Repositórios Seguros de Dados para Proteção de Agentes Móveis contra Plataformas Maliciosas

Michelle S. Wangham *, Joni da Silva Fraga , Rafael J. Deitos , Galeno A. Jung

[1]Departamento de Automação e Sistemas – Universidade Federal de Santa Catarina
Campus Universitário, C.P. 476 – CEP 88040-900 Florianópolis, SC

{wangham, fraga, deitos, galeno}@das.ufsc.br

Resumo. *Este trabalho define um esquema de segurança que visa diminuir os riscos que os agentes móveis com itinerário livre estão suscetíveis quando visitam plataformas maliciosas. O esquema é composto por três repositórios seguros de dados que visam garantir a integridade dos dados armazenados, sendo que dois deles oferecem ainda a confidencialidade dos dados.*

Abstract. *This work defines a security scheme that reduces the risks to which mobile agents are exposed when they visit malicious platforms. The scheme is composed of three secure data repositories that ensure integrity of stored data. In addition, two of these repositories also provide data confidentiality.*

1. Introdução

Um agente móvel pode ser definido como um agente de software que migra de um sítio para outro em uma rede heterogênea se executando de maneira autônoma no seu destino. A utilização de agentes móveis requer a necessidade de um ambiente computacional nas máquinas onde esses códigos irão ser executados — plataforma de agentes.

A habilidade para mover agentes (código + estado) em um sistema distribuído permite o desenvolvimento de serviços e aplicações mais flexíveis e dinâmicos quando comparado com o paradigma cliente-servidor [Vigna, 1998b]. Apesar das suas vantagens, o paradigma de agentes móveis introduz novas ameaças de segurança ao sistema. Devido a essas ameaças, os mecanismos de segurança devem ser projetados para proteger a infra-estrutura de comunicação, as plataformas de agentes e os próprios agentes móveis. Os mecanismos descritos neste artigo se concentram na problemática da **segurança dos agentes móveis** com itinerários livres, considerando sistemas distribuídos de larga escala.

Os ataques de plataformas maliciosas contra os agentes são os problemas de segurança mais difíceis de serem contornados e ainda sem solução adequada. Enquanto os mecanismos direcionados à proteção da plataforma são uma evolução direta dos mecanismos tradicionais que enfatizam medidas de prevenção ativa, mecanismos direcionados à proteção dos agentes correspondem normalmente a medidas de detecção. Isto ocorre devido ao fato de um agente ser completamente suscetível à plataforma e de ser difícil evitar a ocorrência de comportamentos maliciosos. Em [Chess, 1998], há afirmação de que uma plataforma que executa um dado agente tem total controle sobre as ações desse programa. Não há como prevenir a plataforma de analisar o programa, de mudar o estado do agente antes de executá-lo ou de observar seus resultados.

Este trabalho tem por objetivo definir um esquema, baseado em repositórios de dados, que visa diminuir os riscos que os agentes móveis com itinerário livre estão suscetíveis, quando estes visitam plataformas maliciosas, procurando definir protocolos de detecção de violações contra a integridade do

*Todos os autores têm o apoio financeiro do CNPq. Este trabalho foi parcialmente financiado pelos projetos: Cadeias de Confiança (CNPq 552175/01-3) e Instituto Fábrica do Milênio-IFM (MCT/CNPq).

agente. Além disso, para atender às necessidades específicas das aplicações, este esquema pretende ser flexível de modo que o mesmo possa ser especializado através da seleção de um subconjunto de repositórios.

2. Proteção de Agentes Móveis contra Plataformas Maliciosas

A literatura aponta um conjunto de ameaças a que os agentes móveis estão sujeitos, entre elas estão [Jansen e Karygiannis, 1999]:

- **Personificação**. Uma plataforma de agentes pode se passar por outra plataforma no esforço de enganar um agente móvel sobre o seu verdadeiro destino e ainda atrair os agentes para a plataforma com o intuito de extrair informações sensíveis.
- **Negação de Serviço**. Uma plataforma maliciosa pode ignorar requisições de serviço de um agente e introduzir atrasos inaceitáveis a execuções críticas; pode também simplesmente não executar o código do agente.
- **Intromissão**. Uma plataforma de agentes pode monitorar tanto a comunicação, como a execução de agentes. Todos os dados não cifrados e os dados subseqüentemente gerados são apresentados à plataforma.
- **Modificação Não-Autorizada**. Quando um agente chega a uma plataforma, este está revelando seu código e estado à plataforma. Mecanismos devem ser suportados para garantir a integridade do código e do estado do agente.

A segurança de agentes móveis envolve principalmente a **integridade** do agente, para evitar que plataformas alterem o código ou dados que sejam coletados durante as visitas, e a **confidencialidade** do código e do estado do agente, para evitar a violação da propriedade intelectual. Em algumas abordagens que visam proteger um agente móvel, o criador do agente pode restringir o itinerário do mesmo apenas para um conjunto confiável de plataformas previamente conhecidas. Um exemplo é a solução organizacional proposta em [Tardo e Valente, 1996], em que o sistema de agentes não é aberto e apenas partes confiáveis podem operar nos sítios. Apesar destes esquemas simples baseados em confiança terem o seu valor, estes não suportam itinerários livres sem prévia definição. Uma vez que o problema é o comportamento errado do ambiente de execução, em face de um comportamento que atenda a especificação, outra classe de abordagens usa *hardwares* especiais a prova de ataques, visando construir ambientes confiáveis.

Uma classificação dos mecanismos de segurança para proteção dos agentes pode ser feita partindo do seu propósito principal: prevenção ou detecção.

- Os **Mecanismos de Prevenção** tentam tornar mínima a possibilidade de acesso e/ou modificação dos agentes e se utilizam das seguintes abordagens: *Hardware* seguro, Computação com Funções Cifradas [Sander e Tschudin, 1998] e Ofuscamento de Código [Hohl, 1998]
- Os **Mecanismos de Detecção** tentam descobrir se e quando um ataque foi realizado, após a execução do agente, e se utilizam das seguintes abordagens: Encapsulamento de Resultado Parcial [Karjoth et al., 1998, Karnik, 1998], Contêiner de Dados Somente Leitura e Vetor de Dados Direcionados [Karnik, 1998], Registro de Itinerário com Replicação e Votação [Schneider, 1997] e Rastros Criptográficos [Vigna, 1998a].

As abordagens de prevenção baseadas em *Hardwares* Seguros garantem a integridade do agente, porém estas não são práticas devido aos elevados custos associados a necessidade desses *hardwares*. Já a abordagem de Computação com Funções Cifradas, que visa garantir a confidencialidade do agente durante a sua computação, até o momento não apresenta soluções consideradas práticas. Devido à dificuldade em medir a eficiência e o tempo de proteção da abordagem de ofuscamento de código, a sua aplicabilidade se mostra limitada a aplicações que exigem uma segurança branda.

As técnicas de detecção são as que se mostram mais adequadas de serem implantadas. Porém, algumas limitações encontradas nas técnicas de Registro de Itinerário com Replicação e Votação e Rastro de Execução precisam ainda ser cuidadosamente examinadas para que estas técnicas sejam consideradas plenamente aplicáveis. Os problemas e limitações são ainda maiores quando se leva em conta aplicações em sistemas abertos já que estas aplicações envolvem vários domínios e tecnologias heterogêneas. As abordagens de Encapsulamento de Resultados Parciais e dos Contêineres de Dados Somente Leitura e de Dados Direcionados são as que apresentam melhores resultados.

A abordagem de Encapsulamento de Resultados Parciais visa recolher os resultados das ações dos agentes, praticadas nas plataformas visitadas, para verificações subseqüentes. O encapsulamento pode ter diferentes propósitos: fornecer confidencialidade, usando cifragem; fornecer integridade e autenticidade, usando assinatura digital. Em geral, há três formas de encapsular resultados parciais: (1) habilitar o agente com meios para encapsular informações; (2) ou depender das habilidades de encapsulamento da plataforma; (3) ou depender de uma terceira parte confiável (chamada de TTP). A primeira forma apresenta diversas limitações quanto a manipulação de chaves e de funções de geração de chaves por parte do agente [Karjoth et al., 1998]. A terceira forma, além de acarretar perdas de desempenho devido à necessidade do agente migrar para uma TTP sempre que este visitar um sítio não confiável, possui outras limitações e vulnerabilidades, conforme descritos em [Roth, 2001]. A solução que se evidencia como a mais adequada é a que depende da plataforma para executar o encapsulamento. Os seguintes trabalhos seguem esta abordagem: Protocolos KAG [Karjoth et al., 1998], Contêiner Somente para Inclusão [Karnik, 1998] e o Protocolo Múltiplos-saltos [Corradi et al., 1999]. Na seção 5, estes trabalhos serão comparados com a proposta deste artigo descrita, a seguir.

3. Uso de Repositórios Seguros para Proteção de Agentes Móveis

Os mecanismos aqui propostos foram projetados para um modelo de agentes que permite múltiplos-saltos e itinerários livres. Visando possibilitar a sua implantação em aplicações distribuídas em sistemas abertos, como a Internet, as plataformas de agentes estão em conformidade com a especificação MAF (*Mobile Agent Facility*) [OMG, 2000].

Visando detectar possíveis alterações, remoções ou inserções de dados em um agente móvel (violação da integridade), duas técnicas serão empregadas neste trabalho: uma para proteção do código e dos dados imutáveis do agente — **Assinatura do Código do Agente e do Repositório de Dados Somente-Leitura** — e outra para o resultados parciais recolhidos durante as viagens do agente — **Repositório Seguro de Resultados Parciais**. Outra técnica que visa não só a integridade de dados, mas também a confidencialidade de alguns dados para que não sejam revelados a plataformas não autorizadas, também compõe o esquema proposto — **Repositório de Dados Direcionados**. O esquema proposto irá auxiliar o programador do agente na construção de um agente móvel protegido e ainda auxiliar as plataformas visitadas na verificação da integridade do agente (código e repositórios)— **Autenticador *Multi-Hop***.

No esquema proposto, uma plataforma, ao criar um agente móvel em seu contexto, antes de dispará-lo para a primeira plataforma a ser visitada, deve definir a **qualidade de proteção** requerida para o agente, chamada de QoP. QoP é um atributo somente-leitura, que expressa os mecanismos de segurança que devem ser usados pelas plataformas visitadas pelo agente para verificar os repositórios de dados que este carrega consigo.

A Tabela 1 resume as notações criptográficas usadas para manipulação dos repositórios de dados que compõem o esquema proposto.

3.1. Estrutura Proposta para um Agente Móvel

Para implantação dos repositórios de dados propostos, partes do estado do agente precisam ser distintos, conforme ilustrado na Figura 1. A maioria das plataformas de agentes móveis usa apenas a assinatura

P_o	Identidade da plataforma de origem
$P_i, 1 \leq i \leq n$	Plataformas visitadas por um agente móvel
r_i	Número aleatório gerado pela plataforma P_i
$ENC_o(m)$	Mensagem m cifrada com a chave pública de P_o
$SIG_i(m)$	Assinatura da plataforma P_i sobre a mensagem m
$H(m)$	Uma função *hash one-way* livre de colisões (p.ex, SHA-1)
$creds$	Credenciais de um agente móvel

Tabela 1: Notação Criptográfica

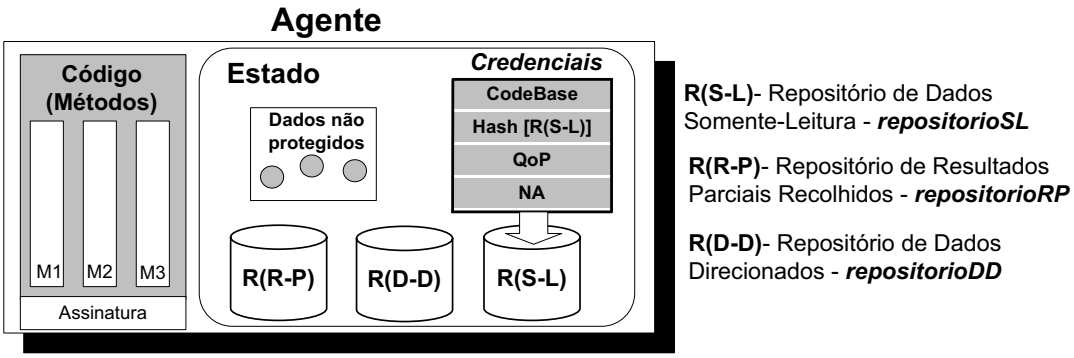

Figura 1: Estrutura Proposta para um Agente Móvel

do código do agente pelo proprietário para expressar a posse de um agente móvel (em nome de quem este agente atua). Porém, segundo Roth [Roth, 2001], isto não é suficiente para distinguir uma instância de um agente de outra. Segundo o autor, o dono do agente deve assinar um núcleo estático do agente que pode incluir o código do agente e uma redundância suficiente para distinguir entre duas instâncias de um mesmo agente. No esquema proposto, após a criação do programa do agente, o proprietário deve gerar o núcleo estático do agente, chamado de credencias do agente (**objeto *Credenciais***) para distinguir instâncias de um mesmo agente. Cada agente deve carregar suas credenciais assinadas como parte do seu estado dentro de um repositório de dados somente-leitura, chamado de *repositorioSL*. O objeto *Credenciais* distingue as instâncias do agente e a sua assinatura liga o agente ao seu proprietário, auxiliando assim o processo de verificação da integridade do agente, que será descrito na seção 3.5, e o processo de controle de acesso[1]. A seguir, são definidos os quatro campos que compõem o objeto *Credenciais* (ver Figura 1).

***Codebase* do agente.** Especifica a localização do servidor das classes que um agente pode requerer. Vale ressaltar a importância de se proteger o *codebase* para que este não aponte para uma entidade não confiável.

***Hash* dos dados do repositorioSL.** Com o objetivo de ligar o objeto *Credenciais* com os dados originais do agente, armazenados em um repositório de dados somente-leitura (*repositorioSL*) que refletem a tarefa do agente, propõem-se que o valor do *hash* criptográfico dos dados somente-leitura [2] seja também armazenado no objeto Credenciais. Esta ligação é útil para detectar ataques onde uma plataforma maliciosa reutiliza as credenciais de um agente visitante em seu próprio agente.

Qualidade de Proteção (QoP). Atributo que identifica a qualidade de proteção requerida que deve ser atendida por todas as plataformas visitadas pelo agente.

Número Aleatório. Por fim, um número aleatório (*NA*) grande o suficiente, para não ser reproduzido duas vezes pelo proprietário do agente, deve ser incluído no objeto.

[1] O esquema de autorização usado para proteger as plataformas de agentes contra agentes maliciosos está fora do escopo deste trabalho e está descrito em detalhes em [Wangham e Fraga, 2003].

[2] Somente os dados estáticos referentes a aplicação, sem o objeto Credenciais.

Na estrutura proposta para um agente móvel ilustrada na Figura 1, três objetos opcionais, chamados de repositórios de dados (R(S-L), R(R-P)e R(D-D)), podem fazer parte do estado do agente. Os repositórios serão utilizadas para armazenar de forma segura os dados coletados/transportados pelo agente, conforme descrito nas seções a seguir.

3.2. Assinatura do Código do Agente e de Dados Somente Leitura

Na maioria das plataformas de agentes móveis baseados na linguagem Java, quando um agente migra, não se transfere o código junto com o estado. Com isso, o código é apenas solicitado sob demanda a partir do *codebase* do agente, se o mesmo não pode ser suprido pela máquina virtual local. Entretanto, isto depende de como a plataforma implementa a transferência de agentes. No esquema proposto, um proprietário deve assinar o código do seu agente para proteger a sua integridade. Sempre que este código for solicitado sob demanda ou for serializado para ser transportado pela rede, esta assinatura deverá acompanhá-lo. Desta forma, modificações feitas no código do agente podem ser facilmente detectadas por qualquer plataforma visitada pelo agente.

Além do código, no esquema proposto, o proprietário do agente pode ainda assinar todos os dados originais indicados pelo programador do agente como sendo somente-leitura e armazenados no ***repositorioSL***. A técnica adotada neste trabalho está baseada no Contêiner Somente-Leitura, proposto em [Karnik, 1998]. Além dos dados específicos de cada aplicação, o repositório somente-leitura conterá o objeto **Credenciais**. Qualquer plataforma visitada pelo agente pode verificar a integridade deste repositório. Para isto, basta a plataforma visitada obter a chave pública da plataforma de origem, decifrar a assinatura e recalcular o *hash* dos dados armazenados no *repositorioSL* e comparar os resultados.

3.3. Repositório Seguro de Resultados Parciais

Neste trabalho, os dados sensíveis, gerados em uma plataforma, devem ser armazenados em um repositório de resultados parciais, chamado de ***repositorioRP***, que será carregado pelo agente, para que possíveis modificações de plataformas possam ser detectadas.

No esquema proposto, os protocolos criptográficos usados pelas plataformas para inserir e proteger um resultado no *repositorioRP* são semelhantes aos protocolos P1 e P2 da Família KAG, definidos por Karjoth et. al [Karjoth et al., 1998]. Algumas adaptações e modificações foram realizadas nos protocolos P1 e P2 da Família KAG e estas serão discutidas a seguir. Seguindo a notação apresentada na Tabela 2, as Figuras 2, 3 e 4 descrevem os procedimentos necessários para inicialização do *repositorioRP*, para inserção de um resultado e para verificação da integridade do *repositorioRP*, respectivamente.

$rp_i, 1 \leq i \leq n$	Resultado parcial obtido na plataforma P_i
$RP_i, 1 \leq i \leq n$	Resultado parcial protegido obtido na plataforma P_i
$h_i, 1 \leq i \leq n$	Valor associado com RP_i para verificação da integridade
$H(creds)$	Hash do objeto Credenciais (identificador único do agente)

Tabela 2: Notação do Repositório Seguro de Resultados Parciais

3.3.1. Iniciando o Repositório de Resultados Parciais

Para iniciar o *repositorioRP*, o proprietário do agente (plataforma de origem P_o) gera um número aleatório r_o e computa o valor de h_o, aplicando um função *hash* sobre r_o e a identidade da primeira plataforma (P_1) a ser visitada (passo 1 da Figura 2). Este h_o é o valor âncora da relação de encadeamento dos resultados parciais recolhidos. O proprietário deve ainda cifrar o valor r_o com a sua própria chave pública. Em seguida, o resultado deste ciframento mais o identificador único do agente e a relação de encadeamento h_o são assinados pelo proprietário do agente, criando o resultado parcial protegido RP_o (passo 2, Figura 2). No esquema proposto, o identificador único do agente é o valor resultante da aplicação da função *hash* sobre o objeto *Credenciais*, já nos protocolos da família KAG,

este identificador é um *token* emitido pela plataforma de origem. Finalmente, no passo 3, o resultado parcial é inserido no *repositorioRP* (RP_o) e enviado para a plataforma P_1.

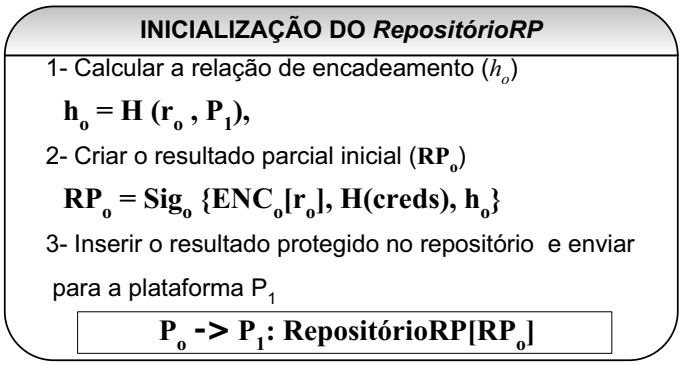

Figura 2: Inicialização do Repositório de Resultados Parciais

Ao contrário do que ocorre nos protocolos da Família KAG, em todos os protocolos de inserção de resultados descritos, a seguir, o processo de inicialização do *repositorioRP* é o mesmo (Figura 2). No protocolo P2 da Família KAG, o resultado parcial inicial é calculado de forma diferente: $RP_o = ENC_o(SIG_o(rp_o), r_o), h_o$. Conforme constatado em [Roth, 2001], o procedimento de inicialização do protocolo P2 da família KAG apresenta um problema não tratado. Uma plataforma não tem como identificar qual plataforma de origem do agente uma vez que a assinatura do resultado inicial está escondida pelo ciframento. Este problema é contornado neste trabalho pelo protocolo de iniciação apresentado na Figura 2.

3.3.2. Inserção de Resultados Parciais

A Figura 3 sintetiza os três protocolos para inserção de resultados parciais no **repositorioRP** propostos neste trabalho. O proprietário do agente deve definir qual protocolo será usado pelas plataformas de agentes para inserir os resultados de acordo com a necessidade de cada aplicação e informar o protocolo usado às plataformas a serem visitadas. Esta informação de qual protocolo deverá ser usado estará contida nas credencias do agente, no atributo que identifica a qualidade de proteção (QoP).

Protocolo A

O primeiro protocolo proposto para inclusão de resultados parciais está baseado no protocolo P1 da família KAG [Karjoth et al., 1998]– *Protocolo de Assinatura Digital Encadeada Verificável Publicamente*. O primeiro passo do protocolo, descrito na Figura 3.a e do protocolo P1, é o mesmo – calcular o valor *hash* sobre o estado parcial protegido anterior mais a identidade da próxima plataforma a ser visitada. O uso de h_i, no passo 2, tem por objetivo ligar o resultado obtido anteriormente com o resultado parcial corrente. Logo, RP_{i-1} não pode ser modificado sem afetar RP_i. A plataforma P_i, em uma segunda visita do agente, também não pode modificar seu próprio resultado sem invalidar a cadeia em RP_i. A inclusão da identidade da próxima plataforma no cálculo de h_i (passo 1) garante ainda que somente P_{i+1} pode adicionar o próximo resultado parcial.

Visando manter a confidencialidade de alguns dados sensíveis gerados nas plataformas e que serão carregados pelos agentes, cada plataforma deve cifrar, com a chave pública do proprietário do agente, o resultado parcial juntamente com um número aleatório gerado pela própria plataforma, para que somente a plataforma de origem tenha acesso aos dados. Para construir o resultado parcial protegido (RP_i), a plataforma deve assinar o resultado do ciframento citado, concatenado com a relação de encadeamento (h_i) mais o identificador do agente (passo 2 da Figura 3.a).

O passo 2 (Figura 3.a) é que difere o protocolo proposto do protocolo P1 da família KAG. A modificação está na inclusão do identificador do agente ($H(creds)$) que passa a ser anexado ao

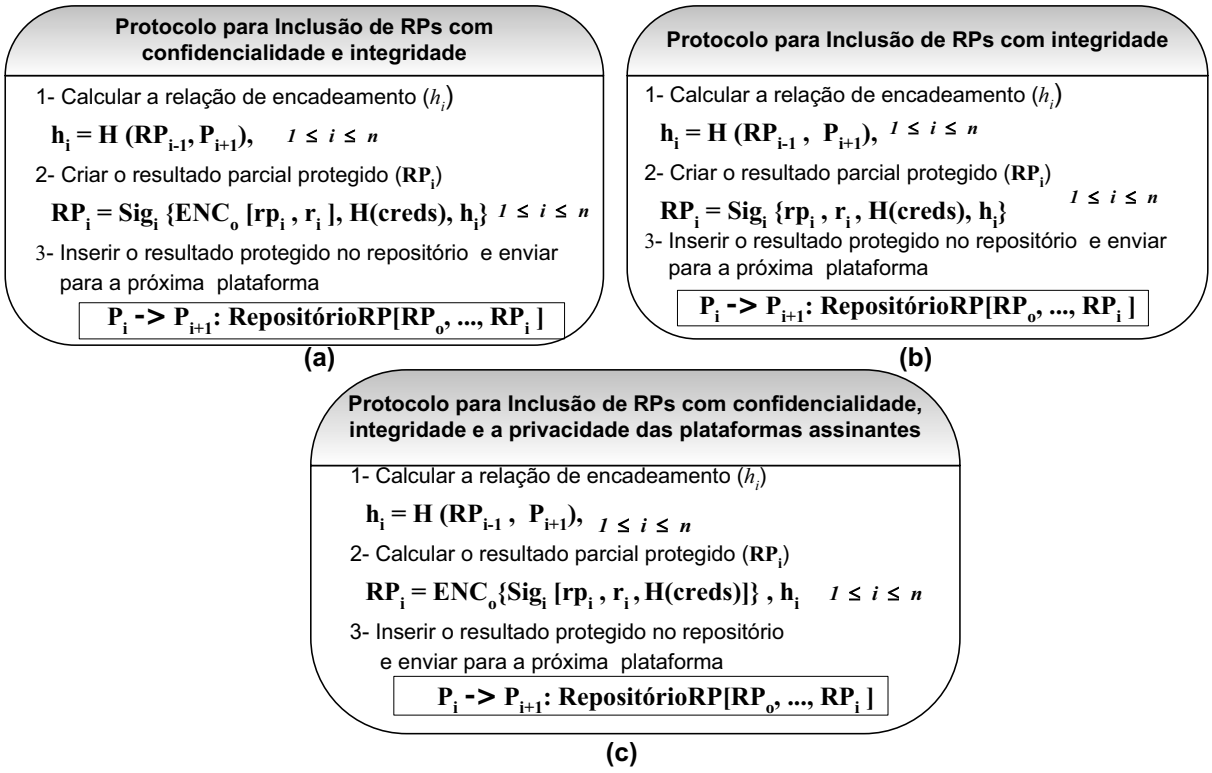

Figura 3: Protocolos para Inclusão de Resultados Parciais

resultado parcial encapsulado em cada plataforma. Esta modificação tem por objetivo relacionar cada resultado parcial coletado com a instância do agente que está em execução, contornando assim o ataque descrito em [Roth, 2001] cujos protocolos P1 e P2 da família KAG são suscetíveis.

Protocolo B

Vale ressaltar que os dados cifrados podem impor um custo ao desempenho das aplicações e que esta degradação deve ser considerada quando o conjunto de mecanismos a ser empregado na proteção do agente for selecionado. Pensando neste problema e tomando ainda como base o protocolo P1 da família KAG, propõem-se um segundo protocolo para inserção de resultados parciais (Figura 3.b), que não exige o ciframento dos dados coletados. Este protocolo é semelhante ao anterior só difere no passo 2, em que os dados não são cifrados.

Protocolo C

O terceiro protocolo proposto está baseado no protocolo P2 da família KAG– *Protocolo de Assinatura Digital Encadeada com Privacidade do Assinante*. O protocolo P2 da família KAG é uma variação do protocolo P1, em que a ordem da cifragem e da assinatura digital são trocadas visando esconder a identidade das plataformas que inseriram resultados, mantendo ainda o objetivo pela integridade. Entretanto, a integridade dos resultados inseridos não poderá mais ser verificada publicamente. No Protocolo C, ilustrado na Figura 3.c, o passo 1 é idêntico ao protocolo P2, já o passo 2 inclui a redundância necessária para associar cada resultado encapsulado com a instância do agente. Porém, conforme discutido anteriormente, o processo de inicialização do Protocolo C difere do processo do protocolo P2 da família KAG.

A Tabela 3 compara os objetivos de segurança de cada protocolo proposto, tendo como base os objetivos apresentados em [Karjoth et al., 1998]. Conforme indicado na tabela, a remoção total de resultados parciais coletados não é detectada em nenhum dos protocolos. Segundo [Karjoth et al., 1998, Roth, 2001], a remoção total de resultados é o problema mais difícil a ser tratado na preservação do

Objetivos de Segurança	Protocol A	Protocolo B	Protocolo C
Confidencialidade dos resultados inseridos	sim	não	sim
Integridade em avanço (forte)	sim	sim	sim
Não-repudiação da inserção de resultados	sim	sim	sim
Integridade verificável publicamente	sim	sim	não
Privacidade do assinante do resultado encapsulado	não	não	sim
Resistência a inserção de resultados falsos	sim	sim	sim
Resistência a remoção total dos dados	não	não	não

Tabela 3: Comparação dos Objetivos de Segurança dos Protocolos Propostos

estado da cadeia de resultados parciais encapsulados. Quando duas plataformas maliciosas conspiram[3], estas podem controlar todos os dados inseridos nos sítios visitados entre as duas plataformas maliciosas.

3.3.3. Verificando a Integridade do *repositorioRP*

Conforme pode ser observado na Tabela 3, nos protocolos A e B, a integridade em avanço dos resultados parciais encapsulados pode ser verificada por qualquer plataforma visitada pelo agente. A Figura 4 ilustra um exemplo, no qual a plataforma P_4 verifica a integridade do *repositorioRP* carregado por um agente que percorreu as plataformas P_1, P_2 e P_3, respectivamente. No exemplo, a plataforma P_3, após inserir o seu resultado parcial, substitui o resultado RP_1, inserido pela plataforma RP_1, por RP_1'. O procedimento necessário para verificar a integridade dos resultados parciais protegidos (RPs) está também descrito na Figura 4. No passo 1, o primeiro resultado parcial, cuja integridade será verificada, é obtido do *repositorioRP* (RP_3). No passo 2, usando a chave pública da plataforma P_2, (X, h_2) é obtido, onde X é uma string não interpretável (resultante do processo de cifragem — ver Figura 3.a) e h_2 é a relação de encadeamento em P_2. Como h_2 foi obtido usando RP_1, quando $H(RP_1', P4)$ for calculado (passo 3), será comprovado que o repositório foi violado em RP_1 e, devido à relação de encadeamento entre os resultados, todos os resultados inseridos após RP_1 também estão corrompidos.

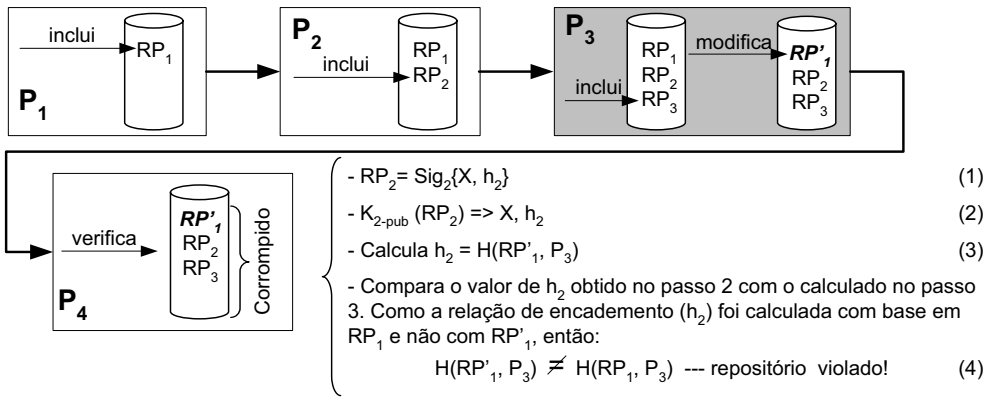

Figura 4: Exemplo de Verificação da Integridade do *repositorioRP*

3.4. Repositório Seguro de Dados Direcionados

Este trabalho propõe ainda um repositório de dados direcionados, chamado de ***repositorioDD***, baseado na técnica implementada na plataforma Ajanta que possibilita uma revelação seletiva do estado de um agente [Karnik, 1998], na qual o programador do agente pode implementar um vetor de dados direcionados em que cada entrada do repositório tenha plataformas específicas como destinatárias. Esta técnica exige que as plataformas definidas como receptoras de informações estejam pré-determinadas.

[3]Duas plataformas conspiram quando estas trocam segredos usados na construção da relação de encadeamento.

Conforme descrito em [Roth, 2001], esta técnica, proposta por Karnik, possui algumas vulnerabilidades que permitem que dados sejam revelados a plataformas maliciosas. Visando contornar estas vulnerabilidades, propõem-se neste trabalho um aprimoramento desta técnica com a inclusão de uma redundância criptográfica que liga cada entrada do *repositorioDD* com a instância do agente móvel correspondente.

A Figura 5 ilustra um exemplo, em que o programador do agente cria o *repositorioDD* com duas entradas de dados direcionados (DD_1 e DD_3), para as plataformas P_1 e P_3, respectivamente. Para garantir a confidencialidade dos dados, o programador, com a chave pública da plataforma receptora, deve cifrar o dado concatenado com o identificador único da instância do agente. A plataforma de origem deve ainda assinar este valor para garantir a sua integridade (ver Figura 5). Quando o agente é recebido em uma plataforma receptora (plataforma P_1), antes de continuar a execução do agente, a plataforma deve verificar a origem deste dado, sua integridade e se este está realmente associado ao agente recebido. Para isto, os passos ilustrados na Figura 5 devem ser seguidos.

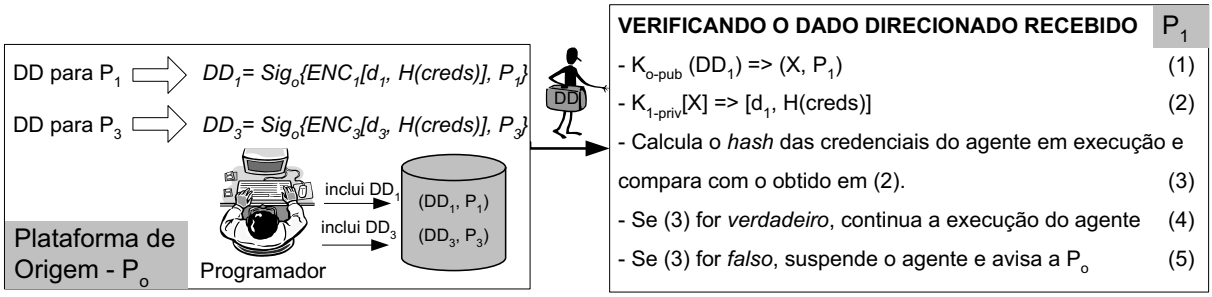

Figura 5: Incluindo e Verificando Dados Direcionados no *repositorioDD*

3.5. Autenticador *Multi-Hop*

Visando prover segurança para as plataformas de agentes móveis, em [Wangham e Fraga, 2003], foi proposto um autenticador de agentes *multi-hop* que visa estabelecer a confiança em um agente móvel. Este mecanismo auxilia não só a proteção da plataforma mas também a proteção dos agentes, pois detecta as possíveis violações da integridade do agente. A Figura 6 ilustra os passos que auxiliam a verificação dos repositórios de dados quando um agente chega em uma plataforma[4]. Nos **passos 1 e 2**, da Figura 6, uma plataforma deve, através da verificação da assinatura do código e do *repositorioSL*, confirmar que este agente não foi corrompido e confirmar a sua associação a um principal, seu proprietário. Desta forma, modificações feitas por plataformas maliciosas no código e/ou nos dados somente-leitura (*repositórioSL*) podem ser facilmente detectadas por qualquer plataforma visitada pelo agente. Por fim, no passo 3, é necessário verificar a integridade dos dados do *repositorioDD* e do *repositorioRP* que estão sendo carregados pelo agente.

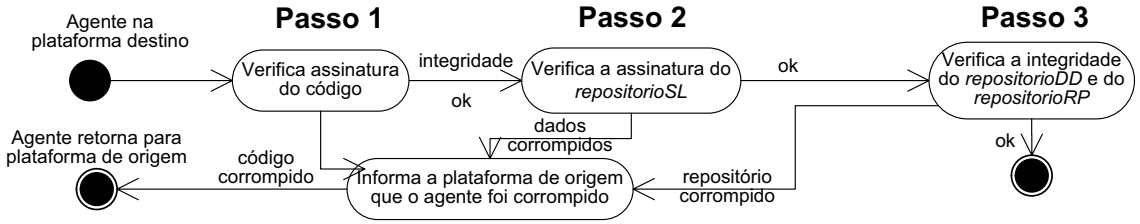

Figura 6: Autenticador Multi-hop

[4]Os passos que auxiliam a proteção das plataformas de agentes, por estarem fora do escopo deste trabalho, não estão ilustrados na Figura 6.

4. Implementação e Resultados

Um protótipo, envolvendo os mecanismos propostos, foi definido e implementado visando comprovar a viabilidade de sua utilização em aplicações distribuídas que embutem a noção de mobilidade de código e, por conseguinte, validar o esquema proposto.

No protótipo, para compor a camada do ambiente computacional que suporta os agentes móveis, foi adotada a plataforma *Aglets*[5] da IBM, primeiramente por usar o Java e também por ser uma iniciativa de código aberto. O *Aglets* oferece mecanismos para mobilidade de código, de dados e de informação de estado em um ambiente computacional.

Para auxiliar o processo de criação de agentes e o uso dos repositórios de dados seguros, uma interface gráfica foi implementada. Esta interface permite a um proprietário definir quais repositórios de dados serão usados e anexados ao agente. A Figura 7 ilustra a visão das classes implementadas (pacote `br::ufsc::das::agletsec`), para dar o suporte necessário para integrar os repositórios propostos à plataforma de agentes *Aglets*. O *repositorioSL*, o *repositorioDD* e o *repositorioRP*, implementados nas classes `RORepository`, `DDRepository` e `PRRepository`, respectivamente, quando selecionados, devem ser criados e definidos no método `oncreation` por qualquer agente que estenda a classe abstrata `com.ibm.aglet.Aglet`. Vale ressaltar que, durante o processo de criação de um objeto `RORepository`, o objeto `Credentials` associado ao agente é criado e armazenado neste repositório.

O algoritmo do autenticador *multihop*, proposto na Figura 6, foi implementado em um agente estacionário chamado `SecurityInterceptor`. Este agente deve ser iniciado em todas as plataformas que irão receber o agente móvel e sua missão é interceptar o processo de recebimento deste agente, para que a integridade do agente móvel possa ser verificada antes que este se inicie no contexto da plataforma. Além disso, este agente verifica ainda a assinatura do código do agente.

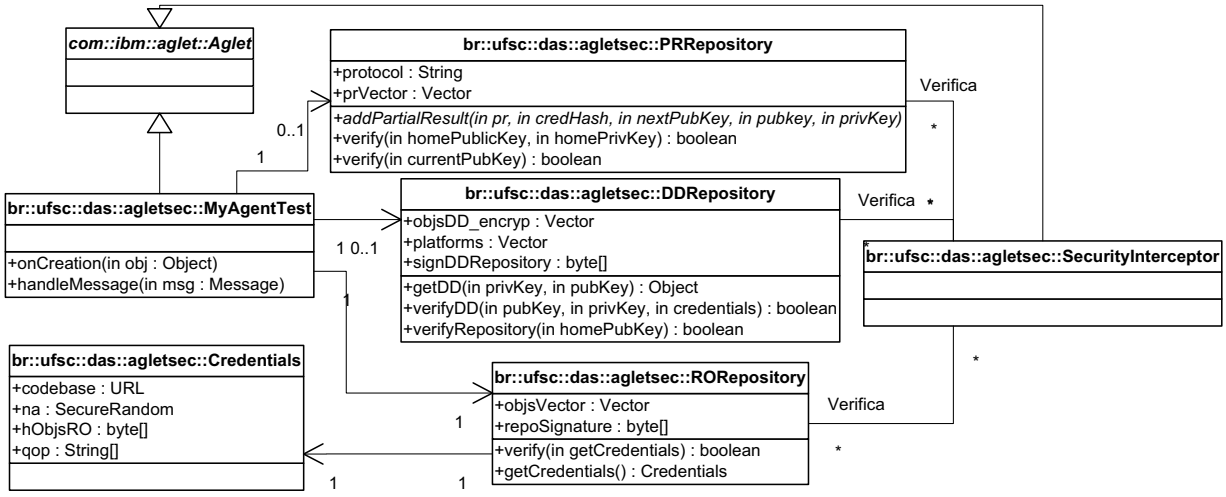

Figura 7: Diagrama de Classes

5. Trabalhos Relacionados

Analisando as plataformas de agentes móveis acadêmicas e comerciais encontradas na literatura, observa-se que a segurança dos agentes móveis contra plataformas maliciosas ainda é um problema não resolvido nestes sistemas. Dentre sete plataformas analisadas[6], somente duas plataformas, SOMA [Corradi et al., 1999] e Ajanta [Karnik, 1998], preocupam-se em prover mecanismos. Ambas procuram

[5] http://aglets.sourceforge.net

[6] Foram analisadas as plataformas comerciais *Aglets*, *Concordia*, *Grasshopper*, *Voyager* e, as plataformas acadêmicas, *Mole*, *SOMA* e *Ajanta*.

detectar os ataques provenientes de plataformas maliciosas. Para a proteção do estado do agente, a plataforma *SOMA* utiliza técnicas de detecção, através dos protocolos TTP (*Trusted Third Parties*) e MH (*Multiple-Hop*). O primeiro protocolo necessita de uma terceira parte confiável para encapsular os resultados e, conforme citado anteriormente, esta dependência não é desejável. No protocolo MH, assim como no protocolo de encapsulamento de resultados proposto neste artigo, é responsabilidade da plataforma o encapsulamento dos resultados parciais no estado do agente de forma que seja impossível modificar um resultado intermediário sem modificar também todos os resultados parciais posteriores. Para isto, uma relação de encadeamento entre os resultados necessita ser criptograficamente estabelecida.

A plataforma *Ajanta*, através do mecanismo Contêiner Somente para Inclusão, também usa a mesma abordagem de encapsulamento de resultados baseada em relações de encadeamento para garantir a integridade dos resultados parciais. Outro trabalho que segue essa abordagem, são os protocolos da Família KAG. A grande diferença entre esses mecanismos está na forma como criptograficamente estas relações de encadeamento são construídas e como estes resultados são anexados ao agente. A Tabela 4 visa comparar algumas características destes mecanismos com os protocolos propostos neste trabalho.

Características e Propriedades de Segurança	MH SOMA	Contêiner Ajanta	P1 KAG	P2 KAG	Protocolos Propostos		
					PA	PB	PC
Verificável publicamente	não	não	sim	não	sim	sim	não
Integridade em avanço[a]	não	não	sim	sim	sim	sim	sim
Confidencialidade	não	sim	sim	sim	sim	não	sim
Detecção inserção de resultados (redundância criptográfica)	não	não	não	não	sim	sim	sim
Remoção total de resultados	não	não	não	não	não	não	não

[a] Quando um resultado parcial é modificado, somente este resultado e os posteriormente inseridos são comprometidos

Tabela 4: Comparação dos Mecanismos de Encapsulamento de Resultados

6. Conclusão

A aceitação do paradigma de agentes móveis ainda é prejudicada devido a questões de segurança. As limitações das técnicas apontadas nas seções 2 e 5 comprovam que estas ainda não apresentam resultados satisfatórios que garantam a segurança para os agentes móveis. Tais limitações são ainda maiores quando se leva em conta aplicações em sistemas abertos. Foi a constatação dessas limitações e a preocupação com estes aspectos específicos da segurança que motivaram este trabalho que visa detectar ataques contra os agentes móveis, baseado no uso de repositórios seguros de dados e assinatura digital.

Entre os trabalhos relacionados, constatou-se que dois protocolos que visam o encapsulamento seguro de resultados parciais apresentam bons resultados (P1 e P2 da Familia KAG). Com base nestes protocolos, foram propostos três outros protocolos que contornam algumas limitações e vulnerabilidades encontradas nestes primeiros. Outro mecanismo, que visa proteger dados direcionados para plataformas pré-definidas [Karnik, 1998], também serviu como base para a proposta do repositorioDD e a sua vulnerabalidade, apontada em [Roth, 2001], também foi contornada neste trabalho. De forma a automatizar o processo de verificação de possíveis violações do código e dos repositórios de dados carreagados pelo agente, um autenticador *multi-hop* foi proposto. Todo o esquema proposto foi implementado e integrado à plataforma Aglets. Atualmente, testes de *software* estão sendo realizados e em seguida serão iniciados os testes de desempenho. Após concluída esta fase de testes, uma metodologia será elaborada visando auxiliar a escolha dos repositórios mais adequados às funcionalidades de uma dada aplicação.

Referências

Chess, D. M. (1998). Security issues in mobile code systems. Em Vigna, G., editor, *Mobile Agents and Security*, volume LNCS 1419. Springer.

Corradi, A., Cremonini, M., Montanari, R., e Stefanelli, C. (1999). Mobile agents integrity for electronic commerce applications. *Information Systems*, 24(6):519–533.

Hohl, F. (1998). Time limited blackbox security: Protecting mobile agents from malicious hosts. Em Vigna, G., editor, *Mobile Agents and Security*, volume LNCS 1419, páginas 92–113. Springer.

Jansen, W. e Karygiannis, T. (1999). Mobile agent security. Relatório Técnico NIST Special Publication 800-19, National Institute os Standards and Technology.

Karjoth, G., asokan, N., e C.Gülcü (1998). Protecting the computing results of free-roaming agents. Em *Proc. of the Second International Workshop on Mobile Agents*.

Karnik, N. (1998). *Security in Mobile Agent System*. Tese de Doutorado, University of Minnesota.

OMG (2000). Mobile agent facility specification. OMG Document 2000-01-02.

Roth, V. (2001). On the robustness of some cryptographic protocols for mobile agent protection. Em Picco, G. P., editor, *Mobile Agents*, volume LNCS 2240, páginas 1–14. Springer.

Sander, T. e Tschudin, C. (1998). Protecting mobile agens against malicious hosts. Em Vigna, G., editor, *Mobile Agents and Security*, volume LNCS 1419. Springer.

Schneider, F. B. (1997). Towards fault-tolerant and secure agentry. Em Mavronicolas, M. e Tsigas, P., editors, *11th International Workshop on Distributed Algorithms (WDAG'97)*, volume Lecture Notes in Computer Science, páginas 1–14. Springer.

Tardo, J. e Valente, L. (1996). Mobile agent security and telescript. Em *IEEE COMPCON'96*, páginas 58–63. IEEE Computer Society Press.

Vigna, G. (1998a). Cryptographic traces for mobile agents. Em Vigna, G., editor, *Mobile Agents and Security*, volume LNCS 1419, páginas 137–153. Springer.

Vigna, G., editor (1998b). *Mobile Agents and Security*, volume LNCS 1419. Springer.

Wangham, M. e Fraga, J. (2003). Mecanismos de segurança para plataformas de agentes móveis, baseados em redes de confiança spki/sdsi. Em *Simpósio Brasileiro de Redes de Computadores*.