  <sending-agent>
      <receiving-agent>
      <master-key?>
      <key-type>
      <encrypted-key>)
```

Uma instância desta relação especifica uma chave que o agente que envia a mensagem irá usar em uma comunicação segura com seu agente par. Se o seu terceiro argumento é verdadeiro, então a chave em questão é uma chave principal (*master key*), caso contrário, ela é uma chave de sessão.

Se a identificação do agente de destino (`receiving-agent`) for uma variável, então esta chave será usada para codificação para todos os agentes em comunicação. Este valor também pode ser alterado caso o método de criptografia seja por um par de chaves pública e pessoal (criptografia assimétrica).

Assume-se também que a ontologia que representa o endereço dos agentes (*`address/3`*) é a seguinte:

```
(address     <agent>
             <transport>
             <transport-address>)
```

Instâncias desta relação definem o endereço de transporte para o agente do primeiro argumento (`agent`), como nestes exemplos:

```
(address     agente1
             smtp
             143.107.231.231)
(address     agente2
             tcpip
             (143.107.231.231 8080))
```

Estes endereços de transporte são conhecidos por agentes especiais, geralmente os que são responsáveis por resolução de nomes de máquinas (*agent name servers*) ou agentes de autenticação (*authenticator agent*).

---

[26] Message Digest #5 e Secure Hash Algorithm: http://www.linuxsecurity.com/resource_files/documentation/hacking-dict.html#hash

7.2.2 Adições ao KQML

Novos parâmetros e *performatives* são necessários em KQML para que a arquitetura de segurança proposta possa sem implementada. A seguir eles serão listados e explanados:

Novos parâmetros

`:auth-digest (<digest-type> <encrypted-digest>)`

O campo `<digest-type>` especifica a função *hash* que será usada para calcular a assinatura da mensagem (MD5 ou SHA, por exemplo) e `<encrypted-digest>` é o campo que deve conter a assinatura codificada pelo parâmetro `:auth-key`. O uso deste parâmetro evita um ataque como seqüestro de mensagem, provê autenticação do remetente da mensagem e garante a sua integridade.

`:auth-key (<bool> <key-type> <encrypted-key>)`

Este parâmetro especifica a chave utilizada para codificar qualquer `:auth-digest`, ou seja, para a codificação ou assinatura de uma mensagem segura desta arquitetura.

Novos *performatives*

`auth-link`

O remetente deseja se autenticar para um agente de destino e através desta *performative*, ele solicita uma chave de sessão para tal.

`auth-challenge`

O remetente requisita a identidade de um destinatário em resposta a um `auth-link`. Este remetente então codifica uma seqüência de dados aleatória usando sua chave principal $K_{r,d}$ ou K_r e a envia como `:content`, ou seja, como uma mensagem.

`auth-private`

Quando um remetente está enviando uma mensagem confidencial para seu destinatário, o parâmetro `:content` contém a mensagem codificada e o parâmetro `:auth-key` especifica a chave criptográfica. O parâmetro `:auth-digest` deve estar presente para autenticar a identidade do remetente da mensagem.

7.3 O modelo de segurança

Uma implementação que tenha como objetivo a segurança básica na troca de mensagens KQML, ou seja, suportar autenticação, verificação de integridade e privacidade de dados, deve seguir o protocolo definido a seguir. É importante destacar que se caso seja utilizado um algoritmo assimétrico de criptografia, o modelo também suporta o não repúdio

de remetentes de mensagens seguras.

Quando o agente `origem` envia uma mensagem segura para o agente `destino`, ele deve calcular uma assinatura da mensagem e codificá-la usando a chave criptográfica principal, neste caso, o valor do parâmetro `auth-key`.

```
<performative>
  :sender origem
  :receiver destino
  :auth-key K
  :auth-digest (<digest-type><encrypted-digest>)
  ...
```

De forma alternativa, se o agente `origem` precisa enviar uma mensagem confidencial para o agente `destino`, ele pode codificar e anexá-la em uma mensagem `auth-private`, como a seguir:

```
auth-private
  :sender origem
  :receiver destino
  :auth-key K
  :auth-digest (<digest-type><encrypted-digest>)
  :content <encrypted-KQML-message>
  ...
```

Este modelo pode ser usado quando o remetente da mensagem não conhece previamente o seu destino, ou seja, para mensagens que devem ser enviadas por difusão (*broadcast*) ou roteadas por algum agente ou servidor com este fim específico. É importante destacar que esta modelagem não impede ataques de *replay* às mensagens em trânsito e obriga os agentes a usarem uma chave criptográfica única e que deve ser determinada antes da comunicação segura se iniciar, negociação esta feita por desafio-resposta ou através de criptografia assimétrica.

Na mensagem anterior, o parâmetro `:auth-digest` pode ser utilizado para verificar a integridade da mensagem, para autenticar o seu remetente e para garantir o seu não repúdio de origem (neste caso, se uma chave assimétrica for aplicada).

Se a mensagem for alterada, a assinatura da mensagem não irá concordar com o valor do parâmetro `:auth-digest`. Se a assinatura da mensagem foi codificada com a chave criptográfica principal do agente `:origem`, somente ele e os agentes com os quais ele compartilha sua chave poderão ter gerado esta mensagem. Caso esta chave principal seja

assimétrica, somente o próprio remetente poderia ter gerado esta mensagem, já que somente este agente tem acesso a chave pessoal que é usada para codificação.

É importante observar que somente é possível verificar a identidade do gerador da mensagem (ou seja, a mensagem que foi codificada pelo agente `:origem`), ou seja, ela pode ser uma mensagem legítima e duplicada já enviada pelo mesmo agente ou uma mensagem originada de um ataque de *replay*.

7.4 Limitações do modelo proposto

Este modelo de segurança proposto para a comunicação de agentes por KQML possui limitações, que serão aqui enumeradas. É importante destacar que algumas limitações recaem sobre o modelo simplificado de comunicação e dos requisitos apresentados (seção 5.2) e outras são de considerações de implementação deste modelo:

- Credenciais – Este modelo não provê mecanismos de troca de credenciais, ou seja, é possível que um agente nomeie outro para executar a tarefa em seu nome;
- Não repúdio de recebimento – Este modelo não suporta o não repúdio de recebimento de mensagens. Esta característica é desejável, no entanto, não pode ser implementada (Thirunavukkarasu *et al.*, 1995) pela própria natureza assíncrona da troca de mensagens KQML e sim definida pela aplicação que usa o protocolo de comunicação;
- Mensagens para destinos desconhecidos – Uma característica de KQML é permitir o uso de aplicações intermediárias que fazem o roteamento das mensagens entre os agentes. No entanto, uma identificação específica por uma central de identificação pode ser necessária em um modelo avançado de segurança para que tentativas de roteamento de mensagens para destinos inexistentes sejam evitadas;
- Introdução de estado – É desejável que em algumas transações seguras, o estado de alguns agentes seja mantido, para, por exemplo, registrar identificações de mensagens numeradas e assinadas digitalmente;
- *Crypto awareness* – Neste modelo, um agente não é capaz de verificar a autenticidade de um remetente ou de uma mensagem sem que ele possua capacidade de criptografar uma mensagem;
- Considerações de envio de mensagens – Neste modelo, o envio de mensagens deve ser em ordem e o tratamento desta consideração devem ser feitos pelos agentes, pois é considerada como tratada pelo KQML;

- Uso de APIs padronizadas – Este modelo deve ser ampliado para permitir o uso de APIs criptográficas padrão e recomendadas de acordo com a aplicação em que os agentes serão utilizados e com a legislação vigente para tratamento destas tecnologias de criptografia. (Schneier, 1996)

7.5 Continuação do trabalho

Este trabalho é resultado de um projeto de pesquisa para especificação e implementação de segurança no sistema DEEPSIA. Os resultados aqui apresentados são intermediários e devem ser refinados até o fim do projeto, que será em Março de 2003.

Há alguns pontos que estão em estudo e em desenvolvimento e utilizarão este modelo como base para a continuidade no desenvolvimento do sistema DEEPSIA e da colaboração da USP ao DEEPSIA *Consortium* na Europa.

- Implementação do modelo de segurança na comunicação dos agentes do sistema DEEPSIA;
- Especificação de melhorias e tratamento de limitações deste modelo para um modelo avançado de segurança KQML, de acordo com as necessidades do DEEPSIA *Consortium*;
- Estudos e avaliação de aplicabilidade de modelagem padrão FIPA (*Foundation for Intelligent Physical Agents*) [27] em uma nova implementação do sistema Muli-Agentes DEEPSIA.

[27] O modelo FIPA para comunicação de agentes suplantou o padrão KQML e já é sugerido como padrão para as novas implementações de multi-agentes. Para mais informações, consulte FIPA (*Foundation for Intelligent Physical Agents*): http://www.fipa.org

8 Referências

Ambrósio, D. R. *Alternativas de Implementação de Reconhecimento de Padrões para Agentes Móveis em Ambiente de Segurança Computacional.* São Carlos, 2002. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo.

Bernardes, M. C. *Avaliação do uso de agentes móveis em segurança computacional*. São Carlos, 1999. 105 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Bharat, K.; Broder, A., *A technique for measuring the relative size and overlap of public web search engines*, in Proc. of the 7th World-Wide Web Conference (WWW7), 1998.

Bond, A.; Gasser, L., *Readings in Distributed Artificial Intelligence*, Morgan Kaufman., 1988.

Bonifácio Jr., J. M. *Sistemas de segurança distribuído: integração de firewalls com sistemas de detecção de intrusão*. São Carlos, 1998. 79 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Cansian, A. M. *Desenvolvimento de um sistema adaptativo de detecção de intrusos em redes de computadores*. São Carlos, 1997. 153 p. (Tese de Doutorado). Instituto de Física de São Carlos, Universidade de São Paulo.

Cicilini, R. *Desenvolvimento de um agente SNMP para plataformas rodando DOS.* São Carlos, 1994. 107 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Diffie, W.; Hellman, M. E. *New directions in cryptography*, IEEE Transactions on Information Theory 22 (1976), 644-654.

Finin, T.; Labrou, Y. *A Proposal for a new KQML Specification*. University of Maryland Beltimore Count - UMBC, 1997.

Finin, T.; Weber, J.. *Specification of the KQML Agent-Communication Language*. The DARPA Knowledge Sharing Initiative, 1993.

Garção, A. S.; Sousa, P. A.; Pimentão, J. P.; Santos, B. R.; Blazquéz, V.; Obratanski, L. 2002. *Annex to Deepsia's Deliverable 4 – System Architecture*. January. 135p. Report. IST PROJECT-1999-20483.

Goularte, R. *Utilização de meta-dados no gerenciamento de acesso a servidores de vídeo*. São Carlos, 1998. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo.

Jansen, W.; Karygiannis, T. *Mobile Agent Security*. Technical report, National Institute of Standards of Technology, 1999.

Lee, K.; Mansfield Jr., W. *A framework for controlling Cooperative Agents*, Bellcore, IEEE Computer., Julho, 1993.

Lieira, J. F. *Utilização de áudio e vídeo em sistemas gerenciadores de redes de computadores*. São Carlos, 1995. 111 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Martins Jr. J. *Classificação de Páginas na Internet*. São Carlos, 2002. Monografia de Qualificação (Mestrado) - Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo.

Mayfield, J.; Finin, T. *A Security Architecture for Agents Communication Languages*. University of Maryland Beltimore Count - UMBC, 1995.

Moraes, S. *Voz em sistemas computacionais: projeto e implementação de módulos de processamento de voz em gerenciamento de redes*. São Carlos, 1995. 103 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Moroshita, F. T. *Uma avaliação evolutiva dos protocolos de gerenciamento da Internet: SNMPv1, SNMPv2 e SNMPv3*. São Carlos, 1997. 68 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Murray, E. *SSL Server Security Survey*. Disponível on-line em <http://www.lne.com/ericm/papers/ssl_servers.html> Acesso em 1/06/2002.

Oda, C. S. *Desenvolvimento de um sistema monitor gráfico baseado em protocolo de gerenciamento SNMP*. São Carlos, 1994. 111 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Pereira Filho, S.F. *Avaliação para ambientes servidores para agentes móveis*. São Carlos, 2001. 100 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Pimentão, J. P. *Infra-estrutura de segurança do projeto DEEPSIA*, Universidade Nova de Lisboa, Janeiro, 2002.

Reami, E. R. *Especificação e prototipagem de um ambiente de gerenciamento de segurança apoiado por agentes móveis*. São Carlos, 1998. 82 p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Schneier, B. *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., 2ª ed. Janeiro, 1996.

Silva, F. S. C.; Meneses, E. X. *Integração de Agentes de Informação.* JAIA - Jornada de Atualização em Inteligência Artificial, IME – USP. Disponível on-line em <http://www.ime.usp.br/~eudenia/jaia/> Acesso em 3/06/2002. São Paulo, 2001.

Simon, H. A. *Search and Reasoning in Problem Solving*. Artificial Inteligence, n. 21, pp. 7-30, 1983.

Spafford, G.; Garfinkel, S. *Practical UNIX & Internet Security*. O'Reilly & Associates. 2ª ed. Abril, 1996.

Tavares, D. M. *Avaliação de técnicas de captura para sistemas detectores de intrusão.* São Carlos, 2002. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Thirunavukkarasu, C.; Finin, T.; Mayfield, J. *Secret Agents – A Security Architecture for the KQML Agent Communication Language.* (Versão *draft* submetida para o evento CIKM'95). University of Maryland Beltimore Count - UMBC, 1995.

Uto, N.; Dahab, R. *Segurança de Sistemas de Agentes Móveis* Anais do III Simpósio de Segurança em Informática, São José dos Campos, 2001.

Apêndice A: Links

Neste apêndice são apresentados alguns links de sites para os interessados em assuntos relacionados diretamente ao desenvolvimento deste projeto, com uma breve descrição do conteúdo que será encontrado em cada um deles. A lista é apresentada em ordem alfabética e todos os *links* foram conferidos em 3 de Junho de 2002.

CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico	*http://www.cnpq.br*
DEEPSIA *Consortium*	*http://www.deepsia.com*
ECCMA - *Electronic Commerce Code Management Association*	*http://www.eccma.org*
Empresa Atlante (Espanha)	*http://www.atlante.com*
Empresa ComArch (Polônia)	*http://www.comarch.com*
Empresa Zeus (Grécia)	*http://www.zeusconsult.gr*
Forrester Research	*http://www.forrester.com*
IDC - *International Data Corporation*	*http://www.idc.com*
Information Society Technologies Programme	*http://www.cordis.lu/ist*
Instituto de Ciências Matemáticas e de Computação	*http://www.icmc.usp.br*
KQML - *Knowledge Query and Manipulation Language*	*http://www.cs.umbc.edu/kqml/whats-kqml.html*
Laboratório Intermídia	*http://intermidia.icmc.usp.br*
LinuxSecurity.doc Dictionary	*http://www.linuxsecurity.com/dictionary*
Módulo Security Solutions	*http://www.modulo.com.br*
NUMA - Núcleo de Manufatura Avançada	*http://www.numa.org.br*
Ontology.org Portal	*http://www.ontology.org*
RSA FAQ	*http://www.rsasecurity.com/rsalabs/faq*
SEBRAE-SP – Serviço de Apoio às Micro e Pequenas Empresas de São Paulo	*http://www.sebraesp.com.br*
SSL Weakness	*http://www.lne.com/ericm/papers/ssl_servers.html*
Universidade Nova de Lisboa UNINOVA (Portugal)	*http://www.uninova.pt*
Université Libre de Bruxelles (Bélgica)	*http://www.iihe.ac.be/stc*
University of Sunderland (Inglaterra)	*http://voyager.sunderland.ac.uk*
Webopedia On-line	*http://www.webopedia.com*
Yankee Group	*http://www.yankeegroup.com*

Apêndice B: Lista de Abreviaturas

B2B	*Bussiness to bussiness*
CA	*Crawler Agent*
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DC	*Dynamic catalogue*
DEEPSIA	*Dynamic on-linE IntErnet Purchasing System based on Intelligent Agents*
DNS	*Domain Name Service*
ECCMA	*Electronic Commerce Code Management Association*
FA	*Facilitator Agent*
HA	*Human Agent*
ICMC	Instituto de Ciências Matemáticas e de Computação
IDC	*International Data Corporation*
IST	*Information Society Technologies*
KQML	*Knowledge Query and Manipulation Language*
MA	*Miner Agent*
MAS	*Multi-Agent System*
NUMA	Núcleo de Manufatura Avançada
PIA	*Portal Interface Agent*
PMEs	Pequenas e Médias Empresas (vide também SMEs)
SEBRAE	Serviço de Apoio às Micro e Pequenas Empresas
SMEs	*Small and Medium Enterprises* (vide também PMEs)
SSL	*Secure Socket Layer*
WCA	*Web Crawler Agent*
WTC	*World Trade Center*

Apêndice C: Histórico do Grupo de Pesquisa Intermídia

O grupo de pesquisas do Laboratório Intermídia do Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP) tem aplicado estudos ao desenvolvimento de projetos em duas frentes importantes na área da computação.

A primeira iniciou seus trabalhos na década de 90 com projetos relacionados ao gerenciamento de redes de computadores (Oda, 1994; Cicilini, 1994; Lieira, 1995; Moraes, 1995; Moroshita, 1997) e desenvolve atualmente pesquisas na área de segurança computacional com sistemas detectores de intrusões que usam redes neurais para o reconhecimento de padrões de ataques (Cansian, 1997; Bonifácio Jr. 1998) e aplicam a tecnologia de agentes móveis para gerenciamento da segurança (Reami, 1998) e verificação de anomalias (Bernardes, 1999). Pesquisas sobre ambientes servidores para agentes móveis (Pereira Filho, 2001) e tecnologias que podem conferir inteligência a tais agentes (Ambrósio 2002) bem como estudos para implementação de sistemas detectores de intrusão em dispositivos de segmentação de redes (*switches*) (Tavares, 2002) foram também desenvolvidos pelo grupo.

A segunda frente de trabalho desenvolve atualmente pesquisas em sistemas multimídia distribuídos, aplicando estudos em padrões para a representação de informações e meta-dados na identificação de fluxos de mídia contínua e técnicas adequadas para transmissão e distribuição de vídeo na Internet (Goularte, 1998).

A partir do segundo semestre de 2001, o Laboratório Intermídia juntamente com o Núcleo de Manufatura Avançada (NUMA) passou a integrar o projeto DEEPSIA, como cooperação internacional a partir do Brasil e com fomento do CNPq (**DEEPSIA – CNPq-68.0236/01-2**). A participação em tal projeto motivou o grupo pela instituição de uma nova frente de trabalho relacionada ao estudo e ao desenvolvimento de soluções otimizadas para sistemas de Comércio Eletrônico, como a aplicação de técnicas apropriadas à busca, classificação (Martins, 2002), armazenamento e recuperação de informações de produtos a serem utilizadas em processos de compra na *Web* e aplicação de pesquisas do grupo de Segurança da Informação nesta nova frente de trabalho.