Comunicação de Grupo Segura no CORBA: Integridade e Confidencialidade no MIOP

Alysson Neves Bessani[1]*, Lau Cheuk Lung[3], Joni da Silva Fraga[1], Carla Merkle Westphall[2]

[1]DAS-UFSC - Departamento de Automação e Sistemas

[2]INF-UFSC - Departamento de Informática
UFSC - Universidade Federal de Santa Catarina
Caixa Postal 476 - CEP 88040-900 - Trindade - Florianópolis - SC

[3]PPGIA - Programa de Pós-Graduação em Informática Aplicada
PUC-PR - Pontifícia Universidade Católica do Paraná
R. Imaculada Conceição, 1155 - Prado velho - CEP 80215-901 - Curitiba -PR

{neves,fraga}@das.ufsc.br, lau@ppgia.pucpr.br, carla@inf.ufsc.br

Resumo. *A inclusão de propriedades de segurança em suportes de comunicação de grupo permite que novas aplicações, que apresentam fortes requisitos de segurança, também possam se utilizar de suportes deste tipo. Este trabalho apresenta uma arquitetura de comunicação de grupo segura construída a partir do padrão para difusão não confiável definido pela OMG para o CORBA. Esta arquitetura esta centrada na figura do gerenciador de chave de grupo, entidade responsável pela atualização e distribuição da chave compartilhada pelos membros de um grupo.*

1. Introdução

Comunicações ponto a ponto têm se mostrado bastante úteis em aplicações distribuídas desenvolvidas em CORBA (*Common Object Request Broker)* [OMG, 2001a]. Para ampliar o espectro de aplicações que podem se beneficiar desse *middleware*, a OMG (*Object Management Group*) tem trabalhado no sentido de introduzir novos mecanismos e protocolos de comunicação no CORBA, como por exemplo as especificações CORBASec [OMG, 2001b] (comunicação ponto a ponto segura, baseado no SSL (*Secure Socket Layer*) [Freier et al., 1996]) e UMIOP (comunicação multi-ponto não confiável, baseado no IP *multicast*) [OMG, 2001c].

O CORBASec, o modelo CORBA de segurança, especifica objetos de serviço e componentes de tecnologia de segurança (a nível de serviços de segurança subjacentes) e componentes de proteção básica, fornecidos por uma combinação de hardware e sistemas operacionais locais. Com esse modelo é possível estabelecer uma conexão segura cliente/servidor que garanta a integridade e a confidencialidade das mensagens trocadas, além de autenticação.

O UMIOP (*Unreliable Multicast Inter-ORB Protocol*) [OMG, 2001c] define um protocolo de difusão não confiável para ser incluído no ORB. Esse protocolo, chamado de

*Bolsista CNPq.

MIOP (*Multicast Inter-ORB Protocol*), é responsável por mapear mensagens GIOP sobre a pilha UDP/*multicast* IP. O *multicast* IP compreende um conjunto de extensões ao protocolo IP que o habilita na concretização de comunicações multiponto [Deering, 1986]. Este protocolo é caracterizado pela ausência de garantias e pelo alto desempenho, especialmente em LAN. Várias são as aplicações que utilizam *multicast* IP, principalmente em sistemas de difusão multimídia na Internet.

O MIOP não aporta características mais fortes de comunicação como confiabilidade e segurança. A primeira é fundamental em sistemas que não toleram perdas ou inconsistências, como por exemplo os utilizados em aplicações tolerantes a faltas, já as características de segurança são úteis, por exemplo, em sistemas multimídia como aplicações de vídeo conferência em que somente usuários autorizados devem participar ou sistemas de *pay-per-view* onde somente usuários que pagaram devem ter acesso aos dados distribuídos. Assim, esses requisitos podem ser atendido por um *middleware* com suporte a comunicação de grupo que ofereça esse controle de acesso nos dados difundidos. Em [Bessani et al., 2002] são apresentadas nossas experiências com a integração do padrão UMIOP em um ORB de código aberto, dando origem ao ORB MJACO, já em [Bessani et al., 2003] é apresentado o ReMIOP, um conjunto de extensões ao MIOP que acrescentam propriedades de confiabilidade a este. Neste artigo apresentamos uma arquitetura que provê propriedades de segurança, em especial confidencialidade e integridade, ao MIOP (e ao ReMIOP) e uma implementação desta arquitetura no MJACO.

O texto está organizado da seguinte maneira: a seção 2 apresenta uma breve descrição dos protocolos MIOP e ReMIOP. A seção 3 apresenta algumas considerações sobre a comunicação de grupo segura e sua integração ao MJACO. Na seção 4 são listados os requisitos que um gerenciador de chave deve implementar e na seção 5 é apresentada nossa proposta de arquitetura de segurança. A seção 6 apresenta as considerações sobre a implementação desta arquitetura e na seção 7 é feita uma breve análise do sistema considerando as propriedades básicas de segurança. Trabalhos relacionados são referenciados na seção 8, e as considerações finais aparecem na seção 9.

2. Os protocolos MIOP e ReMIOP

Em 2001 a OMG publicou a especificação de um protocolo de difusão não confiável baseado em *multicast* IP e um modelo de grupo de objetos que desse suporte a este protocolo em ORBs CORBA. O objetivo do padrão UMIOP é fornecer um mecanismo de comunicação multi-ponto sem garantias de entrega dentro da arquitetura CORBA. A função básica do protocolo MIOP é segmentar e encapsular as mensagens GIOP enviadas a grupos em vários pacotes (coleções) para serem transportadas via UDP/*multicast* IP. Estes pacotes contém um cabeçalho que contém uma série de campos que permitem a remontagem da mensagem original nos objetos receptores.

O modelo de comunicação convencional do CORBA não suporta grupo de objetos. Portanto, o UMIOP introduziu uma extensão a esse modelo, definindo um identificador de grupo que pode ser associado a múltiplos identificadores de objetos, que são utilizados pelo POA (*Portable Object Adapter*) para a ativação das implementações correspondentes [OMG, 2001c]. Este identificador de grupo é utilizado juntamente com a referência de grupo, que contém as informações de transporte para acesso ao grupo, para realizar a

entrega de mensagens a cada objeto membro de um grupo.

Visando acrescentar propriedades de confiabilidade (garantia de entrega de mensagem) ao MIOP, foram propostas um conjunto de extensões a este, dando origem ao protocolo ReMIOP [Bessani et al., 2003]. O ReMIOP é um protocolo de difusão confiável e escalável iniciado pelo receptor nos moldes de protocolos como SRM [Floyd et al., 1997] e LRMP [Liao, 1998]. Para dar confiabilidade ao MIOP, o ReMIOP acrescenta mecanismo de NACKs (pedidos de retransmissão) e controle de fluxo. Estas extensões foram integradas ao MIOP de forma a não prejudicar a portabilidade deste protocolo, tornando os ORBs que implementam o ReMIOP compatíveis com o mesmo.

3. Considerações Sobre a Integração de Segurança ao MJACO

Muitas aplicações que se utilizam de suportes de comunicação de grupo tem requisitos de segurança. Em [Hardjono and Tsudik, 1997] é apresentada uma série de questões relacionadas à segurança no *multicast* IP. Como o MIOP se baseia neste protocolo e trabalha sobre as mesmas premissas (sistema de comunicação não-confiável e assíncrono), é razoável imaginar que pelo menos uma parte destas questões sejam pertinentes a ele:

- **Confidencialidade e Autenticação de Emissores:** A escolha do mecanismo criptográfico a ser utilizado na implementação destes requisitos deve sempre considerar a aplicação em questão, em sistemas de comunicação de grupo não é diferente: com criptografia simétrica temos uma única chave compartilhada pelos membros do grupo, já com mecanismos assimétricos os membros do grupo compartilham uma chave privada e os emissores possuem a chave pública, existem também esquemas mistos, onde os dois tipos de chaves são utilizados para tarefas diferentes;
- **Gerenciamento de Chave:** Uma vez definido o mecanismo criptográfico a ser utilizado pelo grupo resta a definição de uma política de gerenciamento de chaves que garanta que apenas membros legítimos do grupo tenham acesso aos dados difundidos neste. Um gerenciador de chave de grupo deve gerar e distribuir a chave compartilhada aos membros legítimos do grupo. No caso de criptografia simétrica teríamos grupos fechados (somente os membros teriam a chave para poder enviar mensagens), já no caso de criptografia assimétrica teríamos grupos abertos visto que todo *host* teria acesso a chave pública do grupo, ficando a chave privada apenas para os membros do mesmo;
- **Políticas de Segurança para Grupos:** A correta definição, implementação e manutenção das políticas acerca dos vários aspectos da segurança do tráfego *multicast* é um fator fundamental para a segurança em um ambiente inseguro como a Internet;
- **Certificação para Grupos:** A certificação de grupos passa por dois tópicos específicos: a certificação do grupo em si e a certificação dos membros do grupo. As dificuldades para a utilização de certificados em sistemas de comunicação de grupo dizem respeito principalmente a escalabilidade e ao dinamismo do grupo.

Note que todo sistema de suporte a comunicação segura de grupo deve realizar o gerenciamento das chaves compartilhadas. Assim a figura do gerenciador de chaves é sempre de fundamental importância.

4. O Gerenciador de Chaves de Grupo

O método mais utilizado para controlar o acesso à dados difundidos em um grupo é a criptografia. Através deste método, dados codificados são enviados ao grupo e apenas os membros que estejam de posse da chave corrente (chamados membros legítimos) são capazes de decodificar a mensagem. O gerenciador de chaves de grupo é o componente do sistema responsável pela distribuição desta chave de grupo e também pela sua atualização (*re-key*) em certas condições prescritas na política de segurança definida para o grupo.

Existem alguns requisitos que um gerenciador de chaves de grupo deve atender, conforme apresentado em [Hardjono and Tsudik, 1997]: Escalabilidade, Independência (em termos do suporte de comunicação), Confiabilidade (os membros do grupo devem confiar no gerenciador) e Segurança (a entrega de chaves deve ser feita via um mecanismo seguro). Procurando atender a estes requisitos, é proposto um objeto de serviço CORBA que faça o gerenciamento de chaves de grupo, este objeto de serviço é central em nossa arquitetura de segurança.

5. A Arquitetura de Comunicação de Grupo Segura do MJACO

A arquitetura desenvolvida para integrar aspectos de segurança na comunicação de grupo do MJACO utiliza-se de um mecanismo de criptografia com chave simétrica integrado ao ORB de tal forma que somente os membros que se registrem no gerenciador de chaves tenham acesso à chave atual do grupo, podendo assim realizar comunicações com os demais membros. A figura 1 apresenta esta arquitetura.

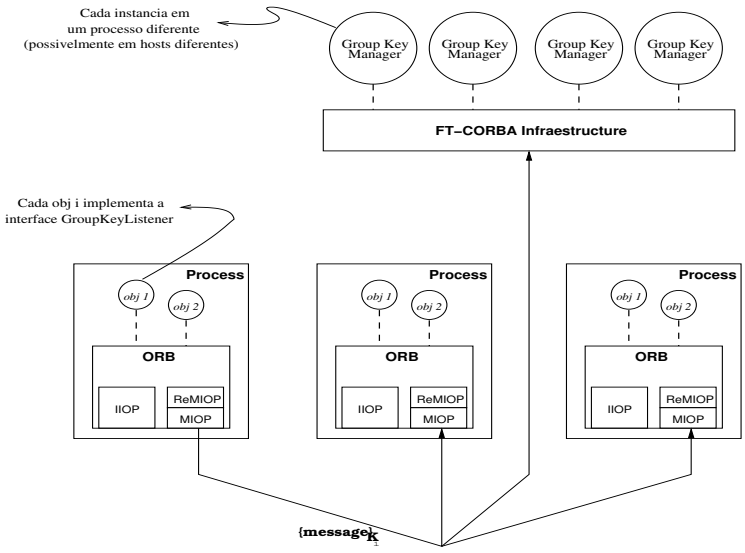

Figura 1: Arquitetura do Sistema com o Gerenciador de Chaves.

Na figura 1 temos uma série de objetos distribuídos por diversos processos, eles trocam mensagens entre si utilizando o protocolo ReMIOP e cifrando as mensagens através de uma chave K_i, que foi distribuída pelo gerenciador de chaves. Este gerenciador é replicado sobre uma infraestrutura FT-CORBA [OMG, 2001a], a fim de oferecer maior disponibilidade para o serviço e evitar que este seja um ponto de falha único. Este gerenciador tem dois propósitos básicos: manter a lista dos membros dos grupos (visão)

atualizada, computando inclusões e remoções de membros, e distribuir chaves de grupo para os membros legítimos. As comunicações entre os membros do grupo e o gerenciador de chaves são feitas sempre em canais ponto a ponto confiáveis e seguros, gerenciados pelos serviços de segurança CORBAsec [OMG, 2001b].

O gerenciador de chaves atualiza e distribui novas chaves de grupos em resposta a determinados eventos, são eles:

- **Alteração na visão do Grupo:** A cada inclusão ou remoção de membros do grupo a chave deve ser trocada para que novos membros não tenham acesso a mensagens anteriores à sua inclusão e membros antigos não tenham acesso a mensagens posteriores à sua remoção;
- **A cada t_{ck} segundos:** A cada t_{ck} segundos a chave é trocada para que não exista tempo hábil para que ela seja quebrada através de alguma técnica computacional.

Apesar da arquitetura suportar a troca de chaves sempre que ocorrer qualquer um destes eventos, para cada grupo é possível definir uma política de atualização de chave diferenciada, refletindo desta forma os requisitos específicos de cada aplicação.

Note que este esquema exige um controle de acesso sobre o gerenciador de chaves de grupo, e nossa arquitetura prevê que este controle seja implementado transparentemente através da utilização dos serviços de autenticação e autorização do CORBAsec nível 1 [OMG, 2001b].

6. Implementação da Arquitetura no MJACO

A implementação da arquitetura apresentada na seção anterior foi realizada através de um objeto de serviço gerenciador de chaves, que deve ser instalado na infraestrutura FT-CORBA para replicação, e de um *plug-in* para a criptografia das mensagens. Este *plug-in* deve ser acessível aos objetos ativos no ORB, em especial membros de grupos seguros, para que estes possam trocar a chave utilizada na criptografia das mensagens quando o gerenciador distribuir uma nova chave. As interfaces definidas para o gerenciador de chaves são apresentadas na figura 2.

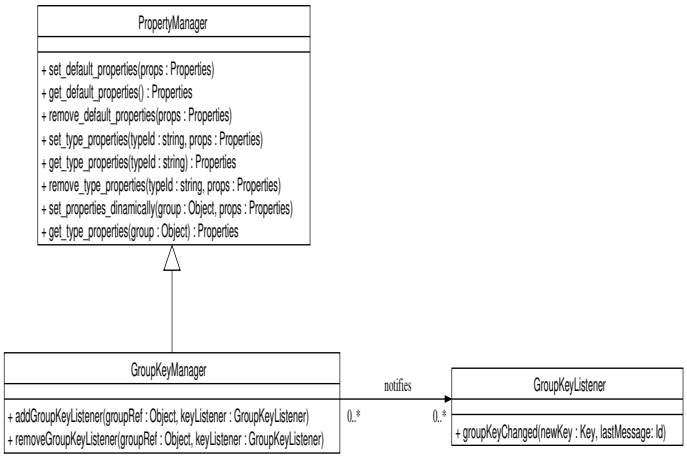

Figura 2: Interfaces Utilizadas na Implementação.

A principal interface da figura 2 é a `GroupKeyManager`. Esta interface define métodos para a inclusão e remoção de membros de um grupo e também para o gerenciamento de propriedades, estes últimos, herdados da interface `PropertyManager`, definida nas especificações UMIOP. Esta interface contém os método que permitem a definição de políticas para todos os grupos, para um tipo de grupo (tipo da IDL do grupo) ou para um grupo específico. Dentre as propriedades que podem ser definidas estão, t_{ck} (tempo de troca de chave), algoritmo de criptografia simétrico utilizado pelo grupo, e o modo que a criptografia é feita (por pacote ou por mensagem), entre outras.

Cada objeto que queira se tornar membro de um grupo seguro deve implementar a interface `GroupKeyListener`. O método `groupKeyChanged`, definido nesta interface, é invocado pelo gerenciador de chaves nos membros de um grupo sempre que ocorre uma atualização na chave compartilhada deste.

O mecanismo de criptografia implementado no *plug-in* utiliza-se da API padrão JCE (*Java Cryptography Extensions*), disponível na distribuição básica da linguagem Java.

Atualmente existem duas dificuldades práticas para a implementação da arquitetura completa sobre o MJACO: *(i.)* O GROUPPAC [Lung et al., 2000], implementação do FT-CORBA utilizada, ainda não está completamente portado para o MJACO; *(ii.)* O MJACO não implementa os serviços do CORBAsec, principalmente os de nível 2, apenas o suporte a SSL para comunicação ponto a ponto segura é suportado. Estas dificuldades dizem respeito a compatibilidade entre ORBs e a integração UMIOP/FT-CORBA/CORBAsec, e devem ser vencidas a medida que as implementações forem evoluindo.

7. Análise das Propriedades de Segurança da Implementação

Se analisarmos as propriedades básicas relacionadas à segurança [Landwehr, 2001] em vista da arquitetura proposta e da implementação realizada é possível verificar a robustez da solução apresentada e possíveis extensões para cobrir aspectos ainda não estudados:

- **Confidencialidade:** O controle acesso aos dados difundidos no grupo é realizado através da aplicação de algoritmos criptográficos nas mensagens. Esta criptografia pode ser feita individualmente em cada pacote ou na mensagem GIOP completa antes dela ser segmentada e enviada. O algoritmo criptográfico utilizado também pode ser definido na política do grupo;
- **Integridade:** A integridade é reforçada através do uso do algoritmo MD5 para obtenção de *hashs* para as mensagens GIOP enviadas. A utilização do ReMIOP sobre o MIOP permite identificar mensagens com *hash* incoerente, e a utilização de pedidos de retransmissões (NACKs) para a obtenção das mensagens corretas. Uma forma de melhorar a integridade das mensagens seria a utilização de assinaturas nas difusões, desta forma, os receptores saberiam que os emissores das mensagens são legítimos;
- **Disponibilidade:** Os pontos fundamentais para garantir a disponibilidade das informações são garantir a recepção das mensagens e das chaves de grupo pelos usuários legítimos. Estas duas garantias são reforçadas através da utilização do protocolo ReMIOP para a difusão e da replicação do gerenciador de chaves;

- **Autenticação:** Este aspecto não é tratado na versão atual da arquitetura, entretanto uma possível extensão seria a utilização de assinaturas para emissores e o uso de certificados;
- **Não repudiação:** Também não é tratado na arquitetura. A não repudiação, neste caso só poderia ser alcançada se tivéssemos autenticação para mensagens difundidas e a integração destas com o mecanismo de auditoria definido no nível 2 do CORBAsec, não implementado no MJACO.

Tendo em vista esta análise pode-se notar que a principal característica a ser implementada no sistema diz respeito a autenticação dos membros do grupo, em especial dos emissores. Além disso a implementação das especificações de segurança no MJACO, em especial o mecanismo de auditoria, permitiria a não repudiação de mensagens.

8. Trabalhos Relacionados

Até onde sabemos, a literatura não apresenta nenhuma iniciativa de se integrar propriedades de segurança ao MIOP, muito menos considerando a integração do CORBAsec e do FT-CORBA para prover essas extensões.

Alguns suportes de comunicação de grupo, em especial HORUS e Ensemble [van Renesse et al., 1998], apresentam a capacidade de composição de microprotocolos para prover diferentes tipos de serviços, inclusive de segurança. A arquitetura de segurança destes sistemas é baseada no modelo desenvolvido em [Reiter et al., 1994].

Existem uma série de trabalhos que enfocam o gerenciamento de chaves de grupo, principalmente em termos de protocolos de distribuição de chave. No protocolo GKMP (*Group Key Management Protocol*) [Harney and Muckenhirn, 1997] cada grupo multicast tem um controlador de grupo dedicado que compartilha uma chave simétrica com cada membro. O protocolo SKMD (*Scalable Multicast Key Distribution*) [Ballardie, 1996] prevê a utilização das entidades de roteamento definidas no protocolo CBT (*Core Based Trees*) para a distribuição das chaves pela árvore de roteamento. Portanto, este protocolo fere o requisito que prevê a independência do gerenciador de chaves de grupo. Alguns trabalhos utilizam a noção de subgrupos melhorando assim a escalabilidade do sistema. Por exemplo, no sistema IOLUS [Mittra, 2000] o grupo é divido em subgrupos hierárquicos, cada um com sua própria chave compartilhada.

9. Considerações Finais

Este trabalho apresentou uma arquitetura que provê confidencialidade e integridade em comunicações de grupo, com confiabilidade ou não. A arquitetura proposta propõem a utilização de algumas especificações já consolidadas dentro da arquitetura CORBA, em especial o FT-CORBA e o CORBAsec, para atender os requisitos exigidos de um gerenciador de chaves de grupo, componente central da arquitetura.

Este trabalho compreende os primeiros passos no sentido de integrar as especificações FT-CORBA e CORBAsec para prover um *middleware* baseado no padrão CORBA para sistemas tolerantes a intrusão [da Silva Fraga and Powell, 1985]. Estes sistemas toleram qualquer intrusão em uma parte do sistema, mantendo as propriedades básicas de segurança no mesmo.

Referências

Ballardie, T. (1996). Scalable multicast key distribution (rfc 1949). IETF Request For Comments.

Bessani, A. N., da Silva Fraga, J., and Lung, L. C. (2002). Mjaco - integração do multicast ip na arquitetura corba. In *Anais do 20o. Simpósio Brasileiro de Redes de Computadores*, Buzios - RJ - Brasil.

Bessani, A. N., Lung, L. C., and da Silva Fraga, J. (2003). Remiop: Projeto e implementação de um mecanismo de difusão confiável no corba. In *Anais do 21o. Simpósio Brasileiro de Redes de Computadores*, Natal - RN - Brasil.

da Silva Fraga, J. and Powell, D. (1985). A fault and intrusion-tolerant file system. In *Proceeedings of the IFIP 3rd Int. Conf. on Computer Security*, pages 203–218.

Deering, S. E. (1986). Host extensions for ip multicasting (rfc 988). IETF Request For Comments.

Floyd, S., Jacobson, V., Liu, C.-G., McCane, S., and Zhang, L. (1997). A reliable multicast framework for light-weight session and application level framing. *IEEE/ACM Transactions on Networking*.

Freier, A. O., Karlton, P., and Kocher, P. C. (1996). *The SSL protocol - version 3*. Internet Draft.

Hardjono, T. and Tsudik, G. (1997). Ip multicast security: Issues and directions. In *Annales de Telecom*, pages 324–340.

Harney, H. and Muckenhirn, C. (1997). Group key management protocol (gkmp) architecture (rfc 2094). IETF Request For Comments.

Landwehr, C. E. (2001). Computer security. online: 27 de Julho - Springer-Verlag.

Liao, T. (1998). Light-weight reliable multicast protocol. Disponível em http://webcanal.inria.fr/lrmp/.

Lung, L. C., da Silva Fraga, J., Farines, J. M., and Oliveira, J. R. (2000). Experiências com comunicação de grupo nas especificações fault tolerant corba. In *Anais do 18o. Simpósio Brasileiro de Redes de Computadores*, Belo Horizonte - MG - Brasil.

Mittra, S. (2000). The iolus framework for scalable secure multicasting. In *Proceedings of ACM SIGCOMM'97*, pages 277–288.

OMG (2001a). The common object request broker architecture specification v2.6. OMG Standart.

OMG (2001b). Security service v1.7. OMG Document 01-03-08.

OMG (2001c). Unreliable multicast inter-orb protocol specification v1.0. OMG Standart.

Reiter, M., Birman, K. P., and Renesse, R. V. (1994). A security architetcure for fault-tolerant systems. *ACM Transactions on Computer Systems*, 12(4):340–371.

van Renesse, R., Birman, K. P., Mark Hayden, A. V., and Karr, D. (1998). Building adaptative systems using ensemble. *Software - Pratice and Experience*, 28(9):963–979.