UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

APLICABILIDADE DO MODELO RBAC NO CONTROLE DE ACESSO PARA A REDE SEM FIO DO SENADO FEDERAL

HERALDO VIEIRA DA CONCEIÇÃO
ROBERTO DE OLIVEIRA SILVA

ORIENTADOR: ROBSON DE OLIVEIRA ALBUQUERQUE

MONOGRAFIA DE ESPECIALIZAÇÃO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: UNB.LABREDES.MFE 003/2006

BRASÍLIA / DF: 08/2006

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

APLICABILIDADE DO MODELO RBAC NO CONTROLE DE ACESSO PARA A REDE SEM FIO DO SENADO FEDERAL

HERALDO VIEIRA DA CONCEIÇÃO
ROBERTO DE OLIVEIRA SILVA

MONOGRAFIA DE ESPECIALIZAÇÃO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ESPECIALISTA.

APROVADA POR:

ROBSON DE OLIVEIRA ALBUQUERQUE, Mestre, UnB
(ORIENTADOR)

ANDERSON CLAYTON ALVES NASCIMENTO, Doutor, UnB
(EXAMINADOR INTERNO)

ODACYR LUIZ TIMM JR, Mestre, OM
(EXAMINADOR)

DATA: BRASÍLIA/DF, 29 de agosto de 2006.

FICHA CATALOGRÁFICA

CONCEIÇÃO, HERALDO VIEIRA DA
SILVA, ROBERTO DE OLIVEIRA
Aplicabilidade do Modelo RBAC no Controle de Acesso para a Rede Sem Fio do Senado Federal [Distrito Federal] 2006.
xiv,76p., 297 mm (ENE/FT/UnB, Especialista, Engenharia Elétrica, 2006).

Monografia de Especialização – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. RBAC 2. Controle de acesso
3. Rede sem fio 4. Wireless

I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

CONCEIÇÃO, HERALDO VIEIRA DA e SILVA, ROBERTO DE OLIVEIRA. 2006. Aplicabilidade do Modelo RBAC no Controle de Acesso para a Rede Sem Fio do Senado Federal. Monografia de Especialização, Publicação UNB.LABREDES.MFE 003/2006, Departamento de Engenharia Elétrica, Universidade de Brasília , Brasília , DF, 76p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Heraldo Vieira da Conceição e Roberto de Oliveira Silva
TÍTULO DA DISSERTAÇÃO: Aplicabilidade do Modelo RBAC no Controle de Acesso para a Rede Sem Fio do Senado Federal.
GRAU/ANO: Especialista/2006.

Heraldo Vieira da Conceição
SQS 216 – bloco E – ap. 504
CEP 70.295-050 – Brasília – DF - Brasil

Roberto de Oliveira Silva
SQN 304 – bloco F – ap. 114
CEP 70.736-060 – Brasília – DF - Brasil

Às nossas famílias, pela educação, pela oportunidade e pelo apoio que nos permitiu chegar a esse ponto.

À Lili, pela generosidade, compreensão, paciência, incentivo, força, amizade e companheirismo.

AGRADECIMENTOS

Ao nosso orientador Professor Robson de Oliveira Albuquerque, pelo tempo, paciência, apoio e incentivo dedicados à orientação para o desenvolvimento deste trabalho.

Aos colegas do SIER – Serviço de Infra-estrutura de Rede e do SPB – Serviço de Projetos Especiais B, do PRODASEN, pelas conversas enriquecedoras, colaboração e amizade.

Ao PRODASEN, pela iniciativa de patrocinar, institucional e financeiramente, um curso de pós-graduação como este.

A todos, nossos sinceros agradecimentos.

A Deus pelo dom da vida e por nos permitir vivê-la com saúde.

Aplicabilidade do Modelo RBAC no Controle de Acesso para a Rede Sem Fio do Senado Federal

RESUMO

A Tecnologia da Informação oferece atualmente diversas opções tecnológicas, amplamente utilizadas no processamento eletrônico de dados e na comunicação entre pessoas e instituições. Uma dessas opções são as redes sem fio (*Wireless Networks*), que têm se tornado bastante populares, tanto no segmento institucional/empresarial quanto no doméstico/residencial. Qualquer tecnologia aplicada ao processamento eletrônico, utilizada para produção, distribuição e compartilhamento de informações, requer mecanismos apropriados de controle de acesso com o objetivo de garantir sua autenticidade, confidencialidade e integridade. O RBAC (*Role Based Access Control*) ou Controle de Acesso Baseado em Papéis (ou Perfis) é um modelo de controle de acesso para proteção de informações e recursos em ambientes informatizados.

Este trabalho apresenta uma descrição do RBAC, mostrando sua funcionalidade, vantagens e implicações, com ênfase na sua aplicação na rede sem fio do Senado Federal. Pretende-se identificar os cenários em que isso pode acontecer, propondo também uma arquitetura de componentes de *hardware* e *software* necessários para a implementação do modelo RBAC.

ABSTRACT

Information Technology offers several technological options widely used into the electronic processing of data and on communication among people and institutions. One of those options is Wireless Networks, which have become quite popular, either into the institutional/business community or into the domestic/residential segment. The technology applied to electronic processing, used for production, distribution and sharing of information, requires appropriate mechanisms of access control with the objective of guaranteeing authenticity, confidentiality and integrity. The RBAC (Role Based Access Control) is a model of access control for protection of information and resources in computerized environments.

This work presents a description of the RBAC model, showing its functionality, advantages and implications, with emphasis in its application on the Wireless Network of Brazilian Federal Senate. It intends to identify the scenarios in which RBAC may fit, also proposing an architecture of hardware and software components necessary for implementation of the RBAC model.

ÍNDICE

Capítulo **Página**

ÍNDICE DE TABELAS

Tabela Página

ÍNDICE DE FIGURAS

Figura Página

ÍNDICE DE ABREVIAÇÕES

ACL	*Access Control List*
ANSI	*American National Standards Institute*
AP	*Access Point*
ATM	*Asynchronous Transfer Mode*
CBAC	*Content Based Access Control*
CDAC	*Content Dependent Access Control*
CDMA	*Code Division Multiple Access*
DAC	*Discretionary Access Control*
DSD	*Dynamic Separation of Duty*
GSM	*Global System for Mobile Communications*
HTTP	*Hiper Text Transfer Protocol*
HTTPS	*Hiper Text Transfer Protocol Secure*
IDS	*Intrusion Detection System*
IEEE	*Institute of Electrical and Electronics Engineers*
INCITS	*International Committee for Information Tecnology Standards*
IP	*Internet Protocol*
IPSEC	*Internet Protocol Security*
ISO	*International Organization for Standardization*
JMX	*Java Management Extensions*
LAN	*Local Area Network*
LDAP	*Lightweight Directory Access Protocol*
LLC	*Logical Link Control*
MAC	*Mandatory Access Control*
MAC	*Media Access Control*
MAN	*Metropolitan Area Network*
MANETS	*Mobile Ad Hoc Networks*
Mbps	*Mega Bits por Segundo*
NAT	*Network Address Translation*
NBR	*Normas Brasileiras*
NIST	*National Institute of Standards and Technology*
PBAC	*Policy Based Access Control*

PDA	*Personal Digital Assistant*
QoS	*Quality of Service*
RADIUS	*Remote Authentication Dial-in User Service*
RBAC	*Role Based Access Control*
RSBAC	*Rule Set Base Access Control*
SSD	*Static Separation of Duty*
SSO	*Single Sign On*
TI	*Tecnologia da Informação*
TKIP	*Temporal Key Integrity Protocol*
UBAC	*User Based Access Control*
VBAC	*View Based Access Control*
VLAN	*Virtual Local Area Network*
VoIP	*Voice Over IP*
WAN	*Wide Area Network*
WEP	*Wired Equivalent Privacy*
WLAN	*Wireless Local Area Network*
WPA	*Wi-Fi Protected Access*
WPA2	*Wi-Fi Protected Access 2*
XML	*Extensible Markup Language*

1. INTRODUÇÃO

O surgimento do computador moderno no início da década de 1960 marcou o começo de uma era de grandes transformações e desenvolvimento de tecnologias voltadas ao processamento eletrônico de dados. Resguardadas as limitações e realidade de cada época, ano a ano, o poder revelado pelos computadores e tecnologias semelhantes foi conquistando espaço na vida das pessoas e instituições. Atualmente, a sociedade como um todo é bastante dependente da tecnologia relacionada a computadores.

A Tecnologia da Informação (TI) — entendida como a aplicação de vários ramos da tecnologia e atividades da área de informática no processamento de dados — tem como componente principal o uso do computador dentro de uma infra-estrutura que permita integrar, compartilhar, disseminar e aumentar a produtividade de informação.

A utilização isolada dos computadores evoluiu para a infra-estrutura das redes. Da perspectiva do usuário final, as Redes de Computadores são sistemas nos quais as informações são armazenadas, processadas e por meio dos quais circulam. São compostos de computadores (centrais e terminais utilizados pelos usuários), componentes ativos e passivos de transmissão (cabos, antenas, satélites, roteadores, *switches*, etc.), serviços de apoio (serviços de atribuição de endereços, serviços de resolução de nomes, serviços autenticação, etc.). Às redes, vincula-se uma crescente variedade de aplicações (sistemas de correio eletrônico – *e-mail*, navegadores e sistemas de exibição de informação, sistemas para compartilhamento e transferência de arquivos, sistemas de informação em geral, etc.), destinadas àqueles que as utilizam para realizar suas tarefas e atingir seus objetivos pessoais e profissionais.

Do ponto de vista técnico, uma Rede de Computadores é um sistema de comunicação compartilhado que suporta comunicação digital entre os computadores interligados [1]. Dependendo do tamanho e abrangência geográfica, as redes podem ser classificadas como Locais, ou LAN (*Local Area Network*) quando o conjunto se encontra dentro de uma mesma localização geral, por exemplo, no mesmo prédio; MAN (*Metropolitan Area Network*) dentro de uma área metropolitana; WAN (*Wide Area Network*) quando os computadores estão mais dispersos, abrangendo uma área ampla, podendo atingir um país ou continente [2], e consistindo, normalmente, de duas ou mais LANs. Além do aspecto geográfico, outras características técnicas são freqüentemente utilizadas para distinguir e categorizar os tipos de

redes. A topologia da rede representa o arranjo geométrico dos computadores que a compõem, sendo as mais comuns as de barramento, anel ou estrela. A arquitetura determina como a comunicação ocorre entre os pontos (nós) da rede, podendo ser ponto-a-ponto (*peer-to-peer*) ou cliente-servidor (*client-server*). Os protocolos de rede são conjuntos de regras e especificações que definem detalhadamente como os computadores devem trocar informação entre si de maneira que possam se entender. O protocolo determina o método de compressão de dados a ser utilizado, o método de verificação de erros, como os dispositivos emissor e receptor indicam o término do envio da mensagem e a confirmação do recebimento, respectivamente. O meio físico que conecta os computadores e demais dispositivos da rede abrange várias opções, podendo ser, entre outras, par trançado, coaxial, fibra ótica. Normalmente, as redes utilizam cabos como meio físico para transportar dados de um ponto a outro, fixos, em um local definido onde está instalado o computador.

Nos últimos anos, houve um rápido desenvolvimento de tecnologias de Redes Sem Fio (*Wireless Networks*), que permitem distribuir e compartilhar informação, acessar sistemas e outros recursos de uma rede mais ampla independente de um local fixo, sem uso de cabos entre o dispositivo (geralmente um *notebook*, *laptop* ou PDA) e o ponto de acesso. Aqui o termo "ponto de acesso" é colocado de uma maneira geral, sem referir-se a um conceito definido. No entanto, uma das tecnologias de rede sem fio citadas neste trabalho apresenta um componente com esse mesmo nome ou simplesmente AP (*Access Point*), cuja função é permitir a conexão do dispositivo móvel à rede. O acesso a recursos de uma rede por tecnologias sem fio tem se tornado cada vez mais popular e, exercido de maneira apropriada, permite atingir um nível de flexibilidade e conveniência com um custo relativamente baixo. Existem várias tecnologias de rede sem fio, tais como comunicação por satélite, que usa microondas para estabelecer contato com outras estruturas; telefonia celular, amplamente utilizada por usuários de operadoras que adotam padrões como o GSM ou CDMA; sistemas sem fio (*Cordless Systems*), usados dentro de casa, sala ou escritório, em pequenas distâncias para comunicação entre telefones sem fio e suas bases de operação; redes locais sem fio ou WLAN (*Wireless Local Area Networks*), usadas em organizações de qualquer porte e residências, permitindo que computadores com placas de comunicação sem fio conversem entre si ou acessem uma estrutura de rede maior. Essa última é o foco deste trabalho, considerando a sua aplicação como uma extensão da rede fixa, visando ampliar as possibilidades de acesso. Além do uso como extensão de rede, há outros tipos de utilização, que estão fora do escopo deste trabalho: conexão entre prédios; *MANETS* (*Mobile Ad Hoc*

Networks), onde os dispositivos estabelecem conexões uns com os outros sem qualquer infra-estrutura preexistente.

Atualmente, as redes locais (com e/ou sem fio) estão presentes em qualquer empresa ou instituição, governamental ou particular, inclusive nas de pequeno porte, permitindo distribuir e compartilhar informação de maneira mais ágil. A necessidade de meios mais ágeis e eficientes para disseminar e trocar informação foi a principal razão do nascimento das redes e ainda representa o motivo de seu constante desenvolvimento e adoção dentro do ambiente institucional. As redes foram concebidas com essa finalidade precípua, e permitir maior acesso à informação por diversas pessoas simultaneamente requer controles de acesso adequados ao uso que se faz dela em determinado momento. Aí surge a necessidade de atenção com a Segurança da Informação, especialmente com as tecnologias de redes sem fio, uma vez que pela sua própria natureza — comunicação por ondas de rádio pelo "ar", que podem ser capturadas por qualquer um dentro do raio de alcance — não atingem o mesmo nível de segurança física de uma rede fixa com cabos. A introdução de outras formas de acesso, como as redes sem fio, eleva o uso da rede e aumenta a capacidade de obtenção da informação, tornando o aspecto da segurança cada vez mais importante dentro da estratégia de TI da instituição. Tão importante quanto a segurança em si são os métodos disponíveis para implementá-la e o impacto que eles têm na estrutura de TI como um todo, buscando sempre uma maneira mais adequada de gerenciá-la.

Os mecanismos disponíveis na Segurança da Informação permitem proteger informações e recursos de uma rede contra diversos tipos de ameaça (acesso não autorizado, roubo ou alteração de informação), e são, também, instrumentos para tentar garantir o uso controlado de tais recursos no sentido de adequá-los à necessidade e função específicas de cada usuário dentro da instituição. Os princípios básicos de segurança que se busca garantir são a autenticidade, a confidencialidade e a integridade. Autenticidade é a propriedade de verificar a identidade da pessoa ou máquina que deseja efetuar qualquer tipo de acesso à rede. Confidencialidade diz respeito à garantia de que a informação só será vista e manipulada por pessoas ou máquinas autorizadas. Integridade é a capacidade de garantir que a informação não seja modificada, alterando (corrompendo) ou não o seu significado original, por quem não tenha a devida autorização.

Várias abordagens têm sido utilizadas para a manutenção desses princípios. Uma delas é adotar algum método de controle de acesso. Toda instituição possui funcionários que

executam atividades diferentes e exercem funções distintas, uns com mais poder e privilégios que outros. Naturalmente, controlar permissões de acesso na utilização dos recursos computacionais por meio da rede torna-se uma atividade não apenas desejável, mas essencial, especialmente quando a infra-estrutura de acesso é mais sensível a problemas relacionados à segurança, como é o caso das redes sem fio. O controle de acesso por si só não significa uma solução completa de segurança, mas representa um componente importante para atingi-la. Uma solução completa de segurança envolve, também, planejamento (definição de políticas de segurança), processos e serviços específicos, tais como comunicações seguras, autenticação, auditoria e administração de segurança.

Diversos modelos de controle de acesso têm sido desenvolvidos e incorporados aos sistemas e estruturas de comunicação. Um deles é o RBAC (*Role Based Access Control*) ou Controle de Acesso Baseado em Papéis (ou Perfis). O RBAC é um método de acesso não discricionário, ou seja, os usuários têm de se submeter às políticas de segurança estabelecidas na organização [3]. Nesse sentido, o RBAC mostra grande habilidade em implementar as políticas de forma eficiente e flexível, tornando menos oneroso o trabalho de gerenciamento dos recursos da rede.

O modelo RBAC baseia o controle de acesso na função ou papel que o usuário exerce dentro da organização [3]. Essas funções definem um elenco de atividades para determinados usuários e podem ser vistas como cargos ou posições que o indivíduo ocupa na organização, representando a autoridade requerida para efetuar tarefas correlatas. Assim, no RBAC as permissões aos recursos da rede não são atribuídas diretamente ao usuário – como acontece em modelos comuns de controle de acesso – mas sim aos papéis, e estes, por sua vez, são atribuídos aos usuários. É esta atribuição indireta que permite que o RBAC seja flexível, pois se podem atribuir vários papéis a cada usuário, e várias permissões a cada papel. A eficiência e facilidade de gerenciamento também são devidas a este mesmo fator, visto que uma vez definidos os papéis e suas permissões de acesso, basta-se associar os usuários aos papéis para definirem-se suas permissões de acesso. Do mesmo modo, uma alteração de permissão para determinado papel permite alterar as permissões dos diversos usuários atribuídos a ele de forma bem mais fácil do que alterar a permissão usuário por usuário, como acontece em outros modelos de controle.

Todos esses aspectos técnicos e a evolução da tecnologia incorporada em seus componentes mostram que as redes são sistemas complexos que requerem gerenciamento e

controle. Mais do que isso, tornando-se parte da estrutura básica da instituição, algumas funções gerenciais exercidas são comuns a outras partes da administração, porém alguns requisitos são específicos da TI, e, em função da complexidade e abrangência, não podem ser gerenciados exclusivamente pelo esforço humano, necessitando, assim, do uso de ferramentas automatizadas de gerência [4]. Dentro das principais áreas de gerenciamento de rede propostas pela *International Organization for Standardization* (ISO) [5] — Gerenciamento de Falhas, Gerenciamento de Contabilidade, Gerenciamento de Configuração e de Nome, Gerenciamento de Desempenho e Gerenciamento de Segurança —, essa última é a possui relação mais direta com o contexto deste trabalho, pois está envolvida com o monitoramento e o controle de acesso às redes e às informações.

O Senado Federal — órgão que compõe o Congresso Nacional, ao lado da Câmara dos Deputados — por meio da sua Secretaria Especial de Informática, o PRODASEN, tem procurado constantemente aperfeiçoar a sua infra-estrutura computacional e de comunicação de dados, desde a implantação da sua rede local de computadores. Hoje, a rede do Senado possui, aproximadamente, 4000 pontos ativos, distribuídos por diversos prédios (incluindo as residências dos senadores na SQS 309), utilizando tecnologia de *switches ethernet* (*fast* e *gigabit*) e ATM, além de acesso remoto via VPN.

A busca de novas soluções e tecnologias de *hardware* e *software* visa atender as necessidades dos usuários, suportando a realidade e evolução das aplicações do mundo legislativo, que possui algumas características peculiares. Dessa forma, o PRODASEN vem implementando um projeto de ampliação da rede do Senado por meio de redes sem fio, onde o plenário da Casa (local de difícil acesso para realização de obras para passagem de cabos) já adota essa tecnologia, permitindo que os senadores acompanhem e tenham acesso às informações da ordem do dia utilizando *notebooks* do tipo *Tablet PC*.

Faz parte do projeto ampliar o acesso à rede com tecnologia sem fio em outras áreas do Senado, como, por exemplo, gráfica, ala das comissões, gabinetes dos senadores, entre outras, suportando não somente aplicações tradicionais de sistemas de informação e acesso a banco de dados, mas também aquelas de áudio e vídeo, que já são uma realidade em função da presença da Rádio e TV Senado.

1.1. OBJETIVO

Considerando esse cenário, o objetivo desse trabalho é analisar a aplicação do modelo RBAC no controle de acesso da rede sem fio do Senado Federal, propondo uma arquitetura de componentes de *hardware* e *software* necessários à sua implantação. Para isso, é necessário um estudo aprofundado do modelo RBAC; uma análise das características da rede do Senado Federal, identificando suas peculiaridades, demandas de usuários, natureza e contexto das aplicações dentro da estrutura hierárquica e das atribuições funcionais do órgão; um estudo de mecanismos de segurança (métodos de acesso, autenticação, etc); um estudo de arquiteturas e técnicas de implementação de redes; um estudo das implicações das características únicas das redes sem fio no contexto do controle de acesso; um estudo de normas, padrões e práticas de gestão de segurança da informação.

1.2. ORGANIZAÇÃO DO TRABALHO

Dessa forma, este trabalho está dividido em cinco capítulos. Este capítulo introdutório apresentou as razões que nos motivaram a desenvolver o este trabalho, e o contexto no qual ele está inserido.

O capítulo 2 descreve as características e funcionalidade do modelo RBAC.

O capítulo 3 mostra as características da rede do Senado Federal, identificando demandas dos usuários, relacionando-as com o contexto do trabalho.

O capítulo 4 identifica os cenários onde a funcionalidade do modelo RBAC se aplica, apresentando uma arquitetura para sua adoção.

Finalmente, o último capítulo apresenta conclusões e possibilidades de continuação do trabalho.

2. ROLE-BASED ACCESS CONTROL

O conceito do RBAC existe a cerca de 30 anos, porém somente tornou-se um modelo maduro há aproximadamente 15 anos. Nesse período, o crescimento das redes de computadores fez com que o nível de intercâmbio de informações tornasse evidente a necessidade de mecanismos segurança, dentre estes o controle de acesso. Nesse contexto, o RBAC surgiu como uma alternativa viável ao DAC e ao MAC.

Uma instituição que fez grandes contribuições para o desenvolvimento do RBAC foi o NIST, *National Institute of Standards and Technology*. Em 1992, o NIST publicou seu primeiro modelo do RBAC. Nos anos de 1997 e 1998, publicou outros artigos que expandiram seu modelo original com a incorporação de diferentes tipos de relacionamentos entre papéis. Também em 1997, publicou documentos sobre ferramentas específicas para implementação de RBAC na *web*. Em 1995 demonstrou o uso do RBAC em sistemas da área de saúde, e, em 1999, em *Intranets* [6].

Mais recentemente, em 2003, desenvolveu o *draft* do padrão para RBAC, proposto pelo ANSI, que foi aprovado em 2004 como ANSI/INCITS 359-2004. Em janeiro deste ano, 2006, desenvolveu a versão 0.1 do *draft* do padrão para implementação do RBAC.

O modelo RBAC proposto pelo NIST tem como raízes os grupos do Unix, o agrupamento de privilégios em sistema gerenciadores de banco de dados e o conceito de separação de tarefas [6]. O sistema operacional Unix possui o papel de administrador, com algumas permissões de acesso específicas para o desempenho desse papel. O RBAC herdou esse conceito.

Alguns papéis podem refletir competências como digitador ou programador; outros refletem responsabilidades ou autoridade do usuário em relação aos recursos, tais como: gerente de projetos ou supervisor ou revisor. Ter competência não implica em ter a responsabilidade. Assim um programador pode compor a equipe de vários projetos, mas ser gerente de apenas um, tendo neste, portanto, as responsabilidades associadas à gerência. O modelo RBAC deve refletir todas estas acepções da palavra papel.

No caso do RBAC, o termo papel define não só os usuários que devem ter as permissões, mas também os recursos que podem ser acessados. Assim, ao se associar um usuário a um papel, concede-se a este todas as permissões de acesso a recursos associados ao

papel. Inversamente, ao se associar a um papel permissões de acesso a um recurso, concede-se permissão de acesso a todos os usuários associados ao papel.

A escolha do RBAC pelos administradores baseia-se principalmente na facilidade deste mapear as funções existentes na empresa. Desta forma, o controle de acesso é feito alterando-se o papel do usuário conforme sua função na empresa, e.g. programador, gerente de projeto, etc. Da mesma forma, novos recursos podem ser atribuídos a vários usuários concedendo-se permissão no recurso para um papel.

Outra característica importante do RBAC é a obediência aos princípios de segurança de atribuição do menor privilégio e a de separação de tarefas.

2.1. MÉTODOS DE CONTROLE DE ACESSO

Este capítulo inicia resumindo os diversos tipos de controle de acesso, destacando as características que os diferenciam conforme descrito por Camelot [7]. Em seguida, cita alguns termos e definições utilizados para descrever o RBAC, extraídas do *draft* do documento padronizador ANSI/INCITS 359 [8]. Posteriormente, descreve o modelo de referência do RBAC proposto por Ferraiolo *et al*.[9] e a especificação funcional.

A segurança de redes tem como um de seus objetivos principais efetuar o controle de acesso de usuários ou computadores a objetos da rede. Para que esse objetivo seja contemplado de maneira mais ampla, diversas ferramentas podem ser utilizadas de acordo com o ambiente em questão. *Firewalls*, *Intrusion Detection System* (IDS) e criptografia são algumas das ferramentas mais usadas. Em muitos casos, no entanto, são usadas apenas as ferramentas fornecidas pelo sistema operacional.

Os métodos de controle de acesso mais utilizados baseavam-se em Matriz de Controle de Acesso e suas variações: Lista de Controle de Acesso (ACL, do inglês *Access Control List*), e *Capabilities.* Métodos mais recentes incluem: *User Based Access Control*, *Policy Based Access Control*, *Context Based Access Control*, *Mandatory Access Control* ou *Discretionary Access Control.*

Qualquer que seja o método utilizado, o grande desafio do administrador do sistema é conceder ao usuário exatamente aqueles direitos de acesso que ele necessita para cumprir seus

deveres na organização. Nem um direito a mais ou a menos deve ser fornecido. Isto é o princípio do mínimo privilégio [7].

Certamente, o grau de dificuldade que o administrador irá enfrentar para chegar ao mínimo privilégio está diretamente relacionado com o método de controle de acesso utilizado e o sistema a ser administrado.

Um conceito utilizado para indicar a eficiência do método de controle de acesso é a granularidade. Diz-se que um método de controle de acesso tem granularidade alta quando ele consegue atribuir direitos de acesso para cada usuário em cada objeto da rede. A granularidade é dita como baixa quando os direitos de acesso são dados para conjuntos de usuários em conjuntos de objetos. Quanto mais alta for a granularidade do método de acesso, mais próximo do menor privilégio será possível alcançar com esse método [7].

As subseções a seguir descrevem os vários métodos, destacando seus pontos fortes e aplicabilidade.

2.1.1. Matriz de Controle de Acesso

A Matriz de Controle de Acesso é o modelo mais simples de controle de acesso. Neste modelo é utilizada uma matriz para definir as permissões que os sujeitos (usuários ou processos) terão ao acessar os objetos (arquivos ou outros recursos da rede).

A Tabela 2.1 exemplifica o método em um sistema operacional hipotético. O processo 1 pode ler os arquivos 1 e 2 e escrever no arquivo 1. Adicionalmente, o processo 1 pode se comunicar com o processo 2 enviando mensagens. O processo 2, por sua vez, pode ler os arquivo 2, escrever no arquivo 1, e receber mensagens do processo 1.

Tabela 2.1 – Exemplo de matriz de controle de acesso

	arquivo 1	arquivo 2	processo 1	processo 2
processo 1	ler, escrever	ler	ler, escrever, executar	escrever
processo 2	escrever	ler	ler	ler, escrever, executar

O significado dessas permissões pode não ser muito intuitivo. Ler e escrever arquivos é algo comum, porém “ler um processo" pode ter um significado específico em cada sistema operacional. Neste exemplo, a permissão de “ler um processo” concede ao processo 2 o recebimento de mensagens do processo 1, mas poderia também significar comunicação através do uso de uma área compartilhada de memória.

A Matriz de Controle de Acesso pode parecer a forma ideal para controlar acessos, contudo, sua implementação de forma direta em um sistema com grande quantidade de sujeitos e objetos pode gerar uma matriz gigantesca que ocuparia grande área de armazenamento e seria de difícil gerenciamento. Daí o surgimento de simplificações tais como ACL e *Capabilities*.

2.1.2. Lista de Controle de Acesso (ACL)

A Lista de Controle de Acesso é a variação mais utilizada da Matriz de Controle de Acesso. Grande parte dos sistemas operacionais comerciais utiliza este modelo. O que o diferencia do modelo original é a forma de armazenar as permissões. Esse modelo armazena as permissões junto aos objetos por meio de pares formados pelo sujeito e suas permissões de acesso. O exemplo da Tabela 2.1 está reescrito como ACL na Tabela 2.2.

Tabela 2.2 – Exemplo de lista de controle de acesso

objeto	ACL
arquivo 1	[processo 1, (ler, escrever)], [processo 2, (escrever)]
arquivo 2	[processo 1, (ler)], [processo 2, (ler)]
processo 1	[processo 1, (ler, escrever, executar)], [processo 2, (ler)]
processo 2	[processo 1, (escrever)], [processo 2, (ler, escrever, executar)]

Semelhantemente à Matriz de Controle de Acesso, ACL pode se tornar muito grande em sistemas com grande quantidade sujeitos e objetos. Para viabilizar o uso de ACL, o Unix, por exemplo, utiliza grupos como sujeitos e não usuários. Esta medida reduz drasticamente a ACL, em contrapartida à perda de granularidade.

2.1.3. Capabilities

Capabilities é uma variação da Matriz de Controle de Acesso em que se armazenam as permissões junto aos sujeitos por meio de pares formados pelo objeto e permissões de acesso

que o sujeito possui naquele objeto. O exemplo da Tabela 2.1 está reescrito através de *Capabilities* na Tabela 2.3.

Tabela 2.3 – Exemplo de capabilities

sujeito	*Capabilities*
processo 1	[arquivo 1, (ler, escrever)], [arquivo 2, (ler)], [processo 1, (ler, escrever, executar)], [processo 2, (escrever)]
processo 2	[arquivo 1, (escrever)], [arquivo 2, (ler)], [processo 1, (ler)], [processo 2, (ler, escrever, executar)]

Como com *Capabilities* as permissões estão armazenadas com o sujeito, este deve apresentá-las quando desejar acessar um objeto. Um problema com *Capabilities* é a possibilidade do sujeito forjar sua lista. Para evitar tal falha, o sistema operacional deve ter uma forma de reconhecer a autenticidade da lista de *Capabilities*.

2.1.4. User Based Access Control (UBAC)

No UBAC as permissões de acesso são dadas a cada usuário individualmente. Potencialmente, esse método permite que a granularidade seja a mais alta possível. Contudo, na prática como esse método exige que o administrador configure as permissões para cada usuário, dependendo do tamanho da rede a ser administrada, esse potencial dificilmente é efetivado.

A necessidade de ajustar as permissões de acesso para cada usuário, por si já poderia ser considerado pouco razoável, mas quando é acrescentado o fato de que os usuários têm necessidades que variam ao longo do tempo o esforço para ajuste atinge um custo proibitivo.

O que acontece em sistemas reais que utilizam esse modelo é que as permissões de acesso são dadas em conjuntos de usuários, não se alcançando assim grande granularidade.

A figura 2.1 representa o UBAC graficamente.

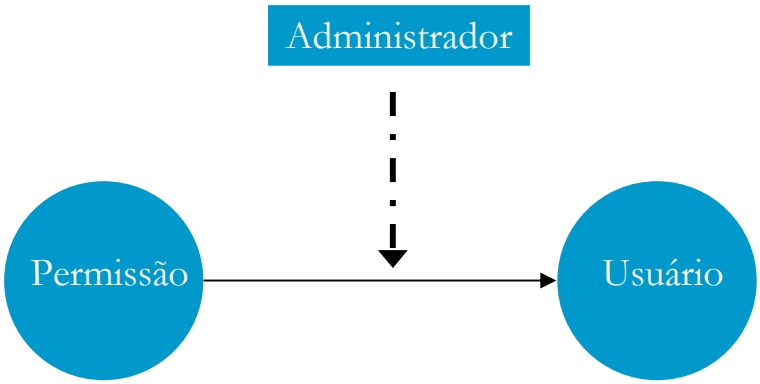

Figura 2.1 - UBAC

2.1.5. Policy Based Access Control (PBAC)

No PBAC a política de acesso é definida por um conjunto de regras que determinam os recursos ao qual o usuário terá permissão de acesso. Por esta característica, o PBAC é também conhecido como *Rule Set Base Access Control* (RSBAC), isto é, controle de acesso baseado em um conjunto de regras.

A implementação do PBAC é muitas vezes feita por meio de ACL. Contudo, os melhores resultados em implementação do controle de acesso por PBAC são obtidos através de linguagem de especificação apropriada para descrever as políticas.

A figura 2.2 representa o PBAC graficamente.

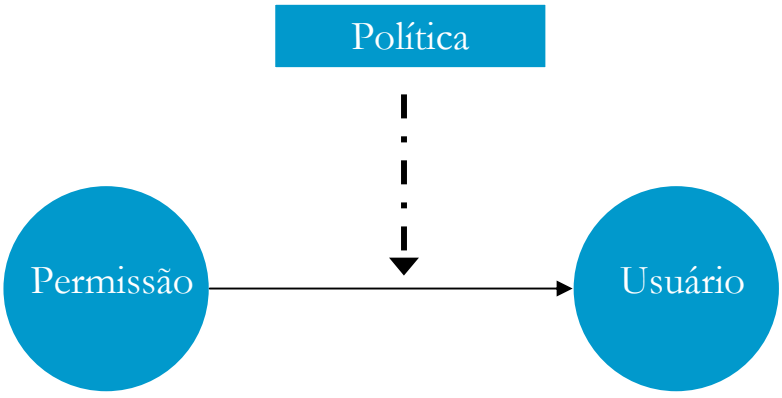

Figura 2.2 – PBAC

2.1.6. Content Dependent Access Control (CDAC)

CDAC é um método para controlar o acesso de usuários a recursos baseado no conteúdo do recurso. É usado, primariamente, para proteger bases de dados contendo dados potencialmente sensíveis.

Um exemplo do uso de CDAC seria de um sistema de armazenamento de prontuário em um hospital. No caso de um paciente diagnosticado com AIDS, o acesso ao prontuário seria restrito ao médico responsável.

Para determinar a permissão de acesso, no caso do CDAC, é necessário analisar o conteúdo, acarretando em *overhead* (sobrecarga) quando o recurso é acessado.

A figura 2.3 representa o CDAC graficamente.

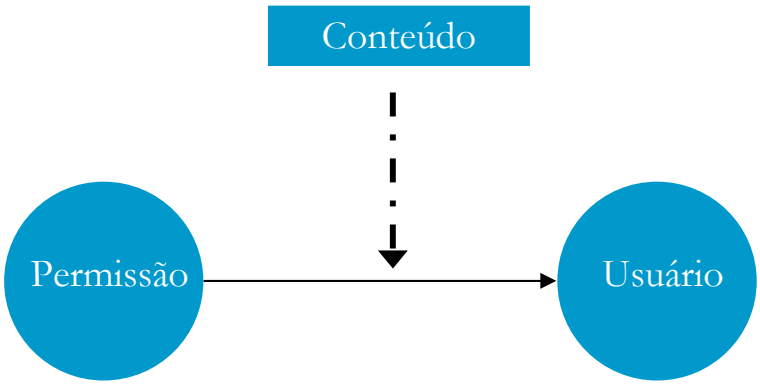

Figura 2.3 – CDAC

2.1.7. Context Based Access Control (CBAC)

Apesar do nome parecido com o CDAC, o CBAC trata de outra característica totalmente distinta. Nesse tipo de controle de acesso o que define a permissão de acesso ao recurso é o contexto em que o acesso se dá, isto é, os fatos que ocorreram previamente.

Um exemplo desse tipo de controle de acesso no dia-a-dia ocorre em máquinas de saque eletrônicas. Nessas máquinas observamos que ainda que o usuário tenha a conta e a senha, se o montante sacado no dia tiver alcançado o limite diário, não terá permissão para sacar mais.

Este tipo de controle de acesso também ocorre em alguns *firewalls* denominados *statefull*. Nesse tipo de *firewall*, os pacotes além de serem analisados quanto à origem e destino, são verificados se pertencem a uma conexão previamente permitida.

A figura 2.4 representa o CBAC graficamente.

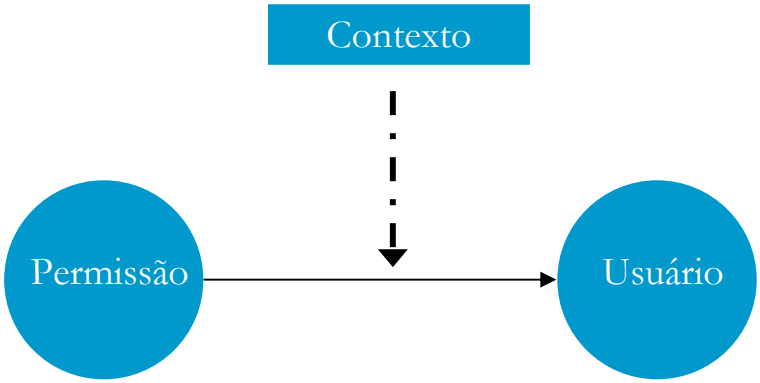

Figura 2.4 – CBAC

2.1.8. View Based Access Control (VBAC)

Este tipo de controle de acesso é utilizado basicamente em sistemas de banco de dados onde visões (*views*) das tabelas restringem o acesso às informações nelas contidas.

Como exemplo, pode-se considerar uma tabela contendo informações sobre pacientes de um hospital. Os médicos teriam acesso a todas as informações relativas ao diagnóstico e tratamento do paciente. As enfermeiras visualizariam apenas as medicações a serem ministradas ao paciente. Já o departamento financeiro teria visão apenas dos procedimentos e materiais despendidos no tratamento para efeito de cálculo do valor devido.

O VBAC permite que se trate o controle de acesso com alta granularidade, contudo é preciso grande conhecimento da estrutura do banco de dados e das aplicações e usuários que o acessam.

A figura 2.5 representa o VBAC graficamente.

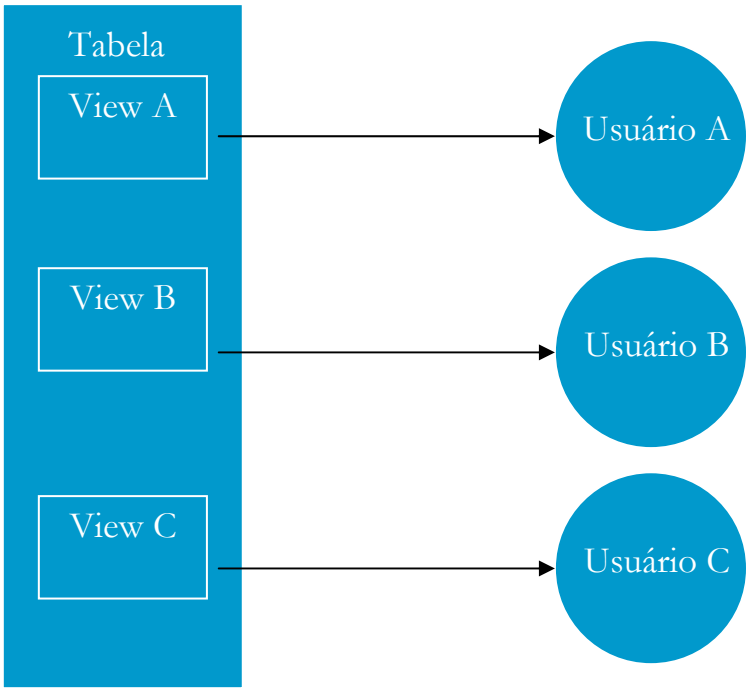

Figura 2.5 – VBAC

2.1.9. Controles de Acesso Discricionário e Mandatório

A filosofia do Controle de Acesso Discricionário (DAC – *Discritionary Access Control*) é de permitir que os usuários decidam sobre que permissões de acesso devem ser atribuídas. Desse modo, mesmo que a política de segurança defina as permissões de acesso, depende dos usuários, que possuam poderes de atribuir permissões, para que a implementação obedeça à política. Assim, esse modelo é bastante sensível ao nível de comprometimento e de conhecimento dos usuários. Esse tipo de acesso é comum em sistemas de arquivo, onde os usuários definem as permissões de acesso de seus arquivos e diretórios.

No Controle de Acesso Mandatório (MAC – *Mandatory Access Control*) as regras são implementadas seguindo as políticas e são impostas aos usuários. Dessa maneira, os usuários não têm influência sobre as permissões de acesso. Esse modelo também é conhecido como modelo militar, isto porque dadas as exigências de segurança do ambiente militar, o controle de acesso deve ser imposto de forma a minimizar as falhas de segurança. Um exemplo comum é o de servidores *proxy web* com controle de acesso a páginas restritas. Nesses servidores, uma vez estabelecidas as políticas de acesso, os usuários *web* terão de seguí-las, não tendo possibilidade de decidir sobre isto.

A figura 2.6 representa os controles DAC e MAC graficamente.

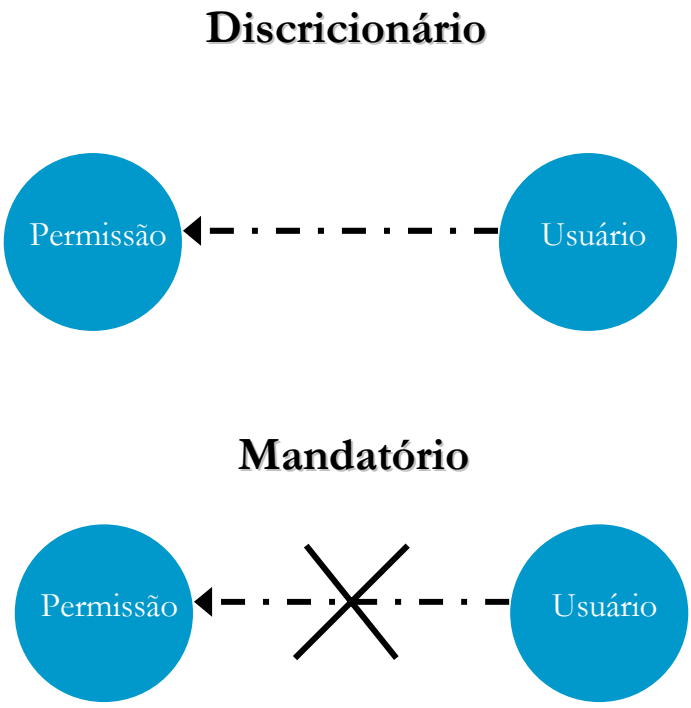

Figura 2.6 - DAC e MAC

2.2. TERMOS E DEFINIÇÕES DO RBAC

Os termos e definições abaixo e seus respectivos significados são utilizados na proposta de padrão do RBAC [9].

Component (Componente) – refere-se a um dos blocos principais do RBAC: *Core* RBAC, *Hierarchal* RBAC, *Static Separation of Duty* (SSD) *Relations* e *Dynamic Separation of Duty* (DSD) *Relations*.

Object (Objeto) – qualquer recurso do sistema sujeito a controle de acesso, tais como: arquivo, impressora, terminal, registro de base de dados, etc.

Operations (Operações) – é uma imagem executável de programa que quando invocado executa alguma função para o usuário.

Permissions (Permissões) – uma aprovação para executar uma operação em um ou mais objetos protegidos pelo RBAC.

Role (Papel) – refere-se a um cargo dentro do contexto de uma organização com algum significado associado relativo à autoridade e à responsabilidade conferidas ao usuário designado à *role*.

User (Usuário) – definido como sendo um ser humano.

2.3. MODELO DE REFERÊNCIA RBAC

Devido à diversidade de implementações do RBAC, o *National Institute of Standards and Technology* (NIST) dos Estados Unidos escreveu uma proposta de padrão para o RBAC. Nessa proposta constam o modelo de referência e a especificação funcional utilizados no RBAC.

2.3.1. Components do RBAC

O modelo de referência do RBAC é definido em função dos componentes principais do RBAC, também conhecidos como modelos de referência. A relação entre os modelos de referência é ilustrada na Figura 2.7.

O *Core* RBAC, ou $RBAC_0$, define uma coleção de componentes e características básicas do RBAC e por ser o requisito mínimo para toda implementação de RBAC fica abaixo.

O $RBAC_1$ representa o *Hierarchal* RBAC que consiste no $RBAC_0$ adicionando-se as hierarquias de papéis.

O $RBAC_2$ representa o *Constrained* RBAC que consiste no $RBAC_0$ adicionando-se as restrições, *SSD Relations* e *DSD Relation*s.

O modelo $RBAC_3$ é o modelo consolidado e inclui os modelos $RBAC_1$ e $RBAC_2$ e, por transitividade, o modelo $RBAC_0$.

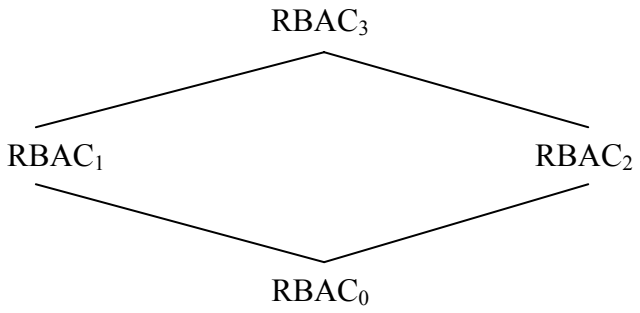

Figura 2.7 – Modelo de referência do RBAC

Cada componente do modelo é definido pelos seguintes sub-componentes:

- Uma coleção de conjuntos de elementos básicos;
- Uma coleção de relações RBAC envolvendo os conjuntos de elementos; e
- Uma coleção de funções de mapeamento que mapeia instâncias de um conjunto de elementos em instâncias de outro conjunto de elementos.

2.3.2. Core RBAC

É composto pelas entidades: *users* (usuários), *roles* (papéis), *objects* (objetos), *operations* (operações) e *permissions* (permissões); e pelas relações dadas pelas *sessions* (sessões). Este modelo é representado pela Figura 2.8.

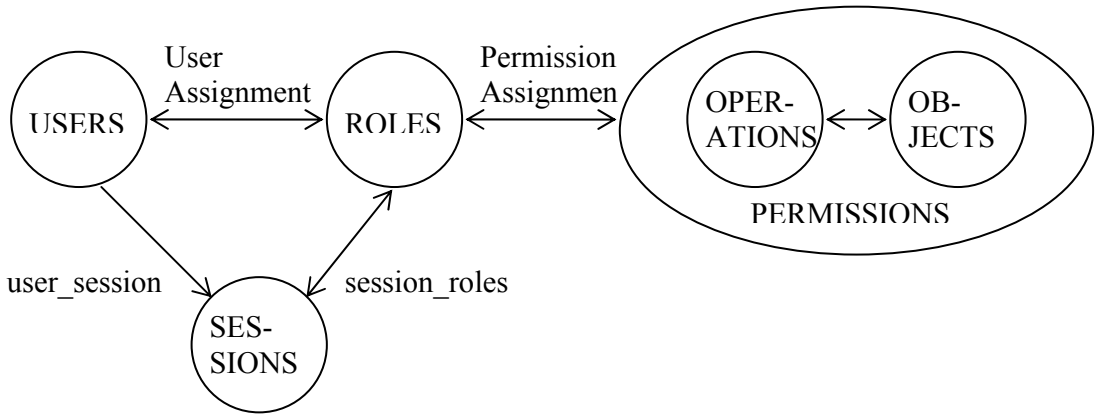

Figura 2.8 – Core RBAC

O termo usuários, apesar de ser definido como um ser humano, pode ser estendido de modo a incluir máquinas, redes, ou agentes inteligentes autônomos, tais como um robô.

As permissões conferem a seus possuidores a habilidade de executar alguma ação no sistema. Nesse sentido, podemos dizer tratar-se de permissões positivas em contraste às chamadas permissões negativas, as quais proíbem o usuário de executar determinadas tarefas. O modelo assume que as permissões negativas são, na verdade, *constraints* (restrições).

Na literatura é comum encontrar termos como autorização, direito de acesso ou privilégio para denotar permissões. Isto acontece porque a denominação utilizada para as permissões reflete o tipo de sistema a que pertencem e está intimamente ligada aos tipos de operações possíveis nesse sistema. Assim, usuários possuem autorização para efetuar *logon* em sistemas. Direitos de acesso são atribuídos a usuários para diretórios, arquivos, impressoras e outros recursos de rede. Sistemas de banco de dados relacional concedem privilégios de *select*, *update*, *delete* e *insert* nas suas tabelas e demais objetos.

O conceito central do RBAC é o das relações de papel. Na Figura 2.8 observamos que os usuários se relacionam com os papéis através da *user assignment* (atribuição de usuário), que por ser do tipo muitos-para-muitos indica que um usuário pode ter vários papéis atribuídos a ele, e, inversamente, um papel pode ser atribuído a vários usuários. Analogamente, a *permission assignment* (atribuição de permissão), que também por ser do tipo muitos-para-muitos, indica que um papel pode ter várias permissões atribuídas a ele, e, inversamente, uma permissão pode ser atribuída a vários papéis. Essa maneira de se atribuir usuários e permissões a papéis é responsável pela grande flexibilidade e granularidade das atribuições de permissão, o que não ocorre na atribuição direta de permissões a usuário, sem o uso de papéis.

O conceito de *sessions* (sessões) está relacionado com a possibilidade dos usuários ativarem e desativarem os papéis a que estão atribuídos em determinado momento. Assim, um usuário estabelece uma sessão quando ativa um subconjunto de papéis aos quais pertence. Dessa forma, as *user_sessions* (sessões de usuário) são definidas como uma relação um-para-muitos em que um usuário pode estabelecer várias sessões, mas cada sessão só pode ser estabelecida por um usuário. Por outro lado, as *session_roles* (sessões de papéis) são definidas como uma relação muitos-para-muitos em que as sessões podem ativar vários papéis e cada papel pode ser ativado por várias sessões. O uso de sessões permite a implementação do princípio do menor privilégio, pois o usuário pode ativar a cada momento apenas os papéis que necessita para efetuar as operações daquele momento.

2.3.3. Hierarchal RBAC

O modelo $RBAC_1$ é criado introduzindo-se o conceito de hierarquia de papéis. Este modelo é representado pela Figura 2.9.

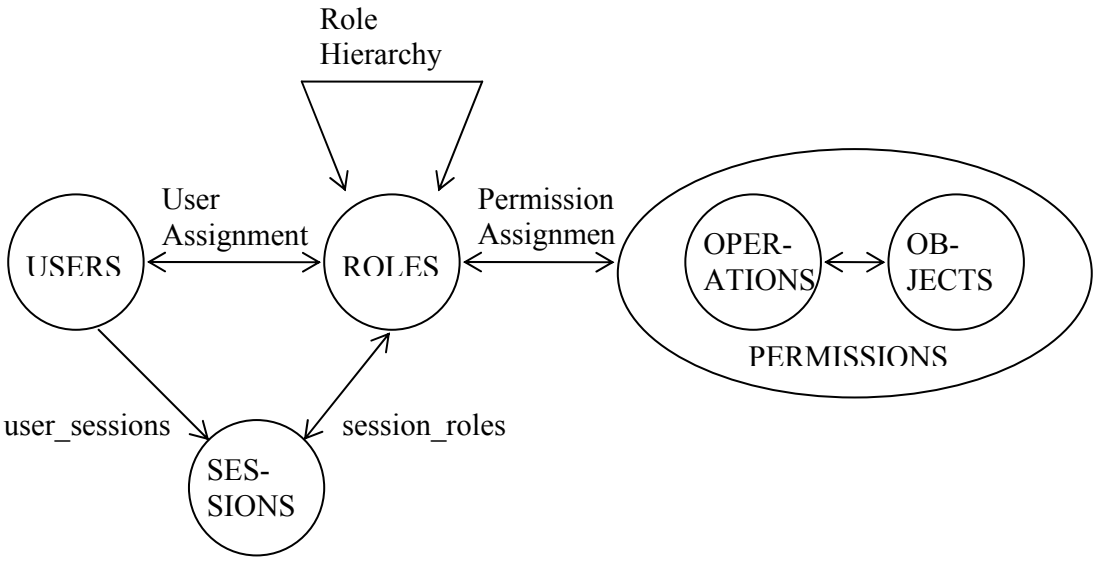

Figura 2.9 – Hierarchal RBAC

As organizações normalmente possuem uma estrutura hierárquica que determina uma cadeia de comando. Dessa maneira, para refletir essa estrutura, o modelo $RBAC_1$ é normalmente implementado nos sistemas baseados em papéis, permitindo que a política de segurança seja descrita de maneira extremamente natural.

No RBAC as hierarquias são representadas posicionando-se os papéis com mais permissões acima e os com menos permissões abaixo. Na Figura 2.10, por exemplo, observamos a hierarquia do Analista Sênior, Analista Pleno e Analista Júnior. Nesta representação temos que o Analista Pleno herda as permissões do Analista Júnior, e o Analista Sênior herda as permissões do Analista Pleno, e, conseqüentemente, as do Analista Júnior. Além das permissões herdadas, o Analista Pleno e Analista Sênior podem possuir permissões específicas atribuídas diretamente a eles. Essa forma de herdar as permissões permite que a hierarquia dos papéis utilize atribuições de permissões diferenciais, isto é, basta atribuir ao Analista Pleno aquelas permissões que ele precisa e que não estão atribuídas ao Analista Júnior. Assim, diz-se que a atribuição de permissões é feita de forma *bottom-up*. A atribuição

de usuários é feita de maneira *top-down*, isto é, os usuários aos quais é atribuído o papel de Analista Pleno é também atribuído o papel de Analista Júnior.

Figura 2.10 – Hierarquia de papéis

O padrão RBAC permite dois tipos de hierarquia: Hierarquia de Papéis Geral e Hierarquia de Papéis Limitada. A Hierarquia de Papéis Geral suporta herança múltipla, o que permite que um papel herde de mais de um papel e tenha mais de um papel como herdeiro. A Hierarquia de Papéis Limitada não suporta herança múltipla. Nesse caso, um papel pode ter mais de um papel como herdeiro, mas deve herdar apenas um papel.

2.3.4. Constrained RBAC

O modelo $RBAC_2$ implementa, por meio de restrições, o conceito de Separação de Tarefas. A Separação de Tarefas é um conceito utilizado nas organizações para dificultar a possibilidade de fraudes. Por este conceito, papéis como o de requisitante de gasto e de autorizador de gasto não devem ser atribuídos ao mesmo usuário. Isto para impedir que uma mesma pessoa possa requisitar um gasto e autorizá-lo logo em seguida. Com esse modelo, para que ocorra uma fraude na organização é preciso que haja conluio entre membros dos dois papéis para viabilizá-la. Isto faz com que aumente o nível de segurança na empresa.

O exemplo citado acima é classificado como sendo um caso de papéis mutuamente exclusivos. Nesse caso, a Separação de Tarefas foi implementada na atribuição de usuários, isto é, um usuário não pode ter os dois papéis atribuídos a ele. Outra possibilidade seria a de implementar por meio da atribuição de permissões, isto é, as permissões para requisitar gasto e autorizar gasto não poderiam ser atribuídas ao mesmo papel, o que evitaria que algum papel se tornasse perigosamente poderoso.

Há, também, Separação de Tarefas por Cardinalidade. Nesse caso, a restrição para atribuição de um papel está na quantidade, geralmente máxima, de usuários que pode ser atribuído a um papel. Um exemplo comum em empresas seria o do papel de Diretor Financeiro, que seria limitado a no máximo um usuário.

Outra restrição possível é a de papéis pré-requisito. Nessa restrição, uma atribuição de usuário ou de papel só é permitida se o usuário ou papel já tiver outra permissão específica atribuída. O exemplo mais conhecido desta restrição é a de direitos de acesso em sistemas de arquivos onde para atribuir a permissão escrita é preciso ter a permissão de leitura atribuída.

A principal classificação para as formas de separação de tarefas possui dois tipos: Separação Estática de Tarefas e Separação Dinâmica de Tarefas.

A Separação Estática de Tarefas, em inglês *Static Separation of Duty* (SSD), segundo o padrão do NIST, é feita implementando-se restrições na atribuição de usuários a papéis. Assim, um usuário não pode ser atribuído a papéis considerados mutuamente exclusivos. O termo "estático" nesta implementação deve-se ao fato de que a restrição manter-se estática ao longo do tempo. O modelo de RBAC com SSD é ilustrado na Figura 2.11.

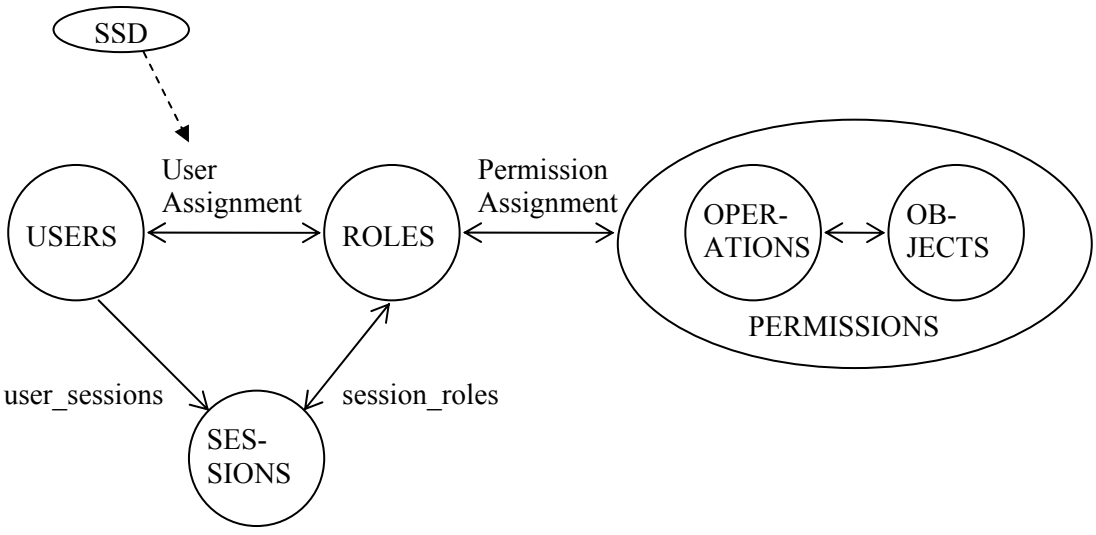

Figura 2.11 – Separação estática de papéis (SSD)

A Separação Dinâmica de Tarefas, em inglês *Dynamic Separation of Duty* (DSD), segundo o padrão do NIST, é feita implementando-se restrições nos papéis que são ativados

em uma sessão de usuário. Assim, uma sessão não pode ativar papéis considerados mutuamente exclusivos ao mesmo tempo. Isto faz com que usuários possam ter diferentes níveis de permissão em diferentes momentos. O modelo de RBAC com DSD é ilustrado na Figura 2.12.

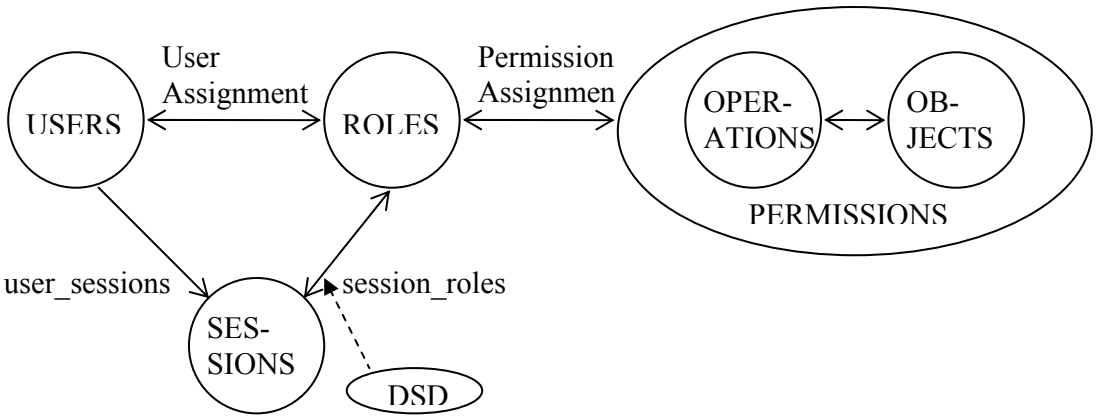

Figura 2.12 – Separação dinâmica de papéis (SSD)

2.3.5. Modelo RBAC Consolidado

O modelo consolidado é obtido combinando-se os modelos $RBAC_1$ e $RBAC_2$, implementando-se tanto hierarquia de papéis quanto restrições. Este modelo é representado pela Figura 2.13.

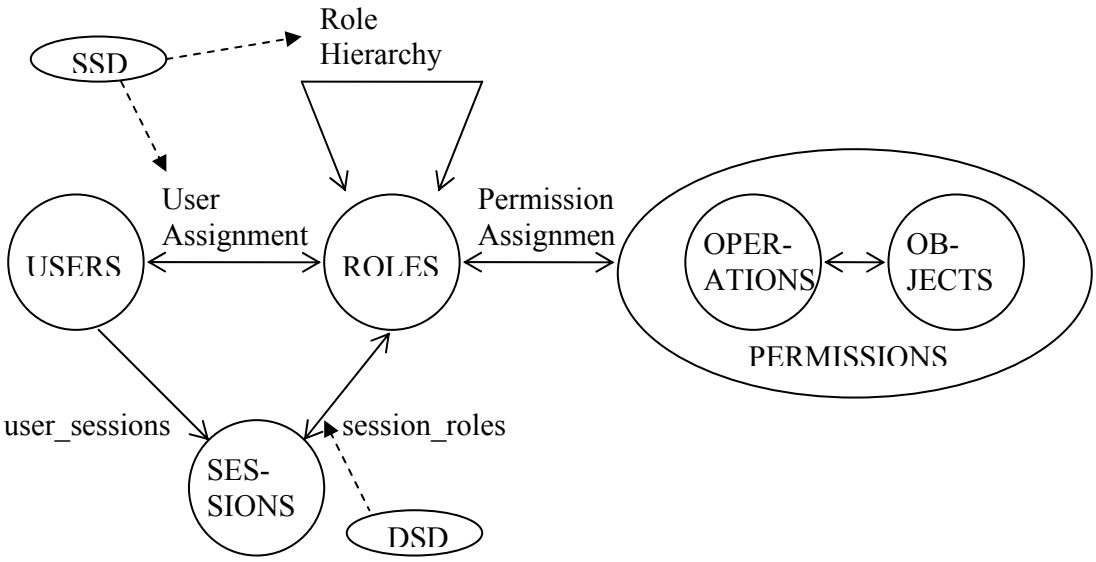

Figura 2.13 – Modelo RBAC Consolidado

Essa combinação, no entanto, não é tão simples e várias situações podem decorrer dela. A implementação de restrições em hierarquia pode se dar de forma simples como, por exemplo, restringindo o número de níveis que uma hierarquia de papéis pode ter. Nesse caso, a hierarquia de Analistas poderia ser restringida a apenas dois níveis: Analista Júnior e Analista Sênior.

Outra restrição possível seria a de dois papéis não poderem herdar de um mesmo papel, no caso de herança simples. No caso de herança múltipla pode-se especificar uma restrição de que dois ou mais papéis não tenham herdeiros em comum.

Uma situação mais complexa pode ocorrer no caso de dois papéis mutuamente exclusivos possuírem um herdeiro em comum. Essa situação pode ser aceitável em algumas implementações e em outras não. O modelo contempla ambas as possibilidades.

Outra situação diz respeito à cardinalidade. No caso de uma restrição em que um usuário pode ter apenas um papel, a atribuição de um papel que descenda de outro pode violar a restrição, pois o usuário seria membro também do papel ascendente.

2.4. ESPECIFICAÇÃO FUNCIONAL DO RBAC

A especificação funcional do RBAC descreve a semântica das várias funções que são necessárias para a criação e manutenção dos componentes do modelo do RBAC [9]. A especificação completa é descrita no Apêndice A do documento de Ferraiolo [9] que se encontra transcrito originalmente na íntegra no Anexo III.

As funções são agrupadas em três categorias: Funções Administrativas, Funções de Suporte do Sistema e Funções de Revisão.

2.4.1. Funções Administrativas

As Funções Administrativas são responsáveis pela criação e manutenção de conjuntos de elementos e relações para construção dos vários modelos do RBAC. A Tabela 2.4 discrimina estas funções para componentes do modelo. A função marcada com "*" (asterisco) na tabela é redefinida no componente.

Tabela 2.4 – Funções RBAC Administrativas

Core RBAC	Hierarchal RBAC	Static Separation of Duty (SSD) Relations	Dynamic Separation of Duty (DSD) Relations
AddUser DeleteUser AddRole DeleteRole AssignUser DeassignUser GrantPermission RevokePermission	Core RBAC + AddInheritance DeleteInheritance AddAscendant AddDescendant	Core RBAC + AssignUser* CreateSsdSet DeleteSsdSet AddSsdRoleMember DeleteSsdRoleMember SetSsdCardinality	Core RBAC + CreateDsdSet DeleteDsdSet AddDsdRoleMember DeleteDsdRoleMember SetDsdCardinality

2.4.2. Funções de Suporte do Sistema

As Funções de Suporte do Sistema são necessárias para o gerenciamento de sessões e decisões de controle de acesso durante a interação com o sistema de TI. A Tabela 2.5 discrimina essas funções para componentes do modelo. As funções marcadas com "*" (asterisco) na tabela são redefinidas no componente.

Tabela 2.5 – Funções RBAC de Suporte do Sistema

Core RBAC	Hierarchal RBAC	Static Separation of Duty (SSD) Relations	Dynamic Separation of Duty (DSD) Relations
CreateSession DeleteSession AddActiveRole DropActiveRole CheckAccess	Core RBAC + CreateSession* AddActiveRole*	Core RBAC	Core RBAC + CreateSession* AddActiveRole*

2.4.3. Funções de Revisão

As Funções de Revisão possibilitam a revisão dos resultados das ações criadas pelas funções administrativas. A Tabela 2.6 discrimina estas funções para componentes do modelo. As funções marcadas com "*" (asterisco) na tabela são redefinidas no componente.

Tabela 2.6 – Funções RBAC de Revisão

Core RBAC	Hierarchal RBAC	Static Separation of Duty (SSD) Relations	Dynamic Separation of Duty (DSD) Relations
AssignedUsers AssignedRoles RolePermissions UserPermissions SessionRoles SessionPermissions RoleOperationOnObject UserOperationOnObject	Core RBAC + AuthorizedUsers AuthorizedRoles RolePermissions* UserPermissions* RoleOperationOnObject* UserOperationOnObject*	Core RBAC + SsdRoleSets SsdRoleSetRoles SsdRoleSetCardinality	Core RBAC + DsdRoleSet DsdRoleSetRoless DsdRoleSetCardinality

3. CARACTERIZAÇÃO DO AMBIENTE

O Senado Federal é um dos órgãos que, ao lado da Câmara dos Deputados, compõem o Congresso Nacional. Trata-se de uma instituição do Poder Legislativo brasileiro, no âmbito federal, que representa os estados da federação na posição hierárquica mais alta da estrutura legislativa do país. Dentro da estrutura bicameral adotada pelo sistema brasileiro, o Senado Federal atua como a Câmara Alta, composta por 3 (três) representantes de cada estado da federação e do Distrito Federal, eleitos por voto majoritário para mandato de 8 (oito) anos.

3.1. PERFIL DA INSTITUIÇÃO SENADO FEDERAL NO CONTEXTO DA TI E DA SEGURANÇA DA INFORMAÇÃO

O Senado Federal é uma instituição civil do governo brasileiro. Suas atribuições estão bem definidas na Constituição Federal brasileira. Além da função precípua de legislar, elaborando leis, o Senado ainda exerce atividades em categorias que podem ser classificadas como fiscalizadora e parlamentar. A proposição, análise e debate de projetos de lei e de decretos legislativos, e a emissão de pareceres sobre diversas matérias fazem parte da atividade legislativa. Requerimentos de informação e a sua análise caracterizam a função fiscalizadora. Os discursos, debates e apartes em plenário e nas comissões tipificam a atividade parlamentar. O Senado Federal ainda possui atribuições exclusivas e de fundamental importância para o correto funcionamento do governo brasileiro — a Constituição Federal estabelece as atribuições exclusivas do Senado Federal no art. 52.

Naturalmente, assim como outras organizações civis, o Senado Federal faz uso intenso de recursos de informática e tecnologia para cumprir suas funções constitucionais e dar suporte às suas atividades operacionais e administrativas. Atualmente, a dependência que as instituições têm de sistemas de processamento de informações é extremamente alta e expõe a preocupação com a integridade, disponibilidade e confidencialidade dos sistemas de comunicação e das informações.

Dentro do aplicável, o Senado Federal tem grande preocupação em assegurar a confidencialidade da informação. Isso se aplica, mas não se limita, a dados pessoais; as fases do processo legislativo em que o dado ainda possui um caráter reservado antes de tornar-se informação pública (lei); informações confidenciais de terceiros quando apreciadas pelo Senado Federal (por exemplo, dados bancários e telefônicos de pessoas investigadas em

Comissões Parlamentares de Inquérito). No entanto, as atribuições do Senado Federal e a natureza de suas atividades conferem à questão da disponibilidade e integridade da informação um peso maior em relação ao aspecto da privacidade.

No contexto da utilização da TI como instrumento para cumprir suas funções, o Senado Federal possui seus próprios requisitos de segurança da informação. Numa instituição governamental civil pública como o Senado Federal, eles se baseiam mais na integridade, disponibilidade, acessibilidade da informação, muito em função da crescente exigência — pela sociedade — por transparência e correção nas atitudes dos agentes públicos. Internamente, o componente político — que é extremamente forte e influencia até algumas funções administrativas — determina uma hierarquia descentralizada onde a figura mais importante, o senador, exige condições e tratamento igualitário em relação aos demais, e que seus subordinados diretos trabalhem com o objetivo de auxiliá-los no cumprimento das suas obrigações regimentais e constitucionais, e que os indiretos observem e colaborem para manter essas premissas. Isso também é um indicativo de que a integridade e disponibilidade das informações e recursos computacionais têm um peso maior. O que o Senado Federal produz como produto final da sua atuação é de propriedade do próprio Senado Federal e da sociedade. A informação é pública e deve estar disponível de forma íntegra, incluindo a maneira como foi trabalhada ao longo do processo de criação. Os usuários dos sistemas e recursos de informática que trabalham no processo de produção da informação não detêm a sua "posse". Eles assumem uma função, um papel dentro desse processo, e é isso que regula o acesso aos recursos computacionais e à própria informação. Não é um usuário que determina, por decisão discricionária sua, quem pode ou não ter acesso a determinado recurso ou informação. Isso pode acontecer, e, de fato, acontece em alguns casos específicos, mas, normalmente, o que ocorre é que a concessão de acesso é feita de acordo com uma norma, diretriz ou política institucional corporativa que se baseia nas funções que os funcionários do Senado desempenham, nas leis que definem as atribuições do Senado, em princípios éticos, em considerações técnicas pertinentes à infra-estrutura disponível (capacidade de processamento, limitações, etc.) e em práticas de uso normalmente aceitas.

Mais especificamente, em termos de controle de acesso, sua importância no contexto da Segurança da Informação e o perfil da instituição Senado Federal, o RBAC representa uma opção interessante, especialmente quando tratamos do acesso por meios mais flexíveis, como é o caso das redes sem fio.

Uma idéia da situação atual rede do Senado, da demanda pela rede sem fio da aplicabilidade do RBAC no contexto abordado neste trabalho serão apresentados nas seções e capítulos seguintes.

3.2. SITUAÇÃO ATUAL DA REDE DO SENADO FEDERAL

Desde que foi criado com o objetivo de promover e desenvolver soluções de processamento eletrônico de dados com a aplicação da TI, o PRODASEN — Secretaria Especial de Informática do Senado Federal — procura adotar tecnologias e soluções mais adequadas, técnica e financeiramente, para atender às necessidades de seus usuários e para permitir que o Senado Federal cumpra suas funções.

Isso tem sido percebido no planejamento e destinação de parcela considerável do orçamento do órgão para manutenção e modernização da infra-estrutura de rede lógica e física, ainda que nem sempre o andamento dos projetos dessa natureza transcorra conforme o desejado em termos de prazo, em função dos processos administrativos (análises técnicas e jurídicas, pareceres, preparação de editais, aprovação e autorização de despesas, etc.) a que os órgãos públicos estão obrigados por lei a cumprir.

Atualmente a rede do Senado Federal possui, aproximadamente, 4000 (quatro mil) computadores conectados. A tecnologia predominantemente utilizada é a *Ethernet*. O núcleo da rede usa *Gigabit Ethernet* e ATM, sendo que essa última foi adotada inicialmente, mas vem sendo gradativamente substituída pela primeira. A rede atende também às residências dos senadores, localizadas na SQS 309 do Plano Piloto. Os servidores principais da rede (de banco de dados, de arquivos, de impressão, de aplicativos) encontram-se instalados e protegidos fisicamente dentro de uma sala cofre, juntamente com robôs utilizados para procedimentos de cópias de segurança, e dispositivos de armazenamento compartilhados pelos servidores.

Todos os pontos da rede são atendidos por *switches*, e aqueles da ponta da topologia se comunicam com os do núcleo por *uplinks Gigabit* ou ATM. Em termos de endereçamento, trata-se de uma rede segmentada logicamente, onde cada *switch* representa uma sub-rede configurada com VLAN´s e roteamento necessários para comunicação com demais equipamentos da rede.

Em termos de conexão externa, o Senado Federal possui ligação dedicada e permanente com a *Internet* e disponibiliza acesso remoto à sua rede por meio de tecnologia de VPN.

Especificamente em relação à rede sem fio, o Senado Federal já adota uma solução localizada dentro do plenário que permite aos senadores acompanhar a pauta das votações por meio de uma aplicação chamada Ordem do Dia Eletrônica. A tecnologia utilizada baseia-se no padrão IEEE 802.11b, operando a uma taxa de transmissão máxima teórica de 11 Mbps. Existe a previsão de iniciar, ainda no ano de 2006, a migração para o padrão IEEE 802.11g, de 54 Mbps — na prática, dependendo da distância entre dispositivo móvel e a ponto de acesso, da quantidade de dispositivos e da sobrecarga de protocolos e método de acesso ao meio, a taxa efetiva (*Throughput*) que uma aplicação pode atingir varia entre 5 a 7 Mbps no padrão IEEE 802.11b e 25 a 30 Mbps no IEEE 802.11a e 802.11g. Considerando-se que o plenário é um local onde o acesso de pessoas é controlado e a oportunidade para conduzir uma reforma que permitisse instalação de rede fixa é bastante restrita, a tecnologia sem fio foi adotada pela sua flexibilidade e pela necessidade de introduzir uma solução mais eficiente e moderna para auxílio ao processo de votação e debate de matérias. Essa flexibilidade é um dos fatores que motivam a expansão da rede sem fio para outras áreas do Senado Federal, onde certas categorias profissionais exercem atividades que seriam executadas com mais agilidade por meio do uso de dispositivos móveis com acesso aos recursos da rede.

3.3. DEMANDA POR REDE SEM FIO NO SENADO FEDERAL

A utilização de rede sem fio no Senado Federal não pretende substituir a rede tradicional fixa (com cabos), mas sim ampliar as possibilidades de acesso, oferecendo a mobilidade e flexibilidade mencionadas.

Assim, como uma tecnologia que agrega outras possibilidades, a demanda por rede sem fio no Senado surge da necessidade crescente, por parte de algumas categorias profissionais, de realizar suas tarefas e obrigações de maneira mais produtiva. A natureza de algumas atividades efetuadas dentro do Senado Federal, em função da sua composição e da forma como funciona, tem exigido um esforço maior dos profissionais, e podem ser feitas de maneira mais eficiente adotando-se instrumentos ou ferramentas mais adequadas. Por exemplo, uma das atividades dos jornalistas do Senado Federal é cobrir e divulgar o trabalho dos oitenta e um senadores e das comissões. Isso é geralmente feito no gabinete do senador —

onde o jornalista comparece para entrevistá-lo ou para conversar com seus assessores —, ou na sala das comissões — onde acontecem as audiências e os trabalhos da comissão. A possibilidade de acesso aos recursos da rede que o jornalista necessita, diretamente do local, por meio de um *notebook*, torna o desempenho da sua tarefa mais produtivo. De maneira semelhante, os funcionários que trabalham com atendimento e resolução de problemas nos computadores dos gabinetes dos senadores e nas demais áreas administrativas podem agilizar o trabalho e oferecer uma posição mais precisa sobre o andamento das ocorrências — aumentando, inclusive, a quantidade de solicitações atendidas — tendo acesso ao sistema de registro de ocorrências a partir do local, sem ter que pedir que alguém do setor onde estão realizando o atendimento pare suas atividades para que eles possam fazê-lo a partir de um computador ligado à rede fixa.

Em meados da década de 90, o Senado Federal consolidou um novo modelo de comunicação social, cujo objetivo principal era levar diretamente ao cidadão as notícias dos trabalhos desenvolvidos pelos senadores, buscando contornar os problemas do pouco espaço de divulgação dado pela imprensa e a eventual publicação de informações distorcidas sobre as atividades dos parlamentares pela mídia comercial. Esse modelo prioriza a instalação de veículos de comunicação de massa como o Jornal do Senado (maio de 1995), a TV Senado (fevereiro de 1996), a Rádio Senado (janeiro de 1997) e a Agência Senado (janeiro de 1997) [10]. Nesse contexto, a rede sem fio representa mais um instrumento de auxílio para esse tipo de atividade, tornando mais dinâmicas as tarefas da área de comunicação.

O atendimento a qualquer demanda que envolva recursos de TI deve ser avaliada sob vários aspectos, e a Segurança da Informação é um dos mais importantes. Se por um lado uma rede sem fio pode trazer benefícios para a instituição (maior flexibilidade, maior produtividade, uso mais eficiente do espaço físico, redução de custos), por outro ela é insegura, devido à própria natureza do meio de comunicação e a medidas de segurança inadequadas [11]. É neste ponto — das medidas de segurança — que o RBAC se insere como mais um mecanismo de proteção para contribuir na missão de atingir o nível de segurança requerido e planejado da instituição.

4. APLICABILIDADE E GESTÃO DO RBAC NA REDE SEM FIO DO SENADO FEDERAL

A adoção de um modelo como o RBAC na rede sem fio do Senado Federal pretende prover essa estrutura com um mecanismo mais flexível para atribuição e gerenciamento de permissões de acesso, visando, dessa forma, preservar os princípios da segurança da informação nos níveis desejados e adequados aos objetivos e à missão dessa instituição legislativa, permitindo, inclusive, a própria disponibilidade do meio de acesso sem fio aos usuários que realmente necessitam, de acordo com a função que desempenham e a política de segurança da Casa.

A adoção de qualquer modelo ou tecnologia implica na avaliação das vantagens e desvantagens, buscando uma relação custo-benefício sempre favorável. A aplicação do RBAC em um ambiente de rede sem fio apresenta diversas considerações que devem ser observadas para permitir a efetividade da sua utilização.

4.1. FUNÇÃO DO RBAC NA REDE SEM FIO DO SENADO FEDERAL

No que se refere à implantação de rede sem fio em uma escala mais abrangente dentro Senado Federal, o RBAC integraria o rol de instrumentos voltados para administração de segurança dessa estrutura, em conformidade com as práticas de segurança da instituição. As necessidades de segurança relacionadas à rede sem fio são identificadas a partir da percepção da realidade do Senado Federal na utilização da TI e procuram seguir a legislação brasileira e padrões nacionais e internacionais relacionadas à segurança da informação. O estabelecimento de uma política que englobe regras de uso geral, avaliação de riscos, procedimentos de monitoramento e auditoria, técnicas de projeto e implementação é fundamental para uma solução efetiva de segurança.

Num plano bem abrangente, a norma NBR ISO/IEC 17799 [12] é um padrão de qualidade e guia de referência para políticas de segurança da informação reconhecido mundialmente, que deve ser observado quando elas forem elaboradas.

Embora o conceito de RBAC não seja novo — a idéia básica do RBAC existe a pelo menos 30 anos —, o uso de papéis na política de administração da rede é relativamente novo [6], e somente a partir de um período mais recente ele passou a atrair atenção como método de

controle de acesso mais voltado para o desenvolvimento de aplicações [13]. No contexto deste trabalho, a adoção e implementação do RBAC não acontecem dentro da camada de aplicação, onde o desenvolvedor constrói o aplicativo inserindo comandos ou chamadas de API´s (*Application Programming Interfaces*, ou Interface de Programação de Aplicativos) que forneçam a funcionalidade do RBAC, mas é dirigida para camadas inferiores, considerando como referência o modelo de camadas OSI [14]. Controle de acesso, incluindo RBAC, compreende elementos de *hardware* e *software*. Embora o *software* que implemente as funcionalidades do RBAC possa ser considerado uma aplicação, o controle de acesso por papéis do RBAC na rede sem fio do Senado produzirá efeitos nos protocolos a partir da camada 3 (três).

Conforme mencionamos anteriormente, a adoção do RBAC pretende, também, assegurar o acesso — neste contexto definido como a habilidade de utilizar um sistema, recurso de TI, por exemplo, usar recursos de comunicação; executar programas; visualizar, alterar ou apagar dados; e semelhantes — à estrutura de rede sem fio do Senado, permitindo que ela possa suportar adequadamente todos os usuários que necessitem utilizá-la. Para isso, o controle de acesso — o meio pelo qual a habilidade de utilizar os sistemas é concedida ou restringida de alguma maneira — deve analisar a identidade do usuário, mapeando-a para um papel, e determinando, assim, se pode (e o quanto pode) ou não usar o recurso. Controle de acesso está diretamente ligado à identidade do usuário. E a identificação do usuário é a base para o processo de autorização, sendo este parte de qualquer solução de segurança projetada para defender uma rede sem fio das vulnerabilidades a que está sujeita. Este é um ponto essencial: todos os processos envolvidos na implementação da segurança devem ser cuidadosamente planejados e, posteriormente, monitorados para garantir a qualidade da solução. O RBAC representa um processo importante que, combinado com outras técnicas específicas para redes sem fio, compõem uma solução robusta para esse ambiente.

4.2. BENEFÍCIOS DO RBAC

Facilitar a administração de autorização, reforçar a aplicação da política de segurança corporativa e aprimorar a segurança e integridade dos sistemas são benefícios que potencializam a utilização do RBAC [3] [6]. Embora exista um esforço inicial considerável para a identificação e determinação dos papéis dentro da organização, após o estabelecimento das permissões de acesso há uma tendência de estabilidade, observada em um grau menor do esforço administrativo requerido para manutenção.

A simplificação do gerenciamento de permissões está relacionada à idéia de que os usuários não têm acesso discricionário aos recursos corporativos, mas são administrativamente associados a papéis, e a esses, por sua vez, são atribuídas as permissões de acesso de acordo com as funções, responsabilidades e qualificações que representam dentro da organização. Assim, os usuários são associados aos respectivos papéis, podendo redefinir a associação de um papel para outro sem modificar estrutura básica de acesso [3]. Vários fatores podem influenciar a simplificação. Quanto maior a rotatividade de pessoal, e, em conseqüência, maior o número de mudanças de papéis, mais simples é o RBAC em comparação com outros métodos de controle de acesso. Até em organizações muito dinâmicas, onde os papéis e as permissões podem mudar rapidamente, o RBAC é mais eficiente na mudança dos papéis dos usuários e na alteração das permissões dos papéis [6].

O reforço da aplicação da política de segurança baseia-se no fato do RBAC ser considerado não discricionário no sentido de que os usuários são forçados a cumprir a política do órgão, mantida de maneira centralizada por um administrador de segurança que representa a instituição.

O aprimoramento da segurança e integridade sistemas se revela na redução do impacto das violações de segurança de duas maneiras: primeiro, o RBAC pode reduzir a predisposição para ocorrências de violações de segurança; segundo, ocorrendo uma violação, o RBAC pode limitar o estrago provocado por ela [6].

4.3. CONSIDERAÇÕES SOBRE ESFORÇO E TAREFAS PARA IMPLEMENTAÇÃO DO RBAC

O esforço necessário para implementar qualquer sistema ou tecnologia voltada para segurança envolve aspectos administrativos, operacionais e financeiros que podem influenciar a decisão de adotar um modelo como o RBAC. No caso do Senado Federal, ou qualquer organização que pretenda desenvolver um projeto em contexto semelhante, alguns pontos devem ser observados com atenção. A natureza do modelo RBAC cria uma relação com as funções desempenhadas pela instituição. A situação ideal seria aquela em que as funções e processos do negócio fossem estabelecidos juntamente com uma estrutura para suportá-los e os sistemas fossem projetados com o conceito de papéis correspondendo às funções e processos definidos. Cada papel teria um conjunto de permissões associado de acordo com a posição e função dentro da instituição [6]. Entretanto, raramente esse cenário ideal ocorre, e é

necessário trabalhar e adaptar-se às condições existentes. Isso implica que, além do custo direto da aquisição do *hardware* e *software* RBAC, algumas tarefas e procedimentos específicos devem ser efetuados.

4.3.1. Definição dos Papéis (Role Engineering)

O processo de definição dos papéis é chamado de Role Engineering ou Role Definition. Dependendo do escopo da implementação e do tamanho da instituição esse processo pode levar de semanas a meses, mas é vital para o sucesso do RBAC.

Esse processo envolve a definição dos papéis que determinarão quais usuários terão acesso a quais informações e aplicações, além do relacionamento entre os papéis (hierarquia, restrições).

Uma característica desse processo é que ele pode identificar acessos informais à medida que os papéis vão sendo definidos. A transição para um sistema de controle baseado em papéis formaliza vários relacionamentos dentro da instituição [6].

No caso do Senado Federal, esse processo envolverá, também, um levantamento e uma avaliação dos atuais grupos de usuários definidos na base de autenticação da rede corporativa, e uma integração com a definição dos papéis que correspondam às funções e categorias profissionais existentes na instituição.

Algumas ferramentas de *software* foram desenvolvidas para auxiliar na definição dos papéis. O NIST, por exemplo, desenvolveu o RGP-Admin, um produto para gerenciar os relacionamentos entre papéis e permissões, e AccesMgr, uma interface gráfica para gerenciamento de listas de controle de acesso de arquivos do *Windows*.

4.3.2. Interoperabilidade e Estrutura Legada

Interoperabilidade é a capacidade de se comunicar e transferir informação entre sistemas de plataformas diferentes. Esse é um requisito importante que precisa ser observado especialmente quando a empresa ou instituição possui uma estrutura legada de sistemas e componentes distribuída e híbrida.

No caso do Senado Federal, o impacto desses requisitos não deve ser tão acentuado, uma vez que a implementação da rede sem fio encontra-se em fase inicial, e a adoção do RBAC tem um escopo bem definido de controlar acesso dentro do limite dessa rede,

oferecendo uma interface para os recursos da rede corporativa, conforme veremos mais adiante. A interoperabilidade seria requerida para uma possível e desejável integração da autenticação do usuário com alguma base de dados ou serviço de diretório já existente.

4.4. CONSIDERAÇÕES SOBRE ARQUITETURA E COMPONENTES PARA IMPLEMENTAÇÃO DO RBAC NA REDE SEM FIO DO SENADO FEDERAL

Conforme mencionamos anteriormente, a rede sem fio do plenário do Senado Federal baseia-se no padrão IEEE 802.11b, funcionando em modo infra-estrutura, ou seja, os usuários com dispositivos sem fio (*notebooks*) se conectam a um outro equipamento chamado AP – *Access Point* (Ponto de Acesso), que possui, também, uma conexão para a rede fixa tradicional, e permite a comunicação entre os usuários conectados a ele ou entre esses e os da rede fixa. O padrão IEEE 802.11b, assim como o 802.11a e 802.11g, faz parte de uma família de especificações e protocolos de comunicação para redes sem fio definida pelo comitê de padronização do IEEE (*Institute of Electrical and Electronics Engineers*). Essa família de padrões é amplamente utilizada para implementação de redes sem fio em ambientes corporativos e domésticos, também, e reside nas camadas física e de enlace do modelo de referência OSI. A implantação da rede sem fio nas outras áreas do Senado Federal continuará adotando o padrão 802.11b, que será atualizado posteriormente para o 802.11g. A figura 4.1 mostra o escopo das especificações IEEE 802.11 em referência às camadas do modelo OSI.

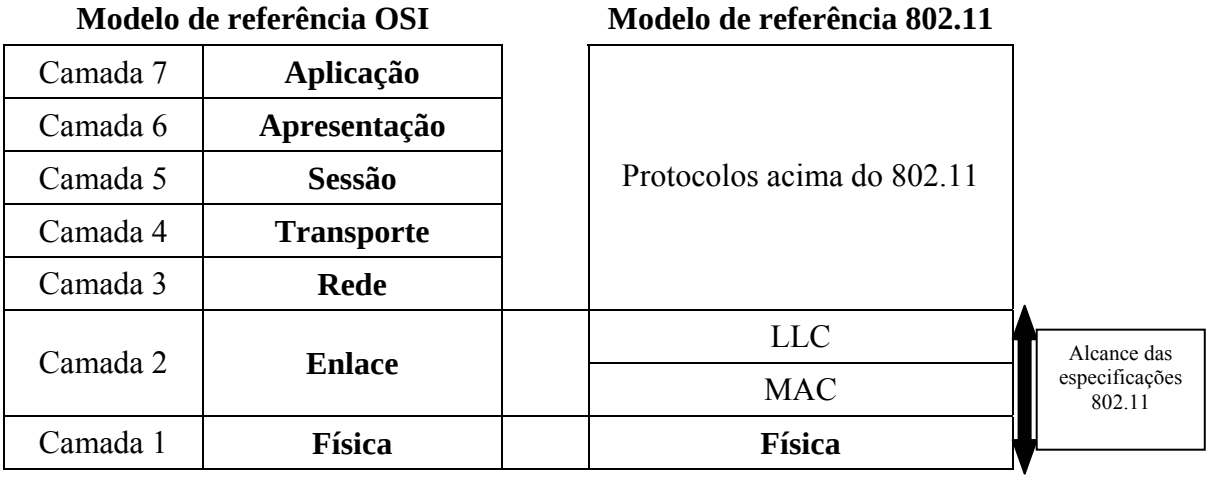

Figura 4.1 - Modelo de referência OSI x 802.11

A implementação eficiente e eficaz de qualquer sistema de controle de acesso requer planejamento. A implantação do RBAC como componente da rede sem fio do Senado Federal não foge a essa regra.

Inicialmente, é oportuno falar sobre algumas características básicas das redes sem fio no que se refere a segurança. Levando-se em conta o nível de segurança que se deseja atingir, há diversas maneiras e métodos de segurança que podem compor uma solução de rede sem fio. Para ilustrar as possibilidades, uma abordagem interessante é a que propõe a classificação das redes sem fio em três tipos [11]: *Basic* (Básica), *Hardened* (Enrijecida) e *Secured* (Fortalecida). A figura 4.2 mostra essa classificação de maneira modular.

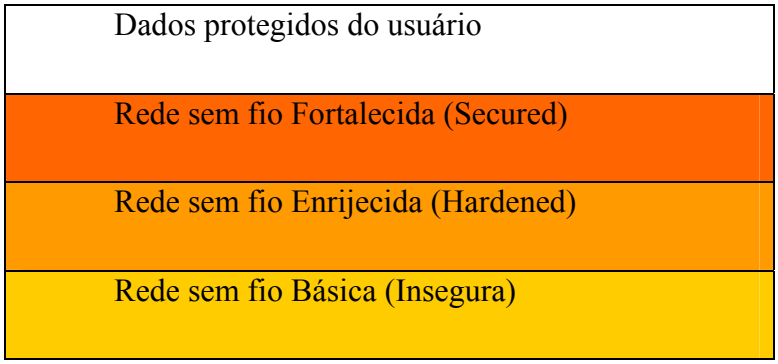

Figura 4.2 – Abordagem modular para solução de segurança

O primeiro nível (Básico – Inseguro) usa as definições e recomendações da especificação original do padrão 802.11, em sua forma mais básica, e configuração padrão dos equipamentos de acesso conforme distribuída por seus fabricantes. Embora esse nível de segurança possa ser apropriado a alguns cenários, não se aplica ao ambiente do Senado Federal.

O nível seguinte (*Hardened* – Enrijecido) emprega mecanismos de proteção mais específicos, tais como criptografia WEP e ACL´s; protocolos de segurança dedicados, tais como 802.1x para requisitar autenticação; RADIUS para gerenciamento centralizado de usuários; WPA, WPA2, TKIP, 802.11i. Esse nível de segurança é adequado ao perfil do Senado Federal.

O nível mais seguro (*Secured* – Fortalecido) é indicado para redes sem fio onde as informações são consideradas confidenciais e requer algum tipo de certificação de segurança, com proteção ponto-a-ponto por VPN ou tecnologia semelhante.

O modelo para controle de acesso RBAC se insere como um componente adicional, complementando os mecanismos do nível intermediário de segurança (*Hardened* – Enrijecido), agregando outras possibilidades de controle que são aplicadas considerando a identidade do usuário e a sua função dentro da instituição.

O emprego de uma solução de segurança contemplando o modelo RBAC envolve componentes de *hardware* e *software*, conforme já dissemos. Os componentes de *hardware* (processador, memória, disco, interfaces de rede) devem ser dimensionados para suportar a carga tanto em termos da quantidade de usuários simultâneos que estarão sujeitos ao controle quanto da complexidade que a funcionalidade do RBAC implementada pelo *software* representa. As opções de *hardware* incluem máquinas servidoras tradicionais, no sentido de que seriam adquiridas para a finalidade de executar o *software* RBAC, ou soluções integradas do tipo *appliances*, compostas do *hardware* e do *software* necessários. Os *appliances* são soluções normalmente proprietárias — os componentes têm especificações próprias que não seguem nenhum padrão, e o *software* só executa sob a combinação específica de componentes de *hardware* para o qual foi desenvolvido —, embora alguns mais modernos já utilizem partes padronizadas e sistemas operacionais comerciais ou de código aberto. Os *appliances* podem não apresentar a mesma capacidade de expansão dos servidores tradicionais porque são geralmente pré-configurados para uma tarefa específica, priorizando aspectos de desempenho e custo para o objetivo a que se destina.

O projeto dessa solução deve integrar todos esses componentes de maneira que a sua atuação conjunta cumpra os objetivos desejados, formando uma arquitetura abrangente em relação ao escopo considerado, que, neste caso, se trata de uma rede sem fio com conexão para a rede fixa corporativa do Senado Federal. Não é objetivo deste trabalho detalhar tecnicamente o projeto da solução, mas é essencial fazer certas considerações sobre a arquitetura indicada para o assunto em questão. A figura 4.3 apresenta essa arquitetura, mostrando os principais elementos relacionados à implementação de controle de acesso em um ambiente de rede sem fio.

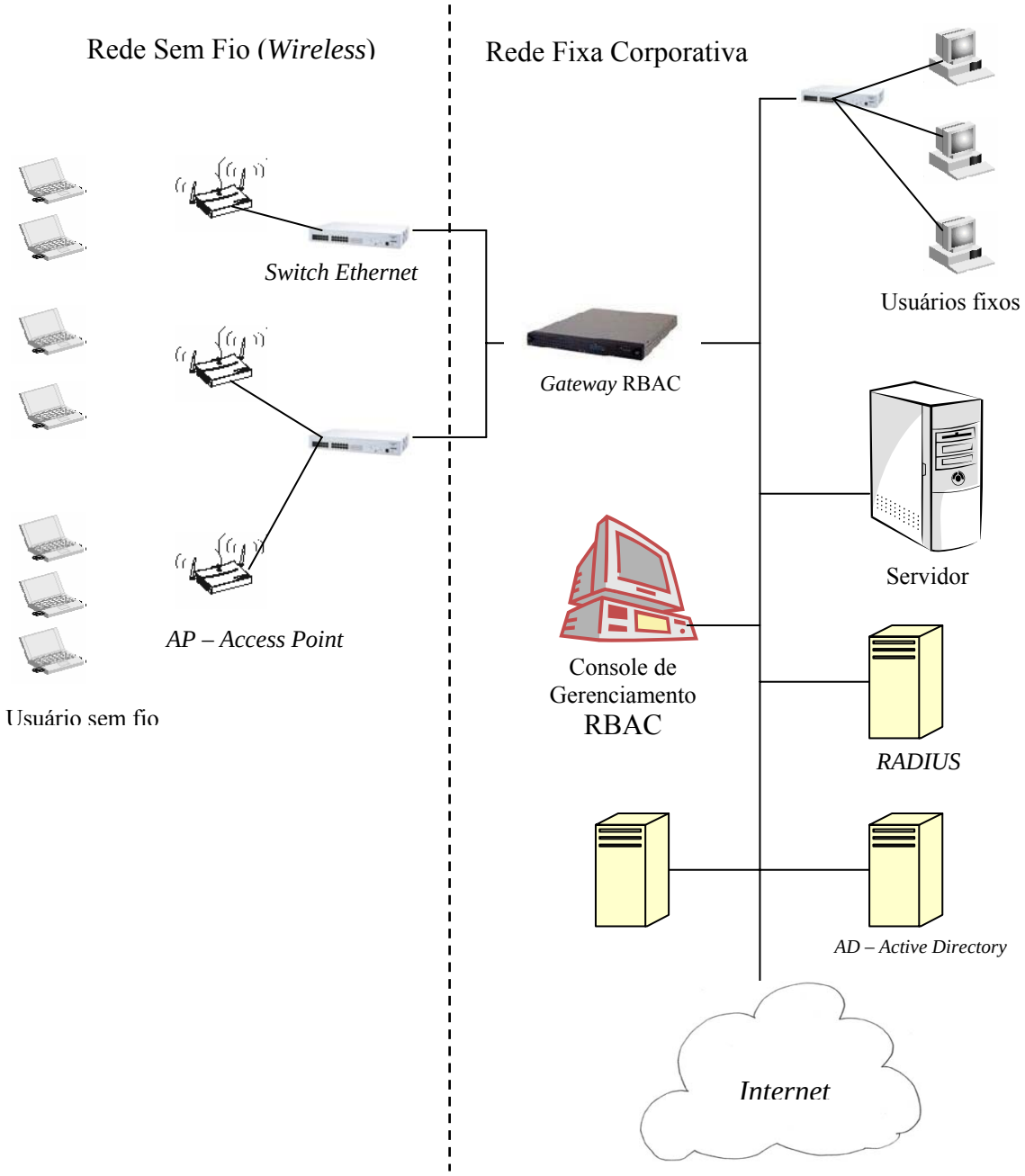

Figura 4.3 – Arquitetura Rede Sem Fio/RBAC

A idéia mostrada nessa arquitetura é que o controle de acesso RBAC é efetuado por um dispositivo dedicado, que é a interface final dos usuários da rede sem fio com a rede corporativa fixa. Todo o tráfego originado na rede sem fio ou a ela dirigido deve, necessariamente, passar por esse dispositivo, que atua como uma porta de comunicação, um *gateway*, para a rede sem fio.

Variações de definição podem existir, mas no contexto deste trabalho um *gateway* é um dispositivo que controla o fluxo de dados que entra ou sai de uma rede, transferindo-os

para outra rede ou sub-rede, efetuando operações a partir da camada 3 (três) do modelo OSI, com base em endereço de origem e/ou destino, tipos de protocolos, identidade do usuário, tipo de aplicação. Na rede sem fio do Senado Federal, o *gateway* implementaria a funcionalidade do modelo RBAC para controlar o acesso aos serviços disponíveis na rede corporativa.

O modo de operação da rede sem fio que se pretende instalar permite a sua integração à rede local corporativa da instituição. A rede sem fio passa a ser uma estrutura empresarial e para que seja funcional, apresente os níveis de segurança requeridos e possa ser gerenciada adequadamente é necessário estar bem familiarizado com a tecnologia utilizada e seus aspectos funcionais, considerando-se como referência conhecida o modelo OSI para compreender extensão da solução de segurança a ser aplicada.

Para a arquitetura apresentada, o tratamento da segurança deve ser distribuído nos níveis em que os componentes atuam, formando uma solução hierarquizada que garanta a proteção requerida em cada camada. Soluções para redes sem fio existem nas camadas 2 e 3, sendo possível combinar os recursos de cada camada para compor uma arquitetura de segurança mais robusta.

Os AP´s são dispositivos que atuam estritamente na camada 2 como pontes entre meio sem fio e a sua interface que permite conexão com a rede fixa, transferindo pacotes de dados. Nessa camada, as medidas de segurança direcionam-se à proteção da comunicação entre o dispositivo do usuário e o AP pelo meio sem fio contra interceptação e alteração indevidas, fazendo uso dos mecanismos e protocolos de criptografia e autenticação disponíveis. Dois pontos são importantes: quem pode se "associar" ao AP e, a partir daí, a comunicação propriamente dita. A questão da associação é tratada com os protocolos de autenticação e a comunicação com os de criptografia.

O *gateway* é um dispositivo que atua a partir da camada 3, podendo executar uma série de serviços, tais como filtragem de pacotes, NAT, roteamento, entre outros, sendo, no nosso contexto, o RBAC o principal deles, em torno do qual os demais se integram.

O posicionamento do *gateway* como interface de duas redes — neste caso, uma sem fio e outra fixa com cabos — põe em prática um conceito chamado Segmentação. Segmentar uma rede sem fio significa colocar os AP´s em uma rede separada da rede fixa por um equipamento de segurança (*gateway*). Há várias maneiras de promover essa separação. A mais efetiva em termos de segurança é aquela que separa os AP´s fisicamente, fazendo do

gateway o único caminho físico para a rede fixa principal. No entanto, nem sempre é possível ou viável construir e manter uma estrutura física independente para essa finalidade, caso em que o Senado Federal se enquadra. Nessa situação, uma opção é fazer uso de VLAN´s ou VPN, configurando a comunicação dos AP´s até o *gateway* por meio de um caminho virtual utilizando parte da rede fixa como meio de transporte.

A razão para se utilizar um dispositivo de segurança numa arquitetura de segmentação envolvendo redes sem fio e redes tradicionais fixas com cabos é proteger os recursos disponíveis de possíveis atividades maliciosas a partir da rede sem fio, que não possui o mesmo nível físico de segurança de uma rede fixa. O princípio aqui é o mesmo da utilização de *firewalls* para proteção de redes locais quando conectadas a redes públicas, porém aplicado num contexto localizado onde as características do meio físico e funcionamento de uma tecnologia (no caso a redes sem fio) podem comprometer a segurança de outras. O *gateway* aqui colocado não é um *firewall* completo com todas as funções desse tipo de dispositivo, embora possa incorporar alguma funcionalidade nesse sentido. O principal é que o *gateway* implementa o modelo RBAC, e qualquer outra funcionalidade está direta ou indiretamente relacionada àquela para aplicação de seus princípios.

Numa abordagem hierarquizada de segurança para redes sem fio, o *gateway* é o próximo e mais poderoso nível de segurança a partir dos mecanismos de proteção dos AP´s. Considerando que os AP´s sejam preparados para enviar todo o tráfego dos usuários para o *gateway* — já que este é o único destino possível —, ainda que o dispositivo sem fio do usuário se associe ao AP, ele estará sujeito às regras de segurança estabelecidas no *gateway* via RBAC para acesso aos recursos da rede corporativa. É nesse momento que a política de segurança é reforçada, e todos os controles de acesso são exercidos para manter o nível de segurança dentro dos requisitos estabelecidos.

O controle de acesso propriamente dito pressupõe alguma maneira de autenticar o usuário. O *gateway* deve fazer isso por meios próprios ou ter capacidade de interagir com algum serviço de autenticação externo em algum ponto da rede. Alguns serviços de autenticação muito utilizados são os baseados em RADIUS e *Windows Active Directory*, por exemplo. O uso de um serviço de autenticação já existente na rede permite aproveitar as credenciais do usuário para autenticá-lo no *gateway* e, no contexto do RBAC, mapeá-lo para o papel apropriado.

A relação entre o RBAC e um serviço de diretórios existente, destinado para autenticação de usuários, pressupõe alguma interoperabilidade entre eles. Ela acontece quando a implementação do *software* RBAC é capaz de reconhecer objetos e propriedades do serviço de diretórios, utilizando-os como critérios de mapeamento dos usuários para os papéis. Por exemplo, a participação em grupos é um dos critérios que podem ser avaliados nesse sentido. Um grupo é uma coleção ou lista nomeada de usuários. Esse conceito de agrupamento está presente na maioria dos sistemas de segurança e pode ser usado para programar alguns aspectos dos papéis definidos no modelo RBAC.

A interoperabilidade entre o *software* RBAC e o serviço de diretórios no que se refere à autenticação do usuário também permite implementar a funcionalidade de *Single Sign On* (SSO). Isso significa, basicamente, compartilhamento de informações de autenticação para evitar que o usuário tenha que apresentar a sua identificação ou credenciais mais de uma vez. É uma forma de autenticação por *software* que permite ao usuário se autenticar uma vez e ter acesso a recursos de vários sistemas dentro do ambiente. No contexto deste trabalho, representa autenticar os usuários no serviço de diretórios e, a partir daí, o RBAC mapeá-los para os seus respectivos papéis e autorizar o acesso aos recursos conforme as regras definidas para cada papel.

Por motivos de simplificação, no desenho da arquitetura da figura 4.3 não aparecem explicitamente elementos de redundância do *gateway* (*cluster*, por exemplo) e outros equipamentos de comunicação e segurança (*firewalls*, roteadores, etc.) para controle de acesso à Internet e outras redes externas. Porém, esses elementos devem fazer parte da solução projetada para evitar um ponto único de falha no dispositivo que concentra todo o tráfego da rede sem fio e proteger os acessos externos.

4.5. A QUESTÃO DA LOCALIZAÇÃO DO USUÁRIO MÓVEL

Ao se considerar a mobilidade, característica marcante das redes sem fio, no modelo RBAC, pode surgir a necessidade de ter-se permissões distintas para um mesmo usuário, dependendo da localização em que este faz o acesso à rede. Um exemplo disto na rede do Senado seria a permissão para acessar o sistema de votação. Para esse sistema, além do usuário, no caso o senador, ter que pertencer a um papel específico, ele deveria estar acessando a rede de um local específico, no caso o plenário, para poder ter acesso. Há mais de

uma maneira de se implementar o controle de acesso através do RBAC considerando-se a localização, onde descrevemos abaixo as mais citadas.

A primeira possibilidade de se implementar o controle de acesso por localização no RBAC é definindo-se papéis por regiões do espaço físico de onde a rede pode ser acessada. Neste caso, determinados papéis só podem ser ativados em determinadas localizações da rede. No nosso exemplo do sistema de votação do Senado definir-se-ia um papel de "votante", o qual se atribuiria ao senador, mas que só seria ativado quando ele estivesse nas dependências do plenário da Casa. Esta implementação é descrita por Montoro e Johnson [15].

Esta forma de considerar a localização no modelo RBAC é bastante intuitiva, porém na prática apresenta algumas dificuldades de implementação. Mesmo no exemplo, pode-se perceber que seria preciso criar pelo menos dois papéis para o Senado, um para quando estivesse no plenário e outro para quando não estivesse. Da mesma forma, muitos outros papéis poderiam ser duplicados se tivessem que atribuir permissões distintas dentro e fora do plenário. O problema se tornaria ainda mais complexo se fossem consideradas outras localizações da Casa, tais como gabinetes, restaurante, hall, etc. Neste caso, alguns papéis teriam de ser replicados para as diversas localidades, criando uma matriz de papéis por localização. Esta solução, além de ter a implementação inicial trabalhosa, seria difícil de ser mantida.

4.5.1. Spatial RBAC

Outra implementação possível, denominada de *Spatial* RBAC é descrita por Hansen e Oleshchuk [16]. Nessa implementação, as permissões são atribuídas dinamicamente aos papéis segundo a localização do usuário. Desse modo, no caso do sistema de votação, o senador seria atribuído a apenas um papel, o de senador, e a este papel seria atribuída a permissão para votar. Contudo, esta permissão só seria ativada quando o senador acessasse a rede de dentro do plenário. Este método elimina a necessidade de se definir vários papéis semelhantes para um mesmo tipo de usuário, com diferenciações feitas apenas devido a permissões específicas para determinadas localizações.

4.5.2. Influência da localização do usuário no contexto do Senado Federal

O exemplo do sistema de votação mencionado anteriormente (com possibilidade de acesso por uma rede sem fio) é hipotético e não representa uma realidade no ambiente do

Senado Federal, uma vez que a rede do sistema de votação é independente e isolada por razões de segurança. Ele foi citado para ilustrar a questão da mobilidade do usuário da rede sem fio e sua relação com as permissões de acesso.

A princípio, essa questão da localização do usuário — colocada como fator único — e a sua relação com permissões de acesso não parece ter relevância no contexto do Senado Federal, uma vez que as suas aplicações e informações não possuem qualquer caráter de restrição de acesso em função do local onde se encontra o usuário.

Podem existir algumas restrições de tempo ou período, associadas ou não com a localização, em determinados casos. Por exemplo, os consultores, assessores de senadores e os próprios senadores, na época da apresentação de emendas ao orçamento, devem ter o acesso ao sistema de emendas ao orçamento garantido, priorizado; e o acesso às aplicações de *streaming*, durante o mesmo período, pode ser impedido ou restringido para qualquer papel, visando evitar saturação do meio de comunicação. Ou o acesso às aplicações de *streaming* poderia ser liberado, durante aquele período, em áreas onde a probabilidade do uso do sistema de emendas fosse baixa ou absolutamente nenhuma. Nesse caso, o fator localização não seria único e estaria associado a um período para determinar a concessão e/ou restrição de acesso.

O controle de permissões envolvendo esse tipo de variáveis pressupõe capacidade de associação ou extensão do modelo RBAC aos fatores que se deseja considerar para determinar as condições de acesso.

4.6. CENÁRIOS PARA APLICAÇÃO DO RBAC

A adoção de um modelo de controle de acesso como o RBAC só tem utilidade se, na prática, existirem situações, cenários nos quais ele possa ser aplicado e produza os resultados esperados. No contexto do Senado Federal, a aplicação do RBAC no controle de acesso da rede sem fio já é — num nível mais abstrato e amplo — um cenário real. No entanto, existem situações mais específicas para as quais os princípios do RBAC podem ser aplicados, considerando-se a realidade dos processos de negócio do Senado Federal.

4.6.1. Acesso a recursos — máquinas servidoras, sistemas

Este é o cenário mais básico para aplicação do RBAC (e de outros métodos de controle de acesso também), onde as permissões mínimas para acesso a cada recurso

disponível na rede são associadas a cada papel, permitindo que os usuários realizem suas tarefas em conformidade com política de segurança definida na instituição. Um recurso pode ser um computador, uma impressora, um sistema de informação, um endereço de um serviço ou computador na Internet, e a atribuição de permissões por "classes" de usuários da rede sem fio permite disponibilizar níveis de acesso diferenciados, com relativa facilidade de administração.

4.6.2. Gerencia de desempenho (performance) e disponibilidade da rede

Conforme já afirmamos, um dos objetivos da utilização do modelo RBAC é garantir a disponibilidade da estrutura de acesso sem fio para os usuários cujas tarefas sejam consideradas mais importantes para o Senado Federal.

As redes sem fio utilizam um meio compartilhado (o "ar") como transporte para os sinais (ondas eletromagnéticas) que representam a informação. Dentro da área de cobertura de um AP não é possível "dividir" o ar em fatias, reservando-as para determinados grupos. Uma maneira de controlar a disponibilidade da rede sem fio é impor limites de utilização da banda da rede para os papéis definidos no modelo RBAC, conforme sua função e relevância para a instituição.

No caso de uma rede sem fio como mostrada na figura 4.3, esse tipo de controle não ocorre diretamente na parte sem fio da arquitetura, mas nas interfaces fixas do *gateway*. Indiretamente, pela limitação imposta na utilização da banda no canal fixo, o efeito obtido no canal sem fio é o desejado, evitando que o meio compartilhado fique saturado por algum usuário de perfil não relevante.

4.6.3. Acesso público ou convidado

Este cenário combina as características dos dois anteriores para compor uma condição na qual certos usuários possam ter, por meio da rede sem fio, acesso limitado a outros recursos disponíveis na rede corporativa.

No Senado Federal, esse é um cenário que pode ocorrer, embora a proposta inicial da rede sem fio seja para atendimento a usuários internos com dispositivos móveis de sua propriedade. Normalmente, instituições governamentais civis que optam por implementar acesso sem fio às suas redes o fazem para fornecer essa capacidade aos seus funcionários. Porém, a área de TI passa a ser solicitada no sentido de permitir que outros usuários

(funcionários temporários, terceirizados, funcionários de outras instituições do governo, profissionais de outras empresas realizando serviços na instituição, visitantes em geral), além dos próprios funcionários, possam utilizar a rede sem fio.

A existência de uma rede sem fio num ambiente onde usuários em potencial estejam portando equipamentos com tecnologia sem fio é um convite ao uso pessoal ou profissional desse tipo de dispositivo. No entanto, essa possibilidade deve ser muita bem controlada para evitar problemas de segurança. No RBAC, a definição de perfis para "convidados", com restrições de acesso aos recursos e limitação no uso de banda, é uma maneira de exercer um controle efetivo sobre essa classe de usuários.

Um exemplo seria o caso de jornalistas externos fazendo a cobertura de algum evento no Senado Federal. Utilizando seus *notebooks*, eles poderiam enviar notícias ou matérias sobre o evento para a redação de seus respectivos jornais pela rede sem fio, que permitiria, neste caso, somente acesso á Internet e serviços de *e-mail* (correio eletrônico), por exemplo.

5. CONCLUSÃO

O principal objetivo da implantação de métodos de controle de acesso em redes de computadores é assegurar o cumprimento de políticas de segurança e garantir a observação dos princípios da Segurança da Informação nos níveis adequados à instituição.

Este trabalho procurou apresentar as características mais relevantes do RBAC, um modelo de controle de acesso baseado em papéis, perfis ou funções, e sua aplicabilidade no controle de acesso de uma rede sem fio. O conceito do RBAC não é novo, e existem implementações em sistemas operacionais, sistemas gerenciadores de banco de dados, entre outros. Vários modelos RBAC surgiram ao longo do tempo, e, atualmente, já existe um padrão proposto pelo NIST definindo um conjunto de componentes e funcionalidades básicas, além de uma terminologia consistente.

A questão da segurança das redes é um assunto sério que deve ser abordado segundo um processo que reúne vários componentes e técnicas, atuando em diversas camadas. A inserção do RBAC nesse contexto representa uma opção de método de controle de acesso que pode atuar em vários níveis, e no caso das redes sem fio complementa mecanismos específicos desse tipo de tecnologia.

Diferentemente de outros métodos de controle de acesso, como as matrizes ou listas de controle de acesso que são usadas para controlar acesso num nível mais baixo (acesso a processos ou arquivo, por exemplo), o RBAC pode ser utilizado também em um nível mais alto, definindo operações mais abstratas, tais como permitir ou negar acesso ao endereço lógico de um recurso na rede, característica essa apropriada para aplicação em redes sem fios, onde uma série de recursos e objetos podem ser tratados nesse nível.

Dentre os diversos modelos de controle de acesso, o RBAC tem como grande vantagem o fato de mapear quais dos recursos disponíveis serão acessíveis a cada papel ou perfil, e não diretamente a cada usuário. O papel, neste caso, pode representar um cargo ou função dentro da corporação. Assim, o RBAC define de forma não discricionária quais recursos cada usuário deve ter acesso baseado nos papéis a que está associado. O modelo pode também ser estendido para considerar a localização do usuário na determinação das permissões nos cenários em que essa variável seja relevante.

Assim, a aplicação do RBAC na rede sem fio do Senado Federal agregaria vantagens ao gerenciamento dos recursos da rede, tornando-o mais simples, porém mais robusto. Mais simples devido à própria natureza do RBAC, e mais robusto por permitir o reforço das políticas de segurança necessárias para a garantia do nível de serviço exigido para a rede corporativa de uma das casas do nível hierárquico mais alto do poder legislativo brasileiro.

Em face do exposto, a conclusão é que o RBAC é um modelo de controle de acesso eficiente e pode ser aplicado ao controle de redes sem fio, facilitando a administração das permissões de acesso após a fase inicial de identificação, definição e mapeamento dos papéis na instituição.

5.1. PERSPECTIVAS FUTURAS

Existe uma grande expectativa em torno da implantação da rede sem fio do Senado Federal num contexto mais abrangente do que o atual (somente no plenário). Passada a fase de *site survey*, realizada nas áreas classificadas como prioritárias para essa demanda, há interesse na adoção de uma solução de rede sem fio que contenha os princípios descritos neste trabalho.

Uma tarefa natural que segue a conclusão deste trabalho e representa uma das primeiras atividades da fase inicial da adoção do RBAC é a definição dos papéis. Numa instituição como o Senado Federal alguns papéis são intuitivos e óbvios, tais como Senadores, Consultores Legislativos, Assessores de Gabinete, Consultores de Orçamento, Jornalistas, Jornalistas Externos, entre outros. De uma forma mais completa e abrangente, esse processo deve ser conduzido e aplicado de maneira estruturada em passos que possam quebrar a complexidade inicial de uma implementação como essa. Alguns passos seriam: um planejamento principal, onde, além deste próprio trabalho, estariam declarados um cronograma, orçamento e alguns marcos para avaliação do progresso das atividades; uma análise dos recursos da rede em termos de requisitos de nível de segurança necessários; a definição dos papéis, baseados nos cargos, funções e perfis profissionais do Senado para estabelecer as regras de acesso aos recursos da rede.

Uma das atividades futuras em relação a este trabalho também seria o estudo e avaliação detalhados de produtos disponíveis no mercado que atendam a esses requisitos. O

Anexo II deste trabalho apresenta as principais características de algumas soluções de mercado disponíveis atualmente.

Outra ação muito importante, sem a qual não haverá progresso para atingir o objetivo final, é a descrição detalhada do projeto em termos técnicos, contendo as especificações necessárias para a aquisição e/ou desenvolvimento dos produtos necessários.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] *Network Technology Planning Guide (NTPG) - Networks and Networking.* California Departament of Education. Acessado em 15/08/2006 em http://www.cde.ca.gov/ls/et/rd/ntpgchap1.asp.

[2] TANENBAUM, ANDREW S.. Redes de Computadores. Rio de Janeiro. Editora Campus. 1997.

[3] FERRAIOLO, DAVID F.; CUGINI, JANER A.; KUHN, RICHARD. *Role-Based Access Control (RBAC): Features and Motivations* – National Institute of Standards and Technology. Acessado em 19/05/2006 em http://www.cs.unibo.it/people/faculty/montreso/master/materiale/ac/rbac.pdf.

[4] STALLINGS, William. Redes e Sistemas de Comunicação de Dados: Teorias e aplicações Corporativas. Rio de Janeiro. Editora Campus. 2005.

[5] ISO/IEC 7498-4:1989(E) – *Information Processing Systems — Open Systems Interconnection — Basic Reference Model – Part 4: Management Framework*

[6] GALLAHER, MICHAEL P.; O´CONNOR, ALAN C.; KROOP, BRIAN. *The Economic Impact of Role-Based Access Control*. Research Triangle Institute. Março de 2002. Acessado em 6/6/2006 em http://www.nist.gov/director/prog-ofc/report02-1.pdf.

[7] CAMELOT, Differentiating Between Access Control Terms, 2001

[8] NIST *Role Based Access Control* – ANSI/INCITS 359 DRAFT 4/4/2003. Acessado em 6/6/2006 em http://csrc.nist.gov/rbac/rbac-std-ncits.pdf.

[9] FERRAIOLO, D. F.; SANDHU, R.; GRAVILA, S.; KUHN, D.R.; CHANDRAMOULI, R. Proposed NIST Standard for Role-Based Access Control - ACM Transactions on Information and System Security, v. 4, n. 3, agosto de 2001, p. 224-274

[10] DUARTE, MARCIA YUKIKO MATSUUCHI; História da Central de Atendimento do Senado Federal. In II Encontro Nacional da Rede Alfredo de Carvalho. Abril de 2004.

[11] METOLLI, BAS; SADLER, ASH; BAXTER, RICHARD; HUGHES, IAN; *Securing Wireless LANS*. BT Exact. 2005.

[12] ABNT NBR ISO/IEC 17799:2005 - Código de Prática para a Gestão da Segurança da Informação Teoria e Prática e ISO/IEC 27001:2005 Sistemas de Gestão da Segurança da Informação – Requisitos

[13] SANDHU, RAVI S.; COYNE, EDWARD J.; FEINSTEIN, HAL L.; YOUMAN, CHARLES E. *Role Based Access Control Models*. Acessado em 6/6/2006 em http://citeseer.ist.psu.edu/cache/papers/cs/15046/http:zSzzSzwww.list.gmu.eduzS

zjournalszSzcomputerzSzpdf_verzSzi94rbac.pdf/sandhu96rolebased.pdf

[14] ISO/IEC 7498-1:1994(E) – *Information Technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*

[15] MONTORO, T.; JOHNSON C.; *Location History in a Low-cost Context Awareness Environment*. In: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003, Australian Computer Society, Inc., v. 21, 2003, p. 153-158.

[16] HANSEN, F.; OLESHCHUK, V.; *Spatial Role Based Access Control Model for Wireless Networks*. In: Vehicular Technology Conference, 2003, v. 3, 2003, p.: 2093–2097.

ANEXO I – TRABALHOS JÁ REALIZADOS NA ÁREA

A pesquisa realizada para o desenvolvimento deste trabalho revelou algumas iniciativas semelhantes de implementação de redes sem fio com controle de acesso baseado no modelo RBAC.

A *Harvard Medical School* (HMS), localizada na cidade de Boston, no estado norte americano do Massachusetts, desenvolveu um projeto chamado HMS *Wireless Quad* e disponibiliza, desde maio de 2002, uma rede sem fio para conexão à rede principal da HMS. O documento *Wireless at 'The Med School': Past, Present and Future*, disponível em http://www.hms.harvard.edu/it/wireless/HarvardMedWLANCaseStudy.pdf, descreve um pouco da historia do projeto, onde é possível identificar a utilização do RBAC no controle de acesso, de acordo com a seguinte transcrição:

> *Com o passar do tempo, HMS-IT planeja estender as capacidades do Controle de Acesso Baseado em Papel (RBAC) construída dentro do mecanismo de autenticação do Bluesocket. Conscientemente gerenciamento classes de usuários sem fio dentro da comunidade escolar, HMS-IT pode permitir que autenticação e controle de acesso disponibilizem diferentes níveis de acesso e suporte. Martino explica: "Por exemplo, no futuro, conectando aos perfis que executamos numa base de dados Oracle, o sistema de gerenciamento sem fio poderia fazer uma chamada SQL de forma que os estudantes obtivessem um documento ou apresentação; poder – alguma outra coisa. RBAC também pode ajudar no gerenciamento da largura de banda — para que estudantes baixando arquivos MP3 não derrubem o desempenho da rede." Estender o RBAC sem fio para sistemas adicionais dentro do campus no futuro disponibilizará privilégios de 'convidado' e acesso a POP e correio eletrônico para uma larga faixa de visitantes: desde estudantes em potencial e seus pais, trabalhadores temporários, até vendedores e fornecedores ... o homem do FedEx que quer atualizar o seu inventário a partir de um porto de carregamento*"

A página principal do projeto HMS *Wireless Quad* está disponível em http://www.hms.harvard.edu/it/wireless/index.html.

Outro trabalho que pode ser citado é o de Frode Hansen and Vladimir Oleshchuk em APPLICATION OF ROLE-BASED ACCESS CONTROL IN WIRELESS HEALTHCARE INFORMATION SYSTEMS, onde a proposta é a utilização do SRBAC (*Spatial* RBAC) — uma extensão do modelo RBAC que incorpora a localização do usuário na determinação e ativação das permissões, conforme mencionado na seção 4.2.1 — em sistemas da área de saúde. Segundo os autores a idéia central baseia-se na seguinte colocação:

> *Entretanto, em um ambiente de tratamento de saúde sem fio, a permissão de um usuário para acessar objetos não se basearia somente em seu papel desempenhado na organização, mas também num contexto relevante de segurança tal como a*

> *localização. Assim, em uma configuração móvel, podemos atingir maior flexibilidade definindo a política de segurança quando as permissões são atribuídas dinamicamente a um papel limitadas pela localização na qual o usuário está situado.*

Ou seja, a permissão de um usuário para acessar objetos em um ambiente sem fio aplicado à área de saúde não se baseia unicamente no papel que ele desempenha na organização, mas também em outras variáveis relevantes tais como a localização.

No Brasil não foram encontradas referências de trabalhos acadêmicos semelhantes a este.

ANEXO II – SOLUÇÕES COMERCIAIS DISPONÍVEIS

Os benefícios potenciais do RBAC têm sido reconhecidos por alguns fabricantes e desenvolvedores de soluções de segurança para redes. Alguns produtos têm incorporado princípios do RBAC, porém o que se percebe hoje, mesmo já existindo um padrão formal proposto pelo NIST, é que as implementações foram adaptadas e baseadas em modelos anteriores não padronizados, com algumas diferenças de terminologia, e nem todas as características do modelo atual estão presentes, tornando mais difícil a tarefa de comparar as soluções disponíveis e avaliar a sua efetividade para determinado ambiente.

A empresa *Cisco Systems* adota princípios do RBAC dentro do que chamam de NAC (*Network Admission Control* – http://www.cisco.com/go/nac) — um conjunto de tecnologias e soluções que usa a infra-estrutura de rede para reforçar a compatibilidade de qualquer dispositivo com a política de segurança da organização, permitindo autenticar, autorizar, avaliar e remediar usuários e suas máquinas antes de conceder acesso à rede. Especificamente, um dos componentes do NAC, o *Clean Access Server* (CAS) e *Clean Access Manager* (CAM), é que implementa a funcionalidade RBAC para controle de acesso. O CAM é um servidor de administração com console WEB de gerenciamento para o CAS. Esse solução pode ser utilizado tanto em redes sem fio quanto em redes tradicionais com cabos e compreende outras características de segurança, opção de implementação dentro e fora da banda, ferramentas de autenticação de usuários, controle de banda e de filtragem tráfego vinculados ao RBAC. Segundo o fabricante, as principais características as solução são:

- Arquitetura baseada em padrões: HTTP, HTTPS, XML e *Java Management Extensions* (JMX);
- Autenticação de usuários: integração com servidores de autenticação existentes, incluindo Kerberos, LDAP, RADIUS e domínio Windows NT;
- Integração com concentrador VPN: integração com concentradores VPN Cisco (VPN 3000, ASA) provendo SSO (*Single Sign-on*);
- Obediência a políticas: permite configurar políticas de avaliação de vulnerabilidade e remedição nos clientes por meio de um programa agente;

- Opção de implementação L2 ou L3 (nível 2 ou 3): o CAS pode ser implementado em nível 2 (L2) de proximidade dos clientes ou nível 3 (L3), afastado por sub-redes;

- Opção de implementação dentro da banda (*in-band* – IB) ou fora da banda (*out-of-band* – OOB): o CAS pode ser colocado dentro da banda, obrigando todo tráfego dos clientes a passar por ele ou fora da banda, onde o tráfego do cliente só é analisado na fase de avaliação de vulnerabilidade e remedição, desviando após a certificação — para redes sem fio só o modo dentro da banda pode ser utilizado;

- Políticas de filtragem de tráfego: efetua filtragem de tráfego por endereço das máquinas baseado em papéis (*role*);

- Controle de gerenciamento de banda: controla banda para *downloads* e *uploads* dos usuários por papéis;

- *Roaming*: permite transferência das conexões dos usuários pelas sub-redes ligadas ao CAS;

- Alta disponibilidade: suporta configuração de alta disponibilidade com mais de um CAS para que outro servidor possa assumir as tarefas em caso de falha do servidor principal.

No que se refere ao RBAC, especificamente, a implementação da *Cisco Systems* parece não possuir as características mais avançadas do modelo, tais como hierarquias de papéis, restrições e separação de tarefas. O mapeamento de um usuário para um papel pode ser feito dinamicamente baseado no endereço físico (MAC), endereço da sub-rede/IP da máquina ou informação de *login* (atributos do servidor de autenticação: nome de usuário, VLAN, etc.). O usuário é mapeado para um único papel de acordo com uma prioridade (endereço MAC, IP, atributos), o que significa que se o endereço MAC se associa ao papel A e o nome do usuário se associa ao papel B, o papel A é o utilizado.

Outra empresa chamada *Bluesocket Inc*., que fabrica *controllers* (controladores - http://www.bluesocket.com/products/controllerfamily.html) para redes sem fio, também incorpora princípios RBAC em alguns de seus produtos. O *BlueSecure* 5.1 é um pacote que

inclui *controller*, gerenciamento e AP´s onde o primeiro implementa funções RBAC para controle de acesso. Segundo o fabricante, muitas características únicas do seu produto derivam da sua capacidade em prover controle de acesso baseado em papéis (RBAC). Os papéis podem ser mantidos localmente ou serem derivados de outras bases centrais de autenticação (domínios Windows NT, *Active Directory*, LDAP, RADIUS). O controlador também permite controle de banda e filtragem de tráfego por papéis.

A implementação RBAC da *BlueSocket Inc*. também parece não suportar as características mais avançadas do modelo. Outras funcionalidades disponíveis são:

- Interoperabilidade com sistemas abertos;
- Criptografia forte: suporte a IPSEC;
- Monitoramento dos clientes e detecção de intrusão: monitoramento em tempo real do cliente, integrando o *Check Point's Integrity Clientless Security* para proteção contra vírus, *worms*, *spyware*, *malware*;
- Segurança e QoS para VoIP.

ANEXO III – ESPECIFICAÇÃO FUNCIONAL DO RBAC

Appendix A: RBAC REQUIREMENTS SPECIFICATION

The RBAC Requirements Specification specifies administrative operations for the creation and maintenance of RBAC element sets and relations; administrative review functions for performing administrative queries; and system functions for creating and managing RBAC attributes on user sessions and making access control decisions. Functions are defined with sufficient precision to meet the needs of conformance testing and assurance, while providing developers with design flexibility and the ability to incorporate additional features to meet the needs of users.

The notation used in the formal specification of the RBAC requirements is basically a subset of the Z notation. The only major change is the representation of a schema as follows:
Schema-Name (Declaration) $\lhd$ Predicate; …; Predicate $\rhd$.

Most abstract data types and functions used in the formal specification are defined in Section 3, RBAC Reference Model. New abstract data types and functions are introduced as needed. *NAME* is an abstract data type whose elements represent identifiers of entities that may or may not be included in the RBAC system (roles, users, sessions, etc.).

A.1 Requirements for Core RBAC

A.1-1 Administrative Commands for Core RBAC

AddUser
This command creates a new RBAC user. The command is valid only if the new user is not already a member of the *USERS* data set. The *USER* data set is updated. The new user does not own any session at the time of its creation. The following schema formally describes the command AddUser:

$$AddUser(user: NAME) \lhd$$
$$user \notin USERS$$
$$USERS' = USERS \cup \{user\}$$
$$user\_sessions' = user\_sessions \cup \{user \mapsto \varnothing\} \rhd$$

DeleteUser
This command deletes an existing user from the RBAC database. The command is valid if and only if the user to be deleted is a member of the *USERS* data set. The *USERS* and *UA* data sets and the *assigned_users* function are updated. It is an implementation decision how to proceed with the sessions owned by the user to be deleted. The RBAC system could wait for such a session to terminate normally, or it could force its termination. Our presentation illustrates the case when those sessions are forcefully terminated. The following schema formally describes the command DeleteUser:

$$DeleteUser(user: NAME) \lhd$$
$$user \in USERS$$
$$[\forall s \in SESSIONS \bullet s \in user\_sessions(user) \Rightarrow DeleteSession(s)]$$
$$UA' = UA \setminus \{r: ROLES \bullet user \mapsto r\}$$
$$assigned\_users' = \{r: ROLES \bullet r \mapsto (assigned\_users(r) \setminus \{user\})\}$$
$$USERS' = USERS \setminus \{user\} \rhd$$

AddRole

This command creates a new role. The command is valid if and only if the new role is not already a member of the *ROLES* data set. The ROLES data set and the functions *assigned_users* and *assigned_permissions* are updated. Initially, no user or permission is assigned to the new role. The following schema formally describes the command AddRole:

$$AddRole(role: NAME) \lhd$$
$$role \notin ROLES$$
$$ROLES' = ROLES \cup \{role\}$$
$$assigned\_users' = assigned\_users \cup \{role \mapsto \varnothing\}$$
$$assigned\_permissions' = assigned\_permissions \cup \{role \mapsto \varnothing\} \rhd$$

DeleteRole

This command deletes an existing role from the RBAC database. The command is valid if and only if the role to be deleted is a member of the *ROLES* data set. It is an implementation decision how to proceed with the sessions in which the role to be deleted is active. The RBAC system could wait for such a session to terminate normally, it could force the termination of that session, or it could delete the role from that session while allowing the session to continue. Our presentation illustrates the case when those sessions are forcefully terminated.

$$DeleteRole(role: NAME) \lhd$$
$$role \in ROLES$$
$$[\forall s \in SESSIONS \bullet role \in session\_roles(s) \Rightarrow DeleteSession(s)]$$
$$UA' = UA \setminus \{u: USERS \bullet u \mapsto role\}$$
$$assigned\_users' = assigned\_users \setminus \{role \mapsto asigned\_users(role)\}$$
$$PA' = PA \setminus \{op: OPS, obj: OBJS \bullet (op, obj) \mapsto role\}$$
$$assigned\_permissions' = assigned\_permissions \setminus \{role \mapsto assigned\_permissions(role)\}$$
$$ROLES' = ROLES \setminus \{role\} \rhd$$

AssignUser

This command assigns a user to a role. The command is valid if and only if the user is a member of the *USERS* data dset, the role is a member of the *ROLES* data set, and the user is not already assigned to the role. The data set UA and the function *assigned_users* are updated to reflect the assignment. The following schema formally describes the command:

$$AssignUser(user, role: NAME) \lhd$$
$$user \in USERS; role \in ROLES; (user \mapsto role) \notin UA$$
$$UA' = UA \cup \{user \mapsto role\}$$
$$assigned\_users' = assigned\_users \setminus \{role \mapsto assigned\_users(role)\} \cup$$
$$\{role \mapsto (assigned\_users(role) \cup \{user\})\} \rhd$$

DeassignUser
This command deletes the assignment of the user *user* to the role *role*. The command is valid if and only if the user is a member of the *USERS* data set, the role is a member of the *ROLES* data set, and the user is assigned to the role.
It is an implementation decision how to proceed with the sessions in which the session user is *user* and one of his/her active roles is *role*. The RBAC system could wait for such a session to terminate normally, or could force its termination, or could inactivate the role. Our presentation illustrates the case when those sessions are forcefully terminated. The following schema formally describes the command DeassignUser:

$$DeassignUser(user, role: NAME) \lhd$$
$$user \in USERS; role \in ROLES; (user \mapsto role) \in UA$$
$$[\forall s: SESSIONS \bullet s \in user\_sessions(user) \wedge role \in session\_roles(s) \Rightarrow DeleteSession(s)]$$
$$UA' = UA \setminus \{user \mapsto role\}$$
$$assigned\_users' = assigned\_users \setminus \{role \mapsto asigned\_users(role)\} \cup$$
$$\{role \mapsto (asigned\_users(role) \setminus \{user\})\} \rhd$$

GrantPermission
This command grants a role the permission to perform an operation on an object to a role. The command may be implemented as granting permissions to a group corresponding to that role, i.e., setting the access control list of the object involved.
The command is valid if and only if the pair (operation, object) represents a permission, and the role is a member of the *ROLES* data set. The following schema formally defines the command:

$$GrantPermission(object, operation, role: NAME) \lhd$$
$$(operation, object) \in PERMS; role \in ROLES$$
$$PA' = PA \cup \{(operation, object) \mapsto role\}$$
$$assigned\_permissions' = assigned\_permissions \setminus \{role \mapsto assigned\_permissions(roles)\} \cup$$
$$\{role \mapsto (assigned\_permissions(role) \cup \{(operation, object)\})\} \rhd$$

RevokePermission
This command revokes the permission to perform an operation on an object from the set of permissions assigned to a role. The command may be implemented as revoking permissions from a group corresponding to that role, i.e., setting the access control list of the object involved.
The command is valid if and only if the pair (operation, object) represents a permission, the role is a member of the *ROLES* data set, and the permission is assigned to that role. The following schema formally describes the command:

$$RevokePermission(operation, object, role: NAME) \lhd$$
$$(operation, object) \in PERMS; role \in ROLES; ((operation, object) \mapsto role) \in PA$$
$$PA' = PA \setminus \{(operation, object) \mapsto role\}$$
$$assigned\_permissions' = assigned\_permissions \setminus \{role \mapsto assigned\_permissions(role)\} \cup$$
$$\{role \mapsto (assigned\_permissions(role) \setminus \{(operation, object)\})\} \rhd$$

A.1-2 System Functions for Core RBAC

CreateSession(*user*, *session*)
This function creates a new session with a given user as owner and an active role set. The function is valid if and only if:
- the user is a member of the *USERS* data set, and

- the active role set is a subset of the roles assigned to that user. In a RBAC implementation, the session's active roles might actually be the groups that represent those roles.

The following schema formally describes the function. The *session* parameter, which represents the session identifier, is actually generated by the underlying system.

$$
\begin{aligned}
&CreateSession(user: NAME; ars: 2^{NAMES}; session: NAME) \triangleleft \\
&\quad user \in USERS; ars \subseteq \{r: ROLES \mid (user \mapsto r) \in UA\}; session \notin SESSIONS \\
&\quad SESSIONS' = SESSIONS \cup \{session\} \\
&\quad user\_sessions' = user\_sessions \setminus \{user \mapsto user\_sessions(user)\} \cup \\
&\qquad \{user \mapsto (user\_sessions(user) \cup \{session\})\} \\
&\quad session\_roles' = session\_roles \cup \{session \mapsto ars\} \triangleright
\end{aligned}
$$

DeleteSession(*user*, *session*)

This function deletes a given session with a given owner user. The function is valid if and only if the session identifier is a member of the *SESSIONS* data set, the user is a member of the *USERS* data set, and the session is owned by the given user. The following schema formally describes the function:

$$
\begin{aligned}
&DeleteSession(user, session: NAME) \triangleleft \\
&\quad user \in USERS; session \in SESSIONS; session \in user\_sessions(user) \\
&\quad user\_sessions' = user\_sessions \setminus \{user \mapsto user\_sessions(user)\} \cup \\
&\qquad \{user \mapsto (user\_sessions(user) \setminus \{session\})\} \\
&\quad session\_roles' = session\_roles \setminus \{session \mapsto session\_roles(session)\} \\
&\quad SESSIONS' = SESSIONS \setminus \{session\} \triangleright
\end{aligned}
$$

AddActiveRole

This function adds a role as an active role of a session whose owner is a given user. The function is valid if and only if:

- the user is a member of the *USERS* data set, and
- the role is a member of the *ROLES* data set, and
- the session identifier is a member of the *SESSIONS* data set, and
- the role is assigned to the user, and
- the session is owned by that user.

In an implementation, the new active role might be a group that corresponds to that role. The following schema formally describes the function:

$$
\begin{aligned}
&AddActiveRole(user, session, role: NAME) \triangleleft \\
&\quad user \in USERS; session \in SESSIONS; role \in ROLES; session \in user\_sessions(user) \\
&\quad (user \mapsto role) \in UA; role \notin session\_roles(session) \\
&\quad session\_roles' = session\_roles \setminus \{session \mapsto session\_roles(session)\} \cup \\
&\qquad \{session \mapsto (session\_roles(session) \cup \{role\})\} \triangleright
\end{aligned}
$$

DropActiveRole

This function deletes a role from the active role set of a session owned by a given user. The function is valid if and only if the user is a member of the *USERS* data set, the session identifier is a member of the *SESSIONS* data set, the session is owned by the user, and the role is an active role of that session. The following schema formally describes this function:

$$DropActiveRole(user, session, role: NAME) \lhd$$
$$user \in USERS; role \in ROLES; session \in SESSIONS$$
$$session \in user\_sessions(user); role \in session\_roles(session)$$
$$session\_roles' = session\_roles \setminus \{session \mapsto session\_roles(session)\} \cup$$
$$\{session \mapsto (session\_roles(session) \setminus \{role\})\} \rhd$$

CheckAccess
This function returns a Boolean value meaning whether the subject of a given session is allowed or not to perform a given operation on a given object. The function is valid if and only if the session identifier is a member of the *SESSIONS* data set, the object is a member of the *OBJS* data set, and the operation is a member of the *OPS* data set. The session's subject has the permission to perform the operation on that object if and only if that permission is assigned to (at least) one of the session's active roles. An implementation might use the groups that correspond to the subject's active roles and their permissions as registered in the object's access control list. The following schema formally describes the function:

$$CheckAccess(session, operation, object: NAME; out\ result: BOOLEAN) \lhd$$
$$session \in SESSIONS; operation \in OPS; object \in OBJS$$
$$result = (\exists r: ROLES \bullet r \in session\_roles(session) \wedge ((operation, object) \mapsto r) \in PA) \rhd$$

A.1-3 Review Functions for Core RBAC

AssignedUsers
This function returns the set of users assigned to a given role. The function is valid if and only if the role is a member of the *ROLES* data set. The following schema formally describes the function:

$$AssignedUsers(role: NAME; out\ result: 2^{USERS}) \lhd$$
$$role \in ROLES$$
$$result = \{u: USERS \mid (u \mapsto role) \in UA\} \rhd$$

AssignedRoles
This function returns the set of roles assigned to a given user. The function is valid if and only if the user is a member of the *USERS* data set. The following schema formally describes the function:

$$AssignedRoles(user: NAME; result: 2^{ROLES}) \lhd$$
$$user \in USERS$$
$$result = \{r: ROLES \mid (user \mapsto r) \in UA\} \rhd$$

A.1-4 Advanced Review Functions for Core RBAC

RolePermissions
This function returns the set of permissions (*op*, *obj*) granted to a given role. The function is valid if and only if the role is a member of the *ROLES* data set. The following schema formally describes the function:

$$RolePermissions(role: NAME; result: 2^{PERMS}) \lhd$$
$$role \in ROLES$$
$$result = \{op: OPS; obj: OBJS \mid ((op, obj) \mapsto role) \in PA\} \rhd$$

UserPermissions
This function returns the permissions a given user gets through his/her assigned roles. The function is valid if and only if the user is a member of the *USERS* data set. The following schema formally describes this function:

$$UserPermissions(user: NAME; result: 2^{PERMS}) \lhd$$
$$user \in USERS$$
$$result = \{r: ROLES; op: OPS; obj: OBJS | (user \mapsto r) \in UA \wedge ((op, obj) \mapsto r) \in PA \bullet (op, obj)\} \rhd$$

SessionRoles
This function returns the active roles associated with a session. The function is valid if and only if the session identifier is a member of the *SESSIONS* data set. The following schema formally describes this function:

$$SessionRoles(session: NAME; out\ result: 2^{ROLES}) \lhd$$
$$session \in SESSIONS$$
$$result = session\_roles(session) \rhd$$

SessionPermissions
This function returns the permissions of the session *session*, i.e., the permissions assigned to its active roles. The function is valid if and only if the session identifier is a member of the *SESSIONS* data set. The following schema formally describes this function:

$$SessionPermissions(session: NAME; out\ result: 2^{PERMS}) \lhd$$
$$session \in SESSIONS$$
$$result = \{r: ROLES; op \in OPS; obj \in OBJS | r \in session\_roles(session) \wedge ((op, obj) \mapsto r) \in PA \bullet (op, obj)\} \rhd$$

A.2 Requirements for Hierarchical RBAC

A.2a General Role Hierarchies

A.2a-1 Administrative Commands for General Role Hierarchies

All functions of section A.1-1 remain valid. In addition, this section defines the following new, specific functions:

AddInheritance
This commands establishes a new immediate inheritance relationship *r_asc* νν *r_desc* between existing roles *r_asc*, *r_desc*. The command is valid if and only if *r_asc* and *r_desc* are members of the *ROLES* data set, *r_asc* is not an immediate ascendant of *r_desc*, and *r_desc* does not properly inherit *r_asc* (in order to avoid cycle creation). The following schema uses the notations:

μ ==/
>> == νν

to formally describes the command:

$$AddInheritance(r\_asc, r\_desc: NAME) \lhd$$
$$r\_asc \in ROLES; r\_desc \in ROLES; \neg(r\_asc >> r\_desc); \neg(r\_desc \geq r\_asc)$$
$$\geq' = \geq \cup \{r, q: ROLES | r \geq r\_asc \wedge r\_desc \geq q \bullet r \mapsto q\} \rhd$$

DeleteInheritance
This command deletes an existing immediate inheritance relationship *r_asc* νν *r_desc*. The command is valid if

and only if the roles *r_asc* and *r_desc* are members of the *ROLES* data set, and *r_asc* is an immediate ascendant of *r_desc*. The new inheritance relation is computed as the reflexive-transitive closure of the immediate inheritance relation resulted after deleting the relationship *r_asc* ∨∨ *r_desc*. The following schema formally describes this command:

$$DeleteInheritance(r\_asc, r\_desc: NAME) \lhd$$
$$r\_asc \in ROLES; r\_desc \in ROLES; r\_asc >> r\_desc$$
$$\geq' = (>> \setminus \{r\_asc \mapsto r\_desc\})^* \rhd$$

AddAscendant

This commands creates a new role *r_asc*, and inserts it in the role hierarchy as an immediate ascendant of the existing role *r_desc*. The command is valid if and only if *r_asc* is not a member of the *ROLES* data set, and *r_desc* is a member of the *ROLES* data set. Note that the validity conditions are verified in the schemas *AddRole* and *AddInheritance*, referred to by *AddAscendant*.

$$AddAscendant(r\_asc, r\_desc: NAME) \lhd$$
$$AddRole(r\_asc)$$
$$AddInheritance(r\_asc, r\_desc) \rhd$$

AddDescendant

This commands creates a new role *r_desc*, and inserts it in the role hierarchy as an immediate descendant of the existing role *r_asc*. The command is valid if and only if *r_desc* is not a member of the *ROLES* data set, and *r_asc* is a member of the *ROLES* data set. Note that the validity conditions are verified in the schemas *AddRole* and *AddInheritance*, referred to by *AddDescendant*.

$$AddDescendant(r\_asc, r\_desc: NAME) \lhd$$
$$AddRole(r\_desc)$$
$$AddInheritance(r\_asc, r\_desc) \rhd$$

A.2a-2 System Functions for General Role Hierarchies

This section redefines the functions CreateSession and AddActiveRole of section A.1-2. The other functions of section A.1-2 remain valid.

CreateSession(*user*, *session*)

This function creates a new session with a given user as owner, and a given set of active roles. The function is valid if and only if:

- the user is a member of the *USERS* data set, and
- the active role set is a subset of the roles authorized for that user. Note that if a role is active for a session, its descendants or ascendants are not necessarily active for that session. In a RBAC implementation, the session's active roles might actually be the groups that represent those roles.

The following schema formally describes the function. The parameter *session*, which identifies the session, is actually generated by the underlying system.

$$CreateSession(user: NAME; ars: 2^{NAME}; session: NAME) \lhd$$
$$user \in USERS; ars \subseteq \{r, q: ROLES \mid (user \mapsto q) \in UA \wedge q \geq r \bullet r\}; session \notin SESSIONS$$
$$SESSIONS' = SESSIONS \cup \{session\}$$
$$user\_sessions' = user\_sessions \setminus \{user \mapsto user\_sessions(user)\} \cup$$
$$\{user \mapsto (user\_sessions(user) \cup \{session\})\}$$
$$session\_roles' = session\_roles \cup \{session \mapsto ars\} \rhd$$

AddActiveRole
This function adds a role as an active role of a session whose owner is a given user. The function is valid if and only if:
- the user is a member of the *USERS* data set, and
- the role is a member of the *ROLES* data set, and
- the session identifier is a member of the *SESSIONS* data set, and
- the user is authorized to that role, and
- the session is owned by that user.

The following schema formally describes the function:

$$AddActiveRole(user, session, role: NAME) \lhd$$
$$user \in USERS; session \in SESSIONS; role \in ROLES; session \in user\_sessions(user)$$
$$user \in authorized\_users(role); role \notin session\_roles(session)$$
$$session\_roles' = session\_roles \setminus \{session \mapsto session\_roles(session)\} \cup$$
$$\{session \mapsto (session\_roles(session) \cup \{role\})\} \rhd$$

A.2a-3 Review Functions for General Role Hierarchies

All functions of section A.1-3 remain valid. In addition, this section defines the following review functions:

AuthorizedUsers
This function returns the set of users authorized to a given role, i.e., the users that are assigned to a role that inherits the given role. The function is valid if and only if the given role is a member of the *ROLES* data set. The following schema formally describes the function:

$$AuthorizedUsers(role: NAME; out\ result: 2^{USERS}) \lhd$$
$$role \in ROLES$$
$$result = authorized\_users(role) \rhd$$

AuthorizedRoles
This function returns the set of roles authorized for a given user. The function is valid if and only if the user is a member of the *USERS* data set. The following schema formally describes the function:

$$AuthorizedRoles(user: NAME; result: 2^{ROLES}) \lhd$$
$$user \in USERS$$
$$result = \{r, q: ROLES \mid (user \mapsto q) \in UA \wedge q \geq r\} \rhd$$

A.2a-4 Advanced Review Functions for General Role Hierarchies

This section redefines the functions RolePermissions and UserPermissions of section A.1-4. All other functions of section A.1-4 remain valid.

RolePermissions
This function returns the set of all permissions (*op*, *obj*), granted to or inherited by a given role. The function is valid if and only if the role is a member of the *ROLES* data set. The following schema formally describes the function:

$$AllPermissions(role: NAME; result: 2^{PERMS}) \lhd$$
$$role \in ROLES$$
$$result = \{q: ROLES; op: OPS; obj: OBJS \mid (role \geq q) \wedge ((op, obj) \mapsto role) \in PA \bullet (op, obj)\} \rhd$$

UserPermissions

This function returns the set of permissions a given user gets through his/her authorized roles. The function is valid if and only if the user is a member of the *USERS* data set. The following schema formally describes this function:

$$UserPermissions(user: NAME; result: 2^{PERMS}) \lhd$$
$$user \in USERS$$
$$result = \{r, q: ROLES; op: OPS; obj: OBJS \mid (user \mapsto q) \in UA \wedge (q \geq r) \wedge ((op, obj) \mapsto r) \in PA \bullet (op, obj)\} \rhd$$

A.2b Limited Role Hierarchies

A.2b-1 Administrative Commands for Limited Role Hierarchies

This section redefines the function AddInheritance of section A.2a-1. All other functions of section A.2a-1 remain valid.

AddInheritance

This commands establishes a new immediate inheritance relationship *r_asc* νν *r_desc* between existing roles *r_asc*, *r_desc*. The command is valid if and only if *r_asc* and *r_desc* are members of the *ROLES* data set, *r_asc* has no descendants, and *r_desc* does not properly inherit *r_asc* (in order to avoid cycle creation). The following schema uses the notations:

$$\mu == /$$
$$>> == \nu\nu$$

to formally describes the command:

$$AddInheritance(r\_asc, r\_desc: NAME) \lhd$$
$$r\_asc \in ROLES; r\_desc \in ROLES; \forall r \in ROLES \bullet \neg(r\_asc >> r); \neg(r\_desc \geq r\_asc)$$
$$\geq' = \geq \cup \{r, q: ROLES \mid r \geq r\_asc \wedge r\_desc \geq q \bullet r \mapsto q\} \rhd$$

A.2b-2 System Functions for Limited Role Hierarchies

All functions of section A.2a-2 remain valid.

A.2b-3 Review Functions for Limited Role Hierarchies

All functions of section A.2a-3 remain valid.

A.2b-4 Advanced Review Functions for Limited Role Hierarchies

All functions of section A.2a-4 remain valid.

A.3 Requirements for Static Separation of Duty (SSD) Relations

The static separation of duty property, as defined in the model, uses a collection SSD of pairs of a role set and an associated cardinality. We define the new data type *SSD*, which in an implementation could be the set of names used to identify the pairs in the collection.

The functions ssd_set and respectively ssd_card are used to obtain the role set and the associated cardinality from each SSD pair:

$$ssd\_set: SSD \rightarrow 2^{ROLES}$$

$$ssd\_card: SSD \rightarrow \mathbb{N}$$

$$\forall ssd \in SSD \bullet ssd\_card(ssd) \geq 2 \wedge ssd\_card(ssd) \leq |ssd\_set(ssd)|$$

A.3a SSD Relations

A.3a-1 Administrative commands for SSD Relations

This section redefines the function AssignUser of section A.1-1 and defines a set of new, specific functions. The other functions of section A.1-1 remain valid.

AssignUser
The AssignUser command replaces the command with the same name of Core RBAC. This command assigns a user to a role. The command is valid if and only if:
- the user is a member of the *USERS* data set, and
- the role is a member of the *ROLES* data set, and
- the user is not already assigned to the role, and
- the SSD constraints are satisfied after assignment.

The data set UA and the function *assigned_users* are updated to reflect the assignment. The following schema formally describes the command:

$$AssignUser(user, role: NAME) \lhd$$

$$user \in USERS; role \in ROLES; (user \mapsto role) \notin UA$$

$$\forall ssd \in SSD \bullet \bigcap_{\substack{r \in subset \\ subset \subseteq ssd\_set(ssd) \\ |subset| = ssd\_card(ssd) \\ us = \text{if } r = role \text{ then } \{user\} \text{ else } \varnothing}} (assigned\_users(r) \cup us) = \varnothing$$

$$UA' = UA \cup \{user \mapsto role\}$$

$$assigned\_users' = assigned\_users \setminus \{role \mapsto assigned\_users(role)\} \cup \{role \mapsto (assigned\_users(role) \cup \{user\})\} \rhd$$

CreateSsdSet
This command creates a named SSD set of roles and sets the cardinality *n* of its subsets that cannot have common users. The command is valid if and only if:
- the name of the SSD set is not already in use
- all the roles in the SSD set are members of the *ROLES* data set
- *n* is a natural number greater than or equal to 2 and less than or equal to the cardinality of the SSD role set, and
- the SSD constraint for the new role set is satisfied.

The following schema formally describes this command:

$$CreateSsdSet(set\_name: NAME; role\_set: 2^{NAMES}; n: \mathbb{N}) \lhd$$

$$set\_name \notin SSD; (n \geq 2) \wedge (n \leq |role\_set|); role\_set \subseteq ROLES$$

$$\bigcap_{\substack{r \in subset \\ subset \subseteq role\_set \\ |subset| = n}} assigned\_users(r) = \varnothing$$

$$SSD' = SSD \cup \{set\_name\}$$

$$ssd\_set' = ssd\_set \cup \{set\_name \mapsto role\_set\}$$

$$ssd\_card' = ssd\_card \cup \{set\_name \mapsto n\} \rhd$$

AddSsdRoleMember

This command adds a role to a named SSD set of roles. The cardinality associated with the role set remains unchanged. The command is valid if and only if:

- the SSD role set exists, and
- the role to be added is a member of the *ROLES* data set but not of a member of the SSD role set, and
- the SSD constraint is satisfied after the addition of the role to the SSD role set.

The following schema formally describes the command:

$$AddSsdRoleMember(set\_name: NAME; role: NAME) \lhd$$
$$set\_name \in SSD; role \in ROLES; role \notin ssd\_set(set\_name)$$
$$\bigcap_{\substack{r \in subset \\ subset \subseteq ssd\_set(set\_name) \cup \{role\} \\ |subset| = n}} assigned\_users(r) = \varnothing$$
$$ssd\_set' = ssd\_set \setminus \{set\_name \mapsto ssd\_set(set\_name)\} \cup$$
$$\{set\_name \mapsto (ssd\_set(set\_name) \cup \{role\})\} \rhd$$

DeleteSsdRoleMember

This command removes a role from a named SSD set of roles. The cardinality associated with the role set remains unchanged. The command is valid if and only if:

- the SSD role set exists, and
- the role to be removed is a member of the SSD role set, and
- the cardinality associated with the SSD role set is less than the number of elements of the SSD role set.

Note that the SSD constraint should be satisfied after the removal of the role from the SSD role set. The following schema formally describes the command:

$$DeleteSsdRoleMember(set\_name: NAME; role: NAME) \lhd$$
$$set\_name \in SSD; role \in ssd\_set(set\_name); ssd\_card(set\_name) < |ssd\_set(set\_name)|$$
$$ssd\_set' = ssd\_set \setminus \{set\_name \mapsto ssd\_set(set\_name)\} \cup$$
$$\{set\_name \mapsto (ssd\_set(set\_name) \setminus \{role\})\} \rhd$$

DeleteSsdSet

This command deletes a SSD role set completely. The command is valid if and only if the SSD role set exists. The following schema formally describes the command:

$$DeleteSsdSet(set\_name: NAME) \lhd$$
$$set\_name \in SSD; ssd\_card' = ssd\_card \setminus \{set\_name \mapsto ssd\_card(set\_name)\}$$
$$ssd\_set' = ssd\_set \setminus \{set\_name \mapsto ssd\_set(set\_name)\}$$
$$SSD' = SSD \setminus \{set\_name\} \rhd$$

SetSsdSetCardinality

This command sets the cardinality associated with a given SSD role set. The command is valid if and only if:

- the SSD role set exists, and
- the new cardinality is a natural number greater than or equal to 2 and less than or equal to the number of elements of the SSD role set, and
- the SSD constraint is satisfied after setting the new cardinality.

The following schema formally describes the command:

$$SetSsdSetCardinality(set\_name: NAME; n: \mathbb{N}) \lhd$$
$$set\_name \in SSD; (n \geq 2) \wedge (n \leq |ssd\_set(set\_name)|)$$
$$\bigcap_{\substack{r \in subset \\ subset \subseteq ssd\_set(set\_name) \\ |subset|=n}} assigned\_users(r) = \varnothing$$
$$ssd\_card' = ssd\_card \setminus \{set\_name \mapsto ssd\_card(set\_name)\} \cup \{set\_name \mapsto n\} \rhd$$

A.3a-2 System Functions for SSD

All functions in section A.1-2 remain valid.

A.3a-3 Review Functions for SSD

All functions in section A.1-3 remain valid. In addition, this section defines the following functions:

SsdRoleSets
This function returns the list of all SSD role sets. The following schema formally describes the function:

$$SsdRoleSets(out\ result: 2^{NAME}) \lhd result = SSD \rhd$$

SsdRoleSetRoles
This function returns the set of roles of a SSD role set. The function is valid if and only if the role set exists. The following schema formally describes the function:

$$SsdRoleSetRoles(set\_name: NAME; out\ result: 2^{ROLES}) \lhd$$
$$set\_name \in SSD$$
$$result = ssd\_set(set\_name) \rhd$$

SsdRoleSetCardinality
This function returns the cardinality associated with a SSD role set. The function is valid if and only if the role set exists. The following schema formally describes the function:

$$SsdRoleSetCardinality(set\_name: NAME; out\ result: \mathbb{N}) \lhd$$
$$set\_name \in SSD$$
$$result = ssd\_card(set\_name) \rhd$$

A.3a-4 Advanced Review Functions for SSD

All functions in section A.1-4 remain valid.

A.3b SSD Relations with General Role Hierarchies

A.3b-1 Administrative Commands for SSD with General Role Hierarchies

This section redefines the functions AssignUser and AddInheritance of section A.2a-1, and the functions CreateSsdSet, AddSsdRoleMember, SetSsdSetCardinality of section A.3a-1. The other functions of sections A.2a-1 and A.3a-1 remain valid.

AssignUser

The command AssignUser replaces the command with the same name from Core RBAC with Static Separation of Duties. This command assigns a user to a role. The command is valid if and only if:

- the user is a member of the *USERS* data set, and
- the role is a member of the *ROLES* data set, and
- the user is not already assigned to the role, and
- the SSD constraints are satisfied after assignment.

The data set UA and the function *assigned_users* are updated to reflect the assignment. The following schema formally describes the command:

$$AssignUser(user, role: NAME) \lhd$$

$$user \in USERS; role \in ROLES; (user \mapsto role) \notin UA$$

$$\forall ssd \in SSD \bullet \bigcap_{\substack{r \in subset \\ subset \subseteq ssd\_set(ssd) \\ |subset| = ssd\_card(ssd) \\ au = \text{if } r = role \text{ then } \{user\} \text{ else } \varnothing}} (authorized\_users(r) \cup au) = \varnothing$$

$$UA' = UA \cup \{user \mapsto role\}$$

$$assigned\_users' = assigned\_users \setminus \{role \mapsto assigned\_users(role)\} \cup \{role \mapsto (assigned\_users(role) \cup \{user\})\} \rhd$$

AddInheritance

This commands establishes a new immediate inheritance relationship *r_asc* ≫ *r_desc* between existing roles *r_asc*, *r_desc*. The command is valid if and only if:

- *r_asc* and *r_desc* are members of the *ROLES* data set, and
- *r_asc* is not an immediate ascendant of *r_desc*, and
- *r_desc* does not properly inherit *r_asc*, and

the SSD constraints are satisfied after establishing the new inheritance.

The following schema uses the notations:

$$\mu == /$$

$$>> == \gg$$

to formally describes the command:

$$AddInheritance(r\_asc, r\_desc: NAME) \lhd$$

$$r\_asc \in ROLES; r\_desc \in ROLES; \neg(r\_asc >> r\_desc); \neg(r\_desc \geq r\_asc)$$

$$\forall ssd \in SSD \bullet \bigcap_{\substack{r \in subset \\ subset \subseteq ssd\_set(ssd) \\ |subset| = ssd\_card(ssd) \\ au = \text{if } r = r\_desc \text{ then } authorized\_users(r\_asc) \text{ else } \varnothing}} (authorized\_users(r) \cup au) = \varnothing$$

$$\geq' = \geq \cup \{r, q: ROLES \mid r \geq r\_asc \wedge r\_desc \geq q \bullet r \mapsto q\} \rhd$$

CreateSsdSet

This command creates a named SSD set of roles and sets the associated cardinality *n* of its subsets that cannot have common users. The command is valid if and only if:

- the name of the SSD set is not already in use
- all the roles in the SSD set are members of the *ROLES* data set
- *n* is a natural number greater than or equal to 2 and less than or equal to the cardinality of the SSD role set, and
- the SSD constraint for the new role set is satisfied.

The following schema formally describes this command:

$$CreateSsdSet(set\_name: NAME; role\_set: 2^{NAMES}; n: \mathbb{N}) \lhd$$

$$set\_name \notin SSD; (n \geq 2) \wedge (n \leq |role\_set|); role\_set \subseteq ROLES$$

$$\bigcap_{\substack{r \in subset \\ subset \subseteq role\_set \\ |subset| = n}} authorized\_users(r) = \emptyset$$

$$SSD' = SSD \cup \{set\_name\}$$

$$ssd\_set' = ssd\_set \cup \{set\_name \mapsto role\_set\}$$

$$ssd\_card' = ssd\_card \cup \{set\_name \mapsto n\} \rhd$$

AddSsdRoleMember

This command adds a role to a named SSD set of roles. The cardinality associated with the role set remains unchanged. The command is valid if and only if:

- the SSD role set exists, and
- the role to be added is a member of the *ROLES* data set but not of a member of the SSD role set, and
- the SSD constraint is satisfied after the addition of the role to the SSD role set.

The following schema formally describes the command:

$$AddSsdRoleMember(set\_name: NAME; role: NAME) \lhd$$

$$set\_name \in SSD; role \in ROLES; role \notin ssd\_set(set\_name)$$

$$\bigcap_{\substack{r \in subset \\ subset \subseteq ssd\_set(set\_name) \cup \{role\} \\ |subset| = n}} authorized\_users(r) = \emptyset$$

$$ssd\_set' = ssd\_set \setminus \{set\_name \mapsto ssd\_set(set\_name)\} \cup$$
$$\{set\_name \mapsto (ssd\_set(set\_name) \cup \{role\})\} \rhd$$

SetSsdSetCardinality

This command sets the cardinality associated with a given SSD role set. The command is valid if and only if:

- the SSD role set exists, and
- the new cardinality is a natural number greater than or equal to 2 and less than or equal to the number of elements of the SSD role set, and
- the SSD constraint is satisfied after setting the new cardinality.

The following schema formally describes the command:

$$SetSsdSetCardinality(set\_name: NAME; n: \mathbb{N}) \lhd$$

$$set\_name \in SSD; (n \geq 2) \wedge (n \leq |ssd\_set(set\_name)|)$$

$$\bigcap_{\substack{r \in subset \\ subset \subseteq ssd\_set(set\_name) \\ |subset| = n}} authorized\_users(r) = \emptyset$$

$$ssd\_card' = ssd\_card \setminus \{set\_name \mapsto ssd\_card(set\_name)\} \cup \{set\_name \mapsto n\} \rhd$$

A.3b-2 System Functions for SSD with General Role Hierarchies

All functions of section A.2a-2 remain valid.

A.3b-3 Review Functions for SSD with General Role Hierarchies

All functions of sections A.2a-3 and A.3a-3 remain valid.

A.3b-4 Advanced Review Functions for SSD with General Role Hierarchies

All functions of section A.2a-4 remain valid.

A.3c SSD Relations with Limited Role Hierarchies

A.3c-1 Administrative Commands for SSD with Limited Role Hierarchies

This section redefines the function AddInheritance of section A.3b-1. All other functions of section A.3b-1 remain valid.

AddInheritance

This commands establishes a new immediate inheritance relationship *r_asc* νν *r_desc* between existing roles *r_asc*, *r_desc*. The command is valid if and only if *r_asc* and *r_desc* are members of the *ROLES* data set, *r_asc* has no descendants, and *r_desc* does not properly inherit *r_asc* (in order to avoid cycle creation). The following schema uses the notations:

$\mu == /$

$>> == \nu\nu$

to formally describes the command:

$$AddInheritance(r\_asc, r\_desc: NAME) \lhd$$

$$r\_asc \in ROLES; r\_desc \in ROLES; \forall r \in ROLES \bullet \neg(r\_asc >> r); \neg(r\_desc \geq r\_asc)$$

$$\forall ssd \in SSD \bullet \bigcap_{\substack{r \in subset \\ subset \subseteq ssd\_set(ssd) \\ |subset| = ssd\_card(ssd) \\ au = \text{if } r = r\_desc \text{ then } authorized\_users(r\_asc) \text{ else } \varnothing}} (authorized\_users(r) \cup au) = \varnothing$$

$$\geq' = \geq \cup \{r, q: ROLES \mid r \geq r\_asc \wedge r\_desc \geq q \bullet r \mapsto q\} \rhd$$

A.3c-2 System Functions for SSD with Limited Role Hierarchies

All functions of section A.2a-2 remain valid.

A.3c-3 Review Functions for SSD with Limited Role Hierarchies

All functions of sections A.2a-3 and A.3a-3 remain valid.

A.3c-4 Advanced Review Functions for SSD with Limited Role Hierarchies

All functions of sections A.2a-4 remain valid.

A.4 Requirements for Dynamic Separation of Duties (DSD) Relations

The dynamic separation of duty property, as defined in the model, uses a collection DSD of pairs of a role set and an associated cardinality. We define the new data type *DSD*, which in an implementation could be the set of names used to identify the pairs in the collection.

The functions *dsd_set* and respectively *dsd_card* are used to obtain the role set and the associated cardinality from each DSD pair:

$$dsd\_set: DSD \rightarrow 2^{ROLES}$$
$$dsd\_card: DSD \rightarrow \mathbb{N}$$
$$\forall dsd \in SSD \bullet dsd\_card(dsd) \geq 2 \land dsd\_card(dsd) \leq |dsd\_set(dsd)|$$

4.4a DSD Relations

A.4a-1 Administrative Commands for DSD Relations

All functions of section A.1-1 remain valid. In addition, this section defines the following functions:

CreateDsdSet
This command creates a named DSD set of roles and sets an associated cardinality *n*. The DSD constraint stipulates that the DSD role set cannot contain *n* or more roles simultaneously active in the same session.
The command is valid if and only if:
- the name of the DSD set is not already in use
- all the roles in the DSD set are members of the *ROLES* data set
- *n* is a natural number greater than or equal to 2 and less than or equal to the cardinality of the DSD role set, and
- the DSD constraint for the new role set is satisfied.

The following schema formally describes this command:

$$CreateDsdSet(set\_name: NAME; role\_set: 2^{NAMES}; n: \mathbb{N}) \lhd$$
$$set\_name \notin DSD; (n \geq 2) \land (n \leq |role\_set|); role\_set \subseteq ROLES$$
$$\forall s: SESSIONS; role\_subset: 2^{role\_set} \bullet role\_subset \subseteq session\_roles(s) \Rightarrow |role\_subset| < n$$
$$DSD' = DSD \cup \{set\_name\}$$
$$dsd\_set' = dsd\_set \cup \{set\_name \mapsto role\_set\}$$
$$dsd\_card' = dsd\_card \cup \{set\_name \mapsto n\} \rhd$$

AddDsdRoleMember
This command adds a role to a named DSD set of roles. The cardinality associated with the role set remains unchanged. The command is valid if and only if:
- the DSD role set exists, and
- the role to be added is a member of the *ROLES* data set but not of a member of the DSD role set, and
- the DSD constraint is satisfied after the addition of the role to the DSD role set.

The following schema formally describes the command:

$$AddDsdRoleMember(set\_name: NAME; role: NAME) \lhd$$
$$set\_name \in DSD; role \in ROLES; role \notin dsd\_set(set\_name)$$
$$\forall s: SESSIONS; role\_subset: 2^{dsd\_set(set\_name) \cup \{role\}} \bullet$$
$$role\_subset \subseteq session\_roles(s) \Rightarrow |role\_subset| < dsd\_card(set\_name)$$
$$dsd\_set' = dsd\_set \setminus \{set\_name \mapsto dsd\_set(set\_name)\} \cup$$
$$\{set\_name \mapsto (dsd\_set(set\_name) \cup \{role\})\} \rhd$$

DeleteDsdRoleMember
This command removes a role from a named DSD set of roles. The cardinality associated with the role set remains unchanged. The command is valid if and only if:
- the DSD role set exists, and
- the role to be removed is a member of the DSD role set, and

- the cardinality associated with the DSD role set is less than the number of elements of the DSD role set.

Note that the DSD constraint should be satisfied after the removal of the role from the DSD role set. The following schema formally describes the command:

$$DeleteDsdRoleMember(set\_name: NAME; role: NAME) \triangleleft$$
$$set\_name \in DSD; role \in dsd\_set(set\_name); dsd\_card(set\_name) < |dsd\_set(set\_name)|$$
$$dsd\_set' = dsd\_set \setminus \{set\_name \mapsto dsd\_set(set\_name)\} \cup$$
$$\{set\_name \mapsto (dsd\_set(set\_name) \setminus \{role\})\} \triangleright$$

DeleteDsdSet

This command deletes a DSD role set completely. The command is valid if and only if the DSD role set exists. The following schema formally describes the command:

$$DeleteDsdSet(set\_name: NAME)$$
$$\{$$
$$set\_name \in DSD$$
$$dsd\_card' = dsd\_card \setminus \{set\_name \mapsto dsd\_card(set\_name)\}$$
$$dsd\_set' = dsd\_set \setminus \{set\_name \mapsto dsd\_set(set\_name)\}$$
$$DSD' = DSD \setminus \{set\_name\}$$
$$\}$$

SetDsdSetCardinality

This command sets the cardinality associated with a given DSD role set. The command is valid if and only if:

- the DSD role set exists, and
- the new cardinality is a natural number greater than or equal to 2 and less than or equal to the number of elements of the DSD role set, and
- the DSD constraint is satisfied after setting the new cardinality.

The following schema formally describes the command:

$$SetDsdSetCardinality(set\_name: NAME; n: \mathbb{N}) \triangleleft$$
$$set\_name \in DSD; (n \geq 2) \wedge (n \leq |dsd\_set(set\_name)|)$$
$$\forall s: SESSIONS; role\_subset: 2^{dsd\_set(set\_name)} \bullet$$
$$role\_subset \subseteq session\_roles(s) \Rightarrow |role\_subset| < n$$
$$dsd\_card' = dsd\_card \setminus \{set\_name \mapsto dsd\_card(set\_name)\} \cup \{set\_name \mapsto n\} \triangleright$$

A.4a-2 System Functions for DSD Relations

This section redefines the functions CreateSession and AddActiveRole of section A.1-2. The other functions of section A.1-2 remain valid.

CreateSession

This function creates a new session whose owner is the user *user* and a given active role set. The function is valid if and only if:

- the user is a member of the *USERS* data set, and
- the session's active role set is a subset of the roles assigned to the session's owner, and
- the session's active role set satisfies the DSD constraints.

The following schema formally describes the function. The *session* parameter, which identifies the new session, is actually generated by the underlying system.

$$CreateSession(user: NAME; ars: 2^{NAME}; session: NAME) \lhd$$
$$user \in USERS; ars \subseteq \{r: ROLES \mid (user \mapsto r) \in UA\}; session \notin SESSIONS$$
$$\forall dset: DSD; rset: 2^{NAME} \bullet$$
$$rset \subseteq dsd\_set(dset) \wedge rset \subseteq ars \Rightarrow |rset| < dsd\_card(dset)$$
$$SESSIONS' = SESSIONS \cup \{session\}$$
$$user\_sessions' = user\_sessions \setminus \{user \mapsto user\_sessions(user)\} \cup$$
$$\{user \mapsto (user\_sessions(user) \cup \{session\})\}$$
$$session\_roles' = session\_roles \cup \{session \mapsto ars\} \rhd$$

AddActiveRole

This function adds a role as an active role of a session whose owner is a given user. The function is valid if and only if:

- the user is a member of the *USERS* data set, and
- the role is a member of the *ROLES* data set, and
- the session identifier is a member of the *SESSIONS* data set, and
- the role is assigned to the user, and
- the old active role set completed with the role to be activated satisfies the DSD constraints, and
- the session is owned by that user.

The following schema formally describes the function:

$$AddActiveRole(user, session, role: NAME) \lhd$$
$$user \in USERS; session \in SESSIONS; role \in ROLES; session \in user\_sessions(user)$$
$$user \in assigned\_users(role); role \notin session\_roles(session)$$
$$\forall dset: DSD; rset: 2^{NAME} \bullet$$
$$rset \subseteq dsd\_set(dset) \wedge rset \subseteq session\_roles(session) \cup \{role\} \Rightarrow |rset| < dsd\_card(dset)$$
$$session\_roles' = session\_roles \setminus \{session \mapsto session\_roles(session)\} \cup$$
$$\{session \mapsto (session\_roles(session) \cup \{role\})\} \rhd$$

A.4a-3 Review Functions for DSD Relations

All functions of sections A.1-3 remain valid. In addition, this section defines new, specific functions.

DsdRoleSets

This function returns the list of all DSD role sets. The following schema formally describes the function:

$$DsdRoleSets(out\ result: 2^{NAME}) \lhd result = DSD \rhd$$

DsdRoleSetRoles

This function returns the set of roles of a DSD role set. The function is valid if and only if the role set exists. The following schema formally describes the function:

$$DsdRoleSetRoles(set\_name: NAME; out\ result: 2^{ROLES}) \lhd$$
$$set\_name \in DSD$$
$$result = dsd\_set(set\_name) \rhd$$

DsdRoleSetCardinality
This function returns the cardinality associated with a DSD role set. The function is valid if and only if the role set exists. The following schema formally describes the function:

$$DsdRoleSetCardinality(set\_name: NAME; out\ result: \mathbb{N}) \lhd$$
$$set\_name \in DSD$$
$$result = dsd\_card(set\_name) \rhd$$

A.4a-4 Advanced Review Functions for DSD Relations

All functions of sections A.1-4 remain valid.

A.4b DSD Relations with Role Hierarchies

A.4b-1 Administrative commands for DSD Relations with General Role Hierarchies

All functions of sections A.4a-1 and A.2a-1 remain valid.

A.4b-2 System Functions for DSD Relations with General Role Hierarchies

This section redefines the functions CreateSession and AddActiveRole of section A.1-2 (or A.2a-2). All other functions of section A.1-2 remain valid.

CreateSession
This function creates a new session whose owner is the user *user* and a given active role set. The function is valid if and only if:
- the user is a member of the *USERS* data set, and
- the session's active role set is a subset of the roles authorized for the session's owner, and
- the session's active role set satisfies the DSD constraints.

The underlying system generates a new session identifier, which is included in the *SESSIONS* data set.
The following schema formally describes the function:

$$CreateSession(user: NAME; ars: 2^{NAME}; session: NAME) \lhd$$
$$user \in USERS; ars \subseteq \{r, q: ROLES \mid (user \mapsto q) \in UA \wedge q \geq r \bullet r\}; session \notin SESSIONS$$
$$\forall dset: DSD; rset: 2^{NAME} \bullet$$
$$rset \subseteq dsd\_set(dset) \wedge rset \subseteq ars \Rightarrow |rset| < dsd\_card(dset)$$
$$SESSIONS' = SESSIONS \cup \{session\}$$
$$user\_sessions' = user\_sessions \setminus \{user \mapsto user\_sessions(user)\} \cup$$
$$\{user \mapsto (user\_sessions(user) \cup \{session\})\}$$
$$session\_roles' = session\_roles \cup \{session \mapsto ars\} \rhd$$

AddActiveRole
This function adds a role as an active role of a session whose owner is a given user. The function is valid if and only if:
- the user is a member of the *USERS* data set, and
- the role is a member of the *ROLES* data set, and
- the session identifier is a member of the *SESSIONS* data set, and
- the role is authorized for that user, and

- the old active role set completed with the role to be activated satisfies the DSD constraints, and
- the session is owned by that user.

The following schema formally describes the function:

$$AddActiveRole(user, session, role: NAME) \lhd$$
$$user \in USERS; session \in SESSIONS; role \in ROLES; session \in user\_sessions(user)$$
$$user \in authorized\_users(role); role \notin session\_roles(session)$$
$$\forall dset: DSD; rset: 2^{NAME} \bullet$$
$$rset \subseteq dsd\_set(dset) \wedge rset \subseteq session\_roles(session) \cup \{role\} \Rightarrow |rset| < dsd\_card(dset)$$
$$session\_roles' = session\_roles \setminus \{session \mapsto session\_roles(session)\} \cup$$
$$\{session \mapsto (session\_roles(session) \cup \{role\})\} \rhd$$

A.4b-3 Review Functions for DSD Relations with General Role Hierarchies

All functions of sections A.4a-3 and A.2a-3 remain valid.

A.4b-3 Advanced Review Functions for DSD Relations with General Role Hierarchies

All functions of section A.2a-4 remain valid.

A.4c DSD Relations with Limited Role Hierarchies

A.4c-1 Administrative Commands for DSD Relations with Limited Role Hierarchies

All functions of sections A.2b-1and A.4a-1 remain valid.

A.4c-2 System Functions for DSD Relations with Limited Role Hierarchies

All functions of section A.4b-2 remain valid.

A.4c-3 Review Functions for DSD Relations with Limited Role Hierarchies

All functions of section A.4b-3 remain valid.

A.4c-4 Advanced Review Functions for DSD Relations with Limited Role Hierarchies

All functions of section A.2a-4 remain valid.