ANÁLISE DE ASSOCIAÇÕES SEGURAS EM SISTEMAS MÓVEIS DE TERCEIRA GERAÇÃO

Fabrício Jorge Lopes Ribeiro

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

_____

Prof. Aloysio de Castro Pinto Pedroza, Dr.

_____

Prof. José Ferreira de Rezende, Dr.

_____

Prof. Julius Cesar Barreto Leite, PhD.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2004

RIBEIRO, FABRÍCIO JORGE LOPES

Análise de Associações Seguras em Sistemas Móveis de Terceira Geração [Rio de Janeiro] 2004

XVI, 114 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2004)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Associações Seguras

2. Sistema Móvel de 3ª Geração

3. Métodos Formais

I. COPPE/UFRJ II. Título (série)

Dedicatória:

À minha esposa Aline Familiar Solano Ribeiro, aos meus pais Dalcy Jorge da Cruz Ribeiro e Lysahir Lopes Ribeiro e minha irmã Fabiola Simone Lopes Ribeiro.

Agradecimentos:

A Deus.

À minha esposa sempre presente.

Aos meus pais e irmã que me incentivaram.

Ao Professor Aloysio de Castro Pinto Pedroza, pela orientação que foi fundamental para a realização deste trabalho.

A Sol Rio pela compreensão e colaboração.

Aos amigos de trabalho que me apoiaram durante todo estudo.

Aos meus amigos Bagatelli, David e Jaime que foram colaboradores durante todo o curso.

A todos os amigos da COPPE.

A todos os funcionários do Departamento.

A todos os professores do GTA Jorge Lopes de Souza Leão, José Ferreira de Rezende, Otto Carlos Muniz Bandeira Duarte que muito contribuíram para a conclusão deste trabalho.

Ao professor Julius Cesar Barreto Leite que participou da Comissão Examinadora.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ANÁLISE DE ASSOCIAÇÕES SEGURAS EM SISTEMAS MÓVEIS DE TERCEIRA GERAÇÃO

Fabrício Jorge Lopes Ribeiro

Março/2004

Orientador: Aloysio de Castro Pinto Pedroza

Programa: Engenharia Elétrica

O sistema móvel de terceira geração (3G) tem como ponto principal a convergência das redes de comunicação. No entanto, a busca desse objetivo depende do desenvolvimento de uma arquitetura de segurança robusta. Toda segurança provida pelo sistema depende da realização de acordos entre as partes envolvidas na comunicação. Neste processo, o estabelecimento da associação segura é responsável pelo acordo que é realizado através da troca de parâmetros de proteção das mensagens. Este trabalho apresenta um estudo do protocolo que estabelece as associações seguras, realizando a análise das características essenciais para o seu correto funcionamento no sistema móvel de 3ª geração. Como muitas partes desta arquitetura de segurança ainda não foram definidas pelo fórum 3GPP (*Third Generation Partnership Project*), foi proposta a utilização do protocolo ISA nesta análise. As especificações foram feitas com o auxílio de técnicas de descrição formal utilizando a linguagem LOTOS e uma ferramenta de análise apropriada para teste, simulação e verificação do protocolo. Com a análise, foi possível verificar as propriedades essenciais para o correto funcionamento do protocolo.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

THIRD GENERATION MOBILE SYSTEMS SECURITY ASSOCIATIONS ANALYSIS

Fabrício Jorge Lopes Ribeiro

March/2004

Advisor: Aloysio de Castro Pinto Pedroza

Department: Electrical Engineering

The 3rd Generation Mobile System has focus communications networks convergence. However, this objective depends on a robust security architecture development. The security of all systems depends on the agreement achieved by the communication peers. The security associations are responsible for this deal that is worked out by the exchange of parameters for message protection. This work shows a study of a security associations protocol, through the analysis of its correct behavior in the 3G mobile system. As there aren't definitions of various aspects of the security architecture in the 3GPP forum (Third Generation Partnership Project), the ISA protocol was used in this analysis. Formal description techniques were used to analyze these specifications, employing LOTOS language, and a suitable development package was used to perform protocol test, simulation and verification. With the analysis it was possible to verify the essential properties of the protocol.

Sumário

Lista de Acrônimos:

3GPP:	*Third Generation Partnership Project*
3G:	Terceira Geração
AH:	*Authentication Header*
AL:	Agente Local
API:	*Aplication Programer Interface*
AS:	Associação Segura
AR:	Agente de Rede
AU:	Agente de Usuário
BCG:	*Binary Code Graphic*
CADP:	*Caesar/Aldebaran Developement Packege*
CCS:	*Calculus Comunnication System*
CSCF:	*Call State Control Function*
CSP:	*Calculus Sequencial Processes*
DoI:	*Domain of Interpretation*
DoS:	*Denial of Service*
DES:	*Data Encryption Standard*
ESP:	*Encapsulating Security Payload*
ELUDO:	*Environnement LOTOS de l'Université d'Ottawa*
ETSI:	*European Telecommunications Standards Institute*

EU:	Equipamento do Usuário
GGSM:	*Gateway GSM*
GPRS:	*General Packet Radio Service*
GSM:	*Global System for Mobile communications*
GTP:	*GPRS Tunnelling Protocols*
HE:	*Home Environment*
HSS:	*Home Subscriber Server*
IETF:	*Internet Engineering Task Force*
IKE:	*Internet Key Exchange*
IMSI:	*International Mobile Subscriber Identity*
IP:	*Internet Protocol*
IPsec:	*IP security*
ISA:	*Interconnection Security Association*
ISAKMP:	*Internet Security Association Key Management Protocol*
INRIA:	*Institut National de Recherche em Informatique et em Automatique*
ISAKMP:	*Internet Security Association and Key Management Protocol*
ISO:	*International Standards Organization*
ITU:	*International Telecommunications Union*
KEI:	*Key Exchange Identifier*
LOTOS:	*Language of Temporal Ordering Specification*
LTS:	*Labelled Transition Systems*
ME:	*Mobile Equipment*
OAKLEY:	*Key Determination Protocol*
OSI:	*Open System Interconnection*
PAS:	Pedido de Associação Segura
PDU:	*Protocol Data Unit*

RAN: *Radio Access Network*

RAS: Resposta de Associação Segura

RNC: *Radio Network Controller*

RS: Rede Servidora

RTPC: Rede de Telefonia Pública Comutada

SAD: *Security Association Database*

SCTP: *Stream Control Transmission Protocol*

SDU: *Service Data Unit*

SEG: *Security Gateway*

SPI: *Security Parameters Index*

TCP: *Transmission Control Protocol*

UMTS: *Universal Mobile Telecommunication System*

USIM: *User Services Identity Module*

VASY: *Validation of Systems*

Lista de Figuras:

Lista de Tabelas:

Capítulo 1

Introdução

A expansão da telefonia móvel, que propiciou a universalização nas comunicações de voz, também impulsionou a comunicação móvel de dados no sentido de atingir a ubiqüidade. Para este fim, e considerando a demanda crescente dos usuários por aplicações multimídia de alta qualidade gráfica e sonora, está em desenvolvimento e formulação a arquitetura UMTS (*Universal Mobile Telecommunication Systems*) padronizada pelo ITU (*International Telecommunication Union*) [1] e ETSI (*European Telecommunications Standards Institute*) [2].

A arquitetura UMTS, que está sendo empregada no projeto do sistema móvel de 3ª geração (3G) desenvolvido pelo fórum 3GPP (*Third Generation Partnership Project*) [3], vem sendo discutida por fabricantes, órgãos de padronização e entidades de pesquisa. Dentre os vários assuntos que ainda não foram definidos, está a arquitetura de segurança, que possui diversas lacunas em sua descrição. Esta carência de definições permitiu a realização de uma avaliação das associações seguras, que são responsáveis por acordos de parâmetros de segurança entre as redes neste sistema.

A proposta deste trabalho é realizar uma análise formal do protocolo de estabelecimento de associações seguras de forma a validar o seu comportamento no sistema 3G. Para isso, utilizamos como base o protocolo ISAKMP (*Internet Security Association and Key Management Protocol*) [4], integrante da arquitetura de segurança IPSec [5].

O estudo prevê a especificação e validação do protocolo que estabelece as associações seguras com o auxílio da técnica de descrição formal, utilizando a linguagem LOTOS (*Language Of Temporal Ordering Specification*) [6] e a ferramenta de análise contidas no pacote denominado CADP (*CAESAR/ALDEBARAN Development Package*) [7].

1.1. Sistema Móvel de Terceira Geração

A arquitetura do sistema móvel de 3ª geração deverá ser baseada em uma rede IP [8], já que este protocolo tornou-se universal para comunicações em rede. O uso de pacotes IP na estrutura de transmissão de dados e de sinalização apresenta-se como caminho natural para a convergência entre as redes fixas e móveis. Esta convergência acontecerá pela padronização de uma arquitetura totalmente baseada no protocolo IP, que incluirá o sistema celular, as redes fixas e as redes locais sem fio. Sendo assim, muitos dos requisitos de segurança atualmente existentes, ou em definição, deverão também seguir os aspectos já adotados nas redes convencionais e balizar o desenvolvimento dos seus análogos para redes móveis.

Em termos simples, os serviços de 3ª geração (3G) combinam acesso móvel de alta velocidade com serviços baseados no protocolo de Internet (IP). Esta alta velocidade não representa somente maior rapidez de conexão, mas sobretudo novas formas de comunicar, de aceder à informação, de conduzir os negócios, de aprender e também de entretenimento. Agora, de uma forma liberta de conexões lentas, equipamentos incômodos e pontos de acesso imóveis.

A lista de aplicações possíveis para a tecnologia de 3ª geração é muito expressiva, devido à potencial ubiqüidade do serviço a ser oferecido. Navegação por páginas da Internet, comércio eletrônico e multimídia interativa são alguns desses exemplos. Nesses e em outros casos, a segurança dos serviços utilizados nestas redes tornar-se-á essencial para a garantia de viabilidade dessas comunicações. Usuários móveis que não pertençam à rede 3G, também poderão ser capazes de acessar os serviços, desde que sejam garantidas as características mínimas de acesso à rede.

1.2. Segurança no Sistema Móvel de 3ª Geração

Os aspectos de segurança no sistema móvel de 3ª geração são de grande importância devido à vulnerabilidade intrínseca ao processo de comunicação sem fio, como por exemplo, a facilidade de execução de escutas passivas e ataques ativos. A diversidade de infra-estruturas das redes que compõem este sistema obriga a implementação de mecanismos de segurança robustos que garantam a integridade e a privacidade da comunicação, bem como a autenticação das entidades envolvidas [9].

A maior parte dos estudos sobre segurança em redes sem fio concentra-se na tentativa de buscar uma arquitetura totalmente desvinculada dos processos de segurança aplicados nas redes fixas [10]. Já este estudo, supõe que estes processos podem ser adaptados às características deste novo sistema [11], ou seja, que a utilização de uma arquitetura já estabelecida e fundamentada possa ser utilizada sem maiores problemas. Portanto, o desenvolvimento dos protocolos de segurança de 3ª geração deverá basear-se, necessariamente, na arquitetura de segurança IP (IPSec) [5], garantindo às redes sem fio a interoperabilidade com os serviços já utilizados nas redes atuais.

A arquitetura necessária ao desenvolvimento de garantias de segurança deve ser organizada em camadas, cada uma delas provendo parte da segurança requerida. Estas camadas apresentarão um protocolo de sinalização e funções diversas de monitoramento no domínio da operadora e cada uma delas terá seus requisitos definidos com base em padrões de segurança desejados. A proteção das mensagens em toda comunicação se deve a mecanismos que devem atuar da mesma forma em todas as entidades envolvidas na comunicação. Estes mecanismos e seus parâmetros de segurança são acordados pelo processo chamado de associação segura.

As associações seguras são parte importante do sistema de segurança de 3ª geração, pois seu estabelecimento proporciona a criação dos domínios seguros da rede. Estes domínios são um conjunto de dispositivos capazes de trocar informações entre si de forma autêntica, íntegra e com privacidade, proporcionando uma relação de segurança entre si.

Os dispositivos de borda das redes (*gateways*), responsáveis pela comunicação entre as redes, são sempre pontos vulneráveis a ataques. Assim, busca-se implementar nestes dispositivos a capacidade de estabelecerem relações de segurança entre si, para a

composição de um domínio seguro, onde são garantidos perfis de segurança para cada usuário. As mensagens enviadas entre os membros deste domínio podem então ser protegidas de forma que somente seus membros tenham acesso aos seus conteúdos, e eventuais alterações sejam detectadas, impedindo o acesso dos dispositivos que não participem do domínio.

No sistema 3G estas associações são realizadas nos roteadores de borda da rede, os chamados *security gateways* (SEGs), responsáveis pela criação e manutenção dos domínios seguros. Os SEGs devem realizar associações para acordo de parâmetros e perfis de segurança que serão trocados entre as redes, tornando estas associações fundamentais para a interconexão segura das redes que compõem o sistema 3G.

1.3. Protocolo de Estabelecimento de Associações Seguras

No sistema de 3ª geração haverá a coexistência de diferentes redes, plataformas e requisitos, tornando imprescindível a utilização de um protocolo de estabelecimento de associações que garantam a execução de acordos de segurança entre as redes. Entretanto, um ponto crucial considerado na construção deste ambiente é a utilização da arquitetura IP para o transporte das informações. Desta forma, uma arquitetura de segurança para os dados baseada nos protocolos desenvolvidos no grupo de segurança do IETF [12] torna-se extremamente recomendável.

Uma adaptação da arquitetura de segurança IPSec pode ser desenvolvida para prover confiabilidade, integridade aos dados e autenticação entre as redes, criando um ambiente 3G completamente seguro para o tráfego de informações. Dentro deste contexto serão analisadas as associações seguras responsáveis pelos acordos de segurança ente as redes neste ambiente.

A arquitetura proposta requer um protocolo que garanta os requisitos necessários para a criação dos domínios seguros. Neste sentido, a utilização de uma extensão do protocolo ISAKMP (*Internet Security Association and Key Management Protocol*) [4] torna-se o caminho mais adequado para prestação deste serviço. Embora este protocolo tenha sido inicialmente desenvolvido para o estabelecimento de associações seguras em

ambiente convencional, com algumas adaptações pode-se aplicá-lo em qualquer ambiente que utilize redes IP. Assim, a análise das associações seguras será feita através da especificação de um protocolo, que chamaremos de ISA, baseado nas trocas de informações de segurança do protocolo ISAKMP.

1.4. Projeto

A proposta deste trabalho é realizar uma análise formal do protocolo de estabelecimento de associações seguras, dentro de uma arquitetura de segurança baseada na arquitetura IPSec, focando o estudo na validação do correto comportamento do protocolo adaptado às características do sistema móvel de 3ª geração [13].

O modelo para a validação do protocolo de estabelecimento de associações seguras (ISA) consiste na especificação de seu comportamento através do uso da técnica de descrição formal, baseada na linguagem LOTOS, e na análise utilizando as ferramentas contidas no pacote denominado CADP [7]. A abstração conseguida com este processo permite a avaliação científica com critérios mensuráveis e não empíricos, reduzindo o período de teste de campo na avaliação do protocolo. Esta formalização garante a análise do protocolo através da verificação das interações das PDUs e SDUs entre as camadas.

A análise foi dividida em duas partes:

- Análise funcional do protocolo com a determinação do modo de operação utilizado no estabelecimento das associações seguras;
- Análise da evolução comportamental do protocolo com o aumento do número de associações seguras simultâneas entre as redes.

Neste estudo, o Capítulo 2 apresenta a arquitetura de segurança proposta para o sistema móvel de 3ª geração, baseada na estrutura IPSec, assim como no protocolo ISAKMP. O Capítulo 3 apresenta a metodologia e o procedimento aplicados na análise e verificação do comportamento do protocolo ISA. O Capítulo 4 apresenta os resultados obtidos com as simulações e o Capítulo 5 as conclusões do estudo.

Capítulo 2

Segurança no Sistema Móvel de 3ª Geração

2.1. Introdução

A segurança no sistema móvel de 3ª geração é representada pela necessidade de se adequar as características de universalização que define todo o sistema. Em busca deste objetivo foi estabelecido o grupo de trabalho de segurança no fórum 3GPP que é composto por vários órgãos como ITU [1], ETSI [2] e IETF [12]. A grande questão que se apresenta é se a arquitetura de segurança 3G se baseará em uma nova arquitetura ou em arquiteturas já utilizadas e testadas no ambiente das redes fixas.

A fragilidade intrínseca dos sistemas sem fio obriga a utilização de mecanismos de segurança que executem processos que garantam a integridade dos dados e a privacidade, bem como a autenticação das entidades envolvidas na comunicação. Cada dispositivo tem que acordar os parâmetros de segurança que serão utilizados na comunicação. Estes parâmetros são definidos durante as associações, que tem a função de garantir a segurança no encaminhamento de suas mensagens no sistema.

Para se obter uma comunicação segura nas redes de 3ª geração deve-se atender a três requisitos:

- **Autenticação** - garantir que uma entidade é realmente quem ela diz ser;
- **Privacidade** - garantir que somente o destinatário é capaz de compreender o conteúdo da mensagem;
- **Integridade** - garantir que a mensagem recebida pelo destinatário é idêntica àquela que foi enviada pelo emissor.

Estes requisitos são atendidos pelo uso de algoritmos de criptografia e códigos de integridade de mensagem, que funcionam como código de detecção de alterações indevidas nas mensagens transmitidas.

Nas redes de 3ª geração, a existência de entidades centralizadoras que coordenam a comunicação e gerenciam a implementação de mecanismos de segurança facilitam a adaptação de mecanismos desenvolvidos nas redes convencionais, já que sua estrutura é baseada em servidores de autenticação, repositórios de chaves públicas e servidores de características de segurança que trabalham em conjunto para prover a base de toda a segurança do sistema. A figura 1 mostra a arquitetura geral do sistema móvel de 3ª geração, onde podemos observar estes dispositivos interligados a um núcleo de rede IP, que é responsável pela interconexão de todo o sistema.

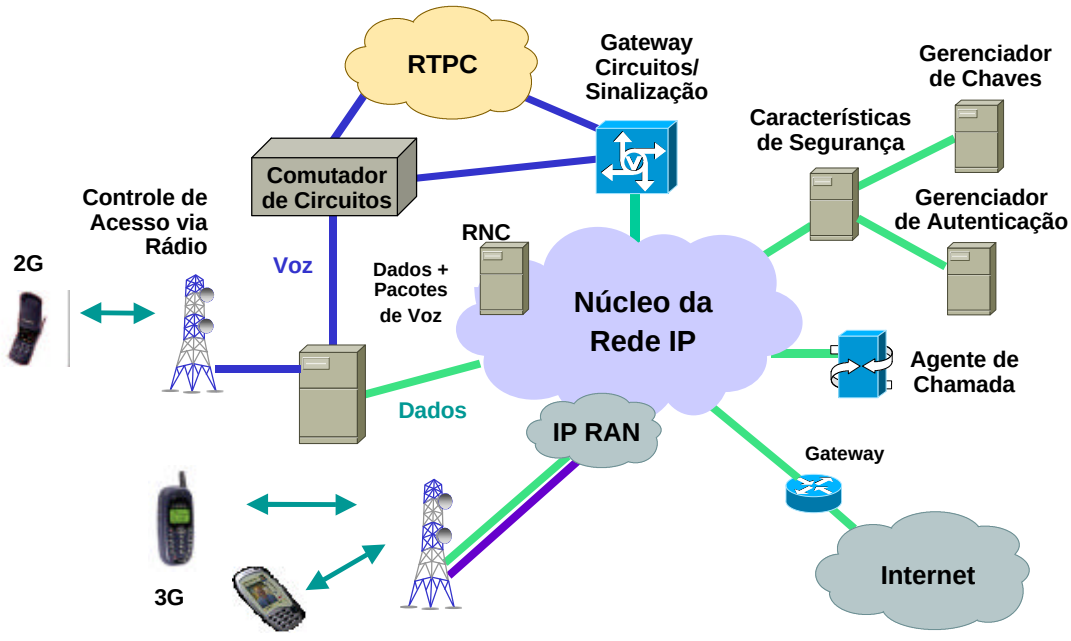

Figura 1. Arquitetura Geral do Sistema Móvel de 3ª Geração.

Este capítulo apresenta na seção 2.2. a arquitetura de segurança das redes de 3ª geração, ressaltando a importância das associações seguras; a seção 2.3. descreve as características da arquitetura IPSec, focando no protocolo responsável pelas associações seguras o ISAKMP; na seção 2.4. é apresentado a arquitetura segurança proposta no estudo; a seção 2.5 apresenta o detalhamento do funcionamento do protocolo ISA; a seção 2.6. apresenta um comentário geral do que foi discutido e o assunto abordado no próximo capítulo.

2.2. A Arquitetura de Segurança nas Redes de 3ª Geração

A arquitetura de segurança proposta será baseada nas recomendações do fórum 3GPP, onde se desenvolve o modelo de segurança do sistema 3G. Esta arquitetura se baseia, necessariamente, nos protocolos que compõem a arquitetura de segurança IP (IPSec) [5], proporcionando às redes sem fio 3G a interoperabilidade dos serviços já utilizados nas redes convencionais.

O desenvolvimento das garantias de segurança do sistema deve ser organizado em camadas, cada qual provendo parte da segurança requerida através de seus protocolos, que exercem funções específicas no domínio da operadora. Por outro lado, os protocolos deverão atuar levando em consideração as limitações do sistema sem fio, como a vulnerabilidade dos canais e as restrições de sinalização.

Com o grande crescimento de usuários na telefonia móvel, adveio a necessidade de aplicações que implementassem o acesso a informações em tempo real e com grande demanda de segurança. Por outro lado, por empregar um meio não delimitado para a comunicação, os sistemas móveis apresentam restrições quanto à segurança das transmissões no que se refere à interceptação. Desta forma, os estudos vinculados à garantia de segurança tornam-se essenciais ao bom desempenho dos sistemas baseados numa arquitetura de 3ª geração.

A comunicação no sistema de 3ª geração seguirá o modelo de domínio de segurança apresentado na figura 2, onde o equipamento móvel para ter acesso a rede deverá ser autenticado pelo CSCF - *Call State Control Function*, através dos parâmetros

de identificação armazenados no HSS - *Home Subscriber Server*. Após da autenticação será definido o perfil de segurança adotado na comunicação.

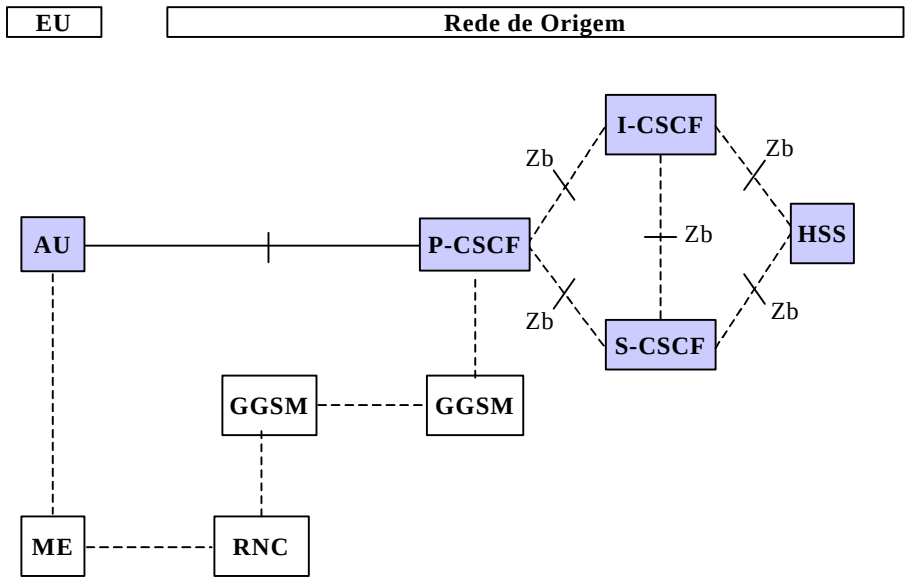

Figura 2. Arquitetura de Comunicação em uma Rede 3G.

A garantia de segurança, deverá ser realmente independente da tecnologia, pelo menos do ponto de vista dos usuários móveis, que desejam usufruir do mesmo conjunto de serviços oferecidos com a mesma segurança, inclusive quando em trânsito. Entretanto, este requisito não será simples de ser atendido, tal é a diversidade de tecnologias e disponibilidade de recursos com que seus equipamentos móveis lidarão durante o trânsito. Assim, foram definidos pontos cruciais para garantia de segurança do sistema como um todo [14], apresentado na figura 3:

- Segurança no Acesso à Rede (I) - define características de segurança para promover acesso seguro aos usuários à rede 3G;
- Segurança na Rede (II) - define características de segurança em um domínio de rede provendo a troca de dados de sinalização e proteção contra ataques;
- Segurança ao Usuário (III) - define características de segurança que asseguram o acesso de estações móveis a um domínio de rede;
- Segurança nas Aplicações (IV) - define características de segurança nas aplicações dos usuários no domínio do provedor, para a troca de mensagens.

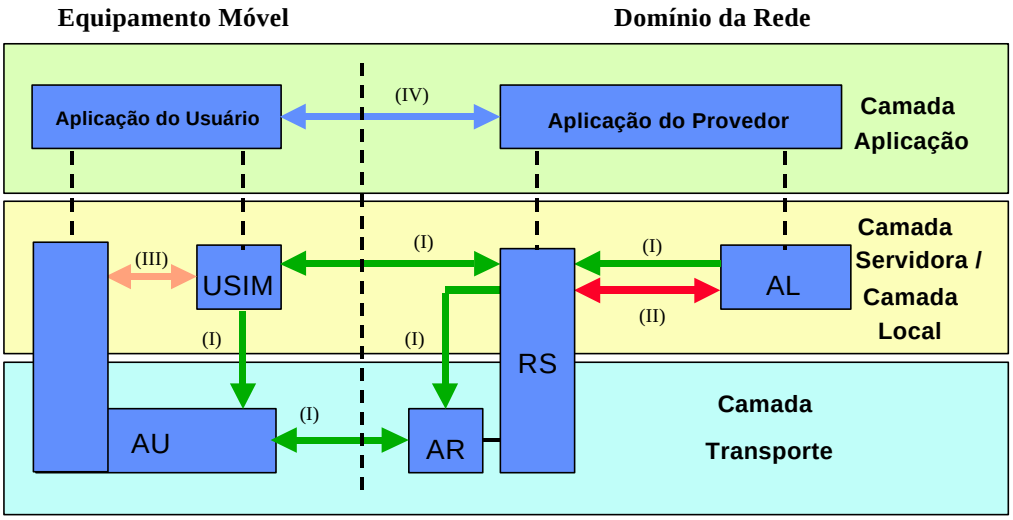

Figura 3.Visão Geral da Arquitetura de Segurança 3G.

Este trabalho pretende incluir conceitos já consagrados na área de segurança adaptados às condições, por muitas vezes desfavoráveis, do ambiente móvel. Como exemplo real, podemos citar a ativação de contexto do protocolo ISAKMP que é um processo de acordo de segurança executado entre os GGSM (*Gateways* GSM) em um sistema GPRS [15, 16]. Durante a ativação de contexto, os parâmetros de segurança (ou seja, o perfil de segurança) são negociados, e desta forma a rede GPRS poderá executar um controle de admissão, por exemplo, através da comparação do perfil de segurança requisitado com os recursos disponíveis no sistema [17].

Para a análise das garantias de segurança assumimos que com as definições de características de segurança da rede 3G [18], podemos modelar comportamentos que levam o protocolo a executar procedimentos seguros [19] e ainda verificar propriedades seguras [20], que são estados que podem acontecer sem prejuízo da segurança no sistema. Por exemplo, a autenticação, o controle de acesso e a integridade são propriedades de segurança, que necessitam de um estado particular que pode acontecer ou não dura nte a execução do protocolo.

Vamos considerar um protocolo de autenticação atuando entre duas entidades, sendo uma delas um provedor de domínio de rede que deve autenticar um usuário. Há dois pontos críticos nesta situação: o primeiro ocorre quando se inicia a autenticação e o segundo, quando é assegurada a identificação do usuário [21]. Outro exemplo de aplicação desse tipo de protocolo segurança é o processo de autenticação e definição de

chaves de criptografia a serem utilizadas por usuários de um sistema UMTS [22].

Estes dois processos se unem quando verificamos o estabelecimento de associações seguras, realizando tanto as negociações de parâmetros e perfis de segurança, quanto a definição das chaves criptográficas utilizadas, que são responsáveis pela criação e manutenção destes domínios. As associações são estabelecidas pelos dispositivos de borda da rede (*Security Gateways* - SEGs) que têm como função garantir a segurança entre conexões extra domínio.

2.2.1. Domínios Seguros

Uma vulnerabilidade identificada no sistema de 2ª geração é a falta de segurança no núcleo da rede, que a princípio não foi tratado como um grande problema, pois era composto por sistemas proprietários e controlados por um número reduzido de instituições. Agora, com a introdução do backbone IP [17], não somente usado para o tráfego de sinalização mas também para o tráfego de usuários, novas ameaças e riscos surgem no sistema 3G.

O estabelecimento de serviços seguros acarreta a necessidade de confiabilidade, integridade e autenticação na comunicação, que podem ser asseguradas com procedimentos padronizados e baseados em técnicas de criptografia. Estes procedimentos tornam imprescindível a implementação dos domínios seguros [5], que são gerenciados por uma única autoridade que define a política de segurança a ser implementada.

O controle dos níveis de segurança é realizado pelos dispositivos de borda da rede (SEGs), sendo responsáveis pela integridade e a autenticação dos dados de origem da comunicação, dividindo lógica e fisicamente as redes em domínios seguros.

2.2.1.1. Segurança nos Roteadores de Borda

Os SEGs são entidades na borda dos domínios que serão usadas na comunicação segura entre as redes para a troca de informações dos serviços, baseados na arquitetura IP [23]. Estes dispositivos controlam as comunicações entre domínios diferentes (interface Za) e entre SEGs e entidades de rede internas no domínio (interface Zb),

conforme a figura 4.

Todo tráfego IP dos domínios seguros deve passar por estes roteadores de borda, sendo a determinação do número destes dispositivos dependente da necessidade do equilíbrio entre a acessibilidade externa e o balanceamento de carga, para evitar um único ponto de falha. Os SEGs são responsáveis por executar a política de segurança nas comunicações entre as redes.

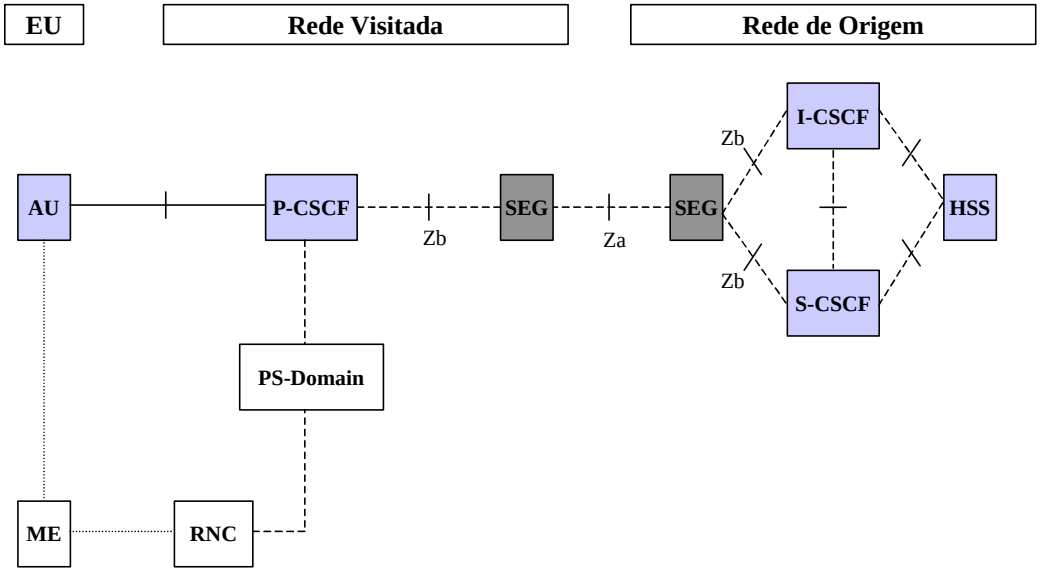

Figura 4. Arquitetura dos Domínios Seguros 3G.

No modelo de segurança proposto para os sistemas de 3ª geração, estes dispositivos de borda devem ter capacidade para oferecer armazenamento seguro das chaves de autenticação usadas na proteção da comunicação.

2.2.2. Associações Seguras

A arquitetura IPSec proporciona segurança aos serviços através do estabelecimento de associações que são conexões unidirecionais, protegidas criptograficamente, que são identificadas através das SPI (*Security Parameter Index*).

Na arquitetura UMTS [22], o estabelecimento das associações seguras poderá ser baseado no protocolo de troca de chaves da Internet (IKE) [24], que realiza todos os procedimentos do protocolo ISAKMP [4]. Este protocolo tem como objetivo principal negociar, estabelecer e manter as associações seguras.

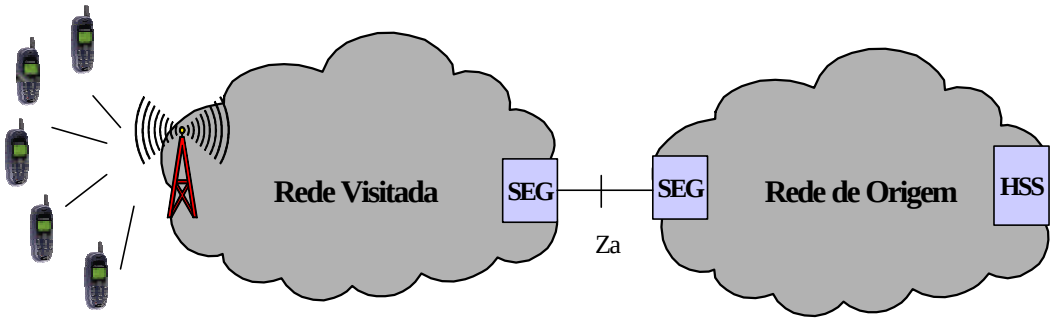

Figura 5. Associações Seguras entre Redes.

Em uma comunicação segura estabelecida entre dois SEGs (vide figura 5), o gerenciamento e a distribuição das chaves poderão ser realizados por um protocolo baseado no ISAKMP [4], pois este é responsável pelo estabelecimento das associações IPSec (uma em cada direção). A criação de uma associação segura é composta por duas fases (ISAKMP AS e IPSec AS).

Na fase ISAKMP AS, que será detalhada na seção 2.3, é realizada a autenticação mútua e o estabelecimento das chaves criptográficas, podendo acontecer de dois modos: o agressivo e o principal, onde são enviados os *cookies* e acordado o algoritmo de criptografia, que resulta em uma chave criptográfica que será utilizada nas mensagens.

Na fase IPSec AS, também conhecida como modo rápido de troca, é estabelecida uma associação AH (*IP Authentication Header*) [25] ou ESP (IP *Encapsulating Security Payload*) [26], que envolve negociação de parâmetros de criptografia e escolha do valor do SPI em cada direção da comunicação.

A próxima seção abordará os procedimentos de segurança do IPSec, relacionados ao estabelecimento das associações seguras nesta arquitetura.

2.3. A Arquitetura IPSec

A arquitetura IPSec [5] permite garantir integridade, confidencialidade e autenticação das comunicações na rede, solucionando os problemas de segurança pertinentes às camadas de rede e transporte. Neste processo, deve ser determinado um conjunto mínimo de algoritmos de criptografia para cada dispositivo IPSec negociar entre si, determinando o melhor algoritmo a ser usado na conexão [27]. Esta negociação

é realizada durante o estabelecimento das associações seguras, onde são acordados os modos de proteção das mensagens IPSec. Estes acordos de segurança promovem na rede os seguintes serviços:

- **Autenticação** : capacidade de identificar o usuário aos serviços da rede;
- **Autorização** : autorizar o acesso aos serviços;
- **Controle de Acesso** : controlar o acesso aos serviços de acordo com privilégio predefinidos;
- **Confidencialidade** : Garantir a restrição aos dados da comunicação.

A implementação de segurança nas camadas TCP/IP é composta pela especificação padronizada de protocolos, interfaces e serviços. O padrão IPSec, que é desenvolvido pelo grupo de segurança do IETF [12], esta especificado em 18 RFC's, que abrangem os conceitos gerais que garantem os requisitos de segurança, definições e mecanismos necessários para a obtenção de um sistema seguro.

Toda proteção na arquitetura IPSec é definida no princípio da comunicação através das associações seguras executadas pelos dispositivos envolvidos na comunicação [27]. Estas associações são responsáveis pela determinação do processo de proteção que será realizado durante a comunicação. Podemos observar na figura 6, que a arquitetura IPSec utiliza dois protocolos de proteção que atuam diretamente no pacote IP. Os protocolos ESP (*Encapsulating Security Payload*) [26] e o protocolo AH (*Authentication Header*) [25] são utilizados para garantir vários níveis de proteção às mensagens IP na comunicação.

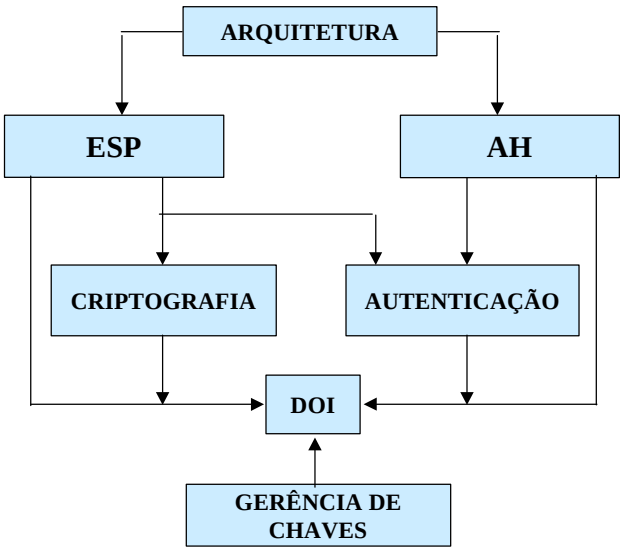

Figura 6. Arquitetura de Proteção IPSec.

As associações também determinam as chaves criptográficas, os algoritmos de criptografia e de autenticação, que são referenciados no DOI (*domain of Interpretation*), onde são armazenados os valores dos parâmetros da comunicação (vide figura 6).

O IPSec pretende suprir as vulnerabilidades do TCP/IP através da especificação dos seguintes serviços de segurança:

- Controle de acesso;
- Integridade de pacotes;
- Autenticação da origem;
- Privacidade dos pacotes;
- Privacidade em fluxo de pacotes;
- Proteção contra *replay*.

O IPSec está baseado em três protocolos que são executados pelos dispositivos antes e durante as comunicações seguras:

- IKE (*Internet Key Exchange*) [24] - protocolo híbrido, formado pelo ISAKMP [4] e pelo OAKLEY [28], que é responsável por gerar um meio seguro para a troca de informações na rede.
- AH (*Authentication Header*) - provê os serviços de autenticação, integridade

e anti-*replay*;

- ESP (*Encapsulating Security Payload*) - provê os serviços de criptografia dos dados com a opção de autenticação e anti- *replay*.

A aplicação de um determinado algoritmo de criptografia em um pacote é chamada de transformação. Durante a configuração de uma conexão podemos definir uma ou mais transformações, dependendo do grau de segurança aplicada na comunicação. Todo o tráfego de uma comunicação via IPSec é executado sob o domínio de uma associação segura (AS), que é uma entidade par-a-par e *simplex* responsável por todas as informações de controle da sessão IPSec entre dois dispositivos.

2.3.1. As Associações Seguras

A associação segura é definida como uma conexão segura entre dois nós, realizada sempre num ambiente criptografado regido segundo as regras definidas na associação. Cada regra determina uma transformação sofrida pelos dados, por exemplo, pode-se determinar que um pacote seja criptografado utilizando-se o algoritmo DES-CBC. A associação também determina qual protocolo (AH ou ESP) atuará nas conexões estabelecidas [27] , estas conexões são sempre em um único sentido, conforme a figura 7.

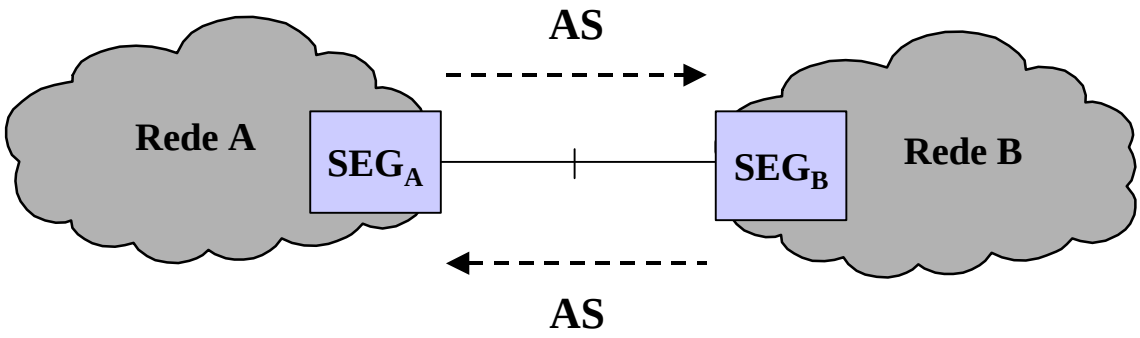

Figura 7. Associação Segura entre Redes.

O ISAKMP é o protocolo responsável pelo estabelecimento e manutenção das associações seguras, definindo a transformação aplicada nos pacotes que trafegam naquela conexão. Isto permite que em um mesmo instante seja possível estabelecer

diversas ASs entre dois nós, cada uma realizando uma transformação diferente em cada pacote.

A associação segura (AS) é uma estrutura dinâmica que somente existe enquanto houver necessidade de conexão entre as entidades que a estabeleceram. Univocamente uma AS é identificada no SAD (*Security Association Database*) através de 3 parâmetros: SPI (*Security Parameters Index*) que é uma *string* de 32 bits associada a cada AS, endereço IP do par e o protocolo de segurança utilizado (AH ou ESP).

O banco de dados SAD armazena também outras informações da AS, que são parâmetros utilizados para o estabelecimento, manutenção e finalização das associações. Estes parâmetros estão apresentados a seguir:

- Seqüência dos datagramas (32 bits);
- *Flag* de estouro (*overflow*) de seqüência;
- Janela de anti-*replay*;
- Informações do AH;
- Informações do ESP;
- Tempo de vida da AS;
- Modo de funcionamento (túnel ou transporte);
- MTU do caminho.

O IPSec pode trabalhar em dois modos de operação. O modo transporte é usado para prover segurança para comunicações fim-a-fim (cliente/servidor, duas estações de trabalho ou console de gerenciamento/dispositivo gerenciado), tem o escopo de proteção do campo de dados do IP (segmento TCP ou UDP e pacote ICMP). Já o modo túnel é usado para prover segurança para comunicações entre redes ou entre uma estação e uma rede (tipicamente aplicações de VPN), tem como objetivo a proteção de todo o pacote IP. Um novo cabeçalho IP é gerado e o cabeçalho original é incluído no campo de dados do novo cabeçalho IP. O modo túnel é mandatório se uma das extremidades da conexão for um SEG.

A tabela 1 ilustra a utilização dos protocolos do IPSec em função dos modos de operação das ASs.

Tabela 1. Quadro Comparativo das Formas de Utilização do IPSec.

Protocolos	Modo de Operação	
	Transporte	**Túnel**
AH	Autentica os dados do pacote IP.	Autentica todo pacote IP original.
	Autentica alguns campos do cabeçalho IP.	Autentica alguns campos do cabeçalho novo.
ESP	Criptografa os dados do IP.	Criptografa todo o cabeçalho IP original.
ESP com autenticação	Criptografa os dados do IP. Autentica os dados IP. Não autentica o cabeçalho IP.	Criptografa todo o cabeçalho IP original. Autentica todo o cabeçalho IP.

2.3.2. A Troca Diffie-Hellman

A troca Diffie-Hellman é utilizada durante o estabelecimento das associações seguras pelo protocolo ISAKMP. Esta troca foi desenvolvida para realizar acordos de chaves usando meios inseguros, chamado também de acordo de chaves exponenciais (*Exponential Key Agreement*), foi desenvolvido por Whitfield Diffie e Martin Hellman em 1976, publicado em um documento "As Novas Direções da Criptografia" [29].

Este processo permite que dois SEGs troquem chaves secretas em um meio inseguro através de dois parâmetros: p e g. Ambos (p e g) podem ser públicos e ser usados por todos em uma rede. O parâmetro p é um primo qualquer e o parâmetro g (chamada de chave geradora) é um número inteiro menor que p, com o seguinte requisito: para cada número n entre 1 e p inclusive, existe um expoente k de g tais que n = gk mod p.

Supondo que SEG_A e SEG_B querem concordar em uma nova chave usando Diffie-Hellman, primeiro SEG_A gera um número aleatório a, e SEG_B gera um número aleatório b. Ambos a e b estão em um conjunto de inteiros {1,...,p-2}. Então eles calculam p e g de suas chaves. O valor público de SEG_A é ga mod p, enquanto o de SEG_B é gb mod p. Agora eles trocam suas chaves públicas. Por fim, SEG_A calcula gab = (gb)a mod p e SEG_B calcula gba = (ga)b mod p. Já que gab = gba = k, então SEG_A e SEG_B tem sua chave secreta simétrica k.

2.3.3. O Protocolo ISAKMP

O protocolo ISAKMP é responsável pelo estabelecimento das associações seguras na arquitetura IPSec, sendo a base de toda segurança provida no sistema. Isto acontece porque através do processo de autenticação e criptografia aplicado pelos protocolos AH e ESP nos dados, conseguimos eliminar os riscos de ataques pela rede e garantir a identidade dos envolvidos na comunicação [27]. Entretanto estes benefícios alcançados com a utilização destas duas técnicas, esbarram na necessidade da troca inicial de parâmetros de segurança entre os nós comunicantes, o que é solucionado com o estabelecimento de associações seguras feito pelo ISAKMP. A figura 8 apresenta a o protocolo ISAKMP dentro da arquitetura de segurança IPSec.

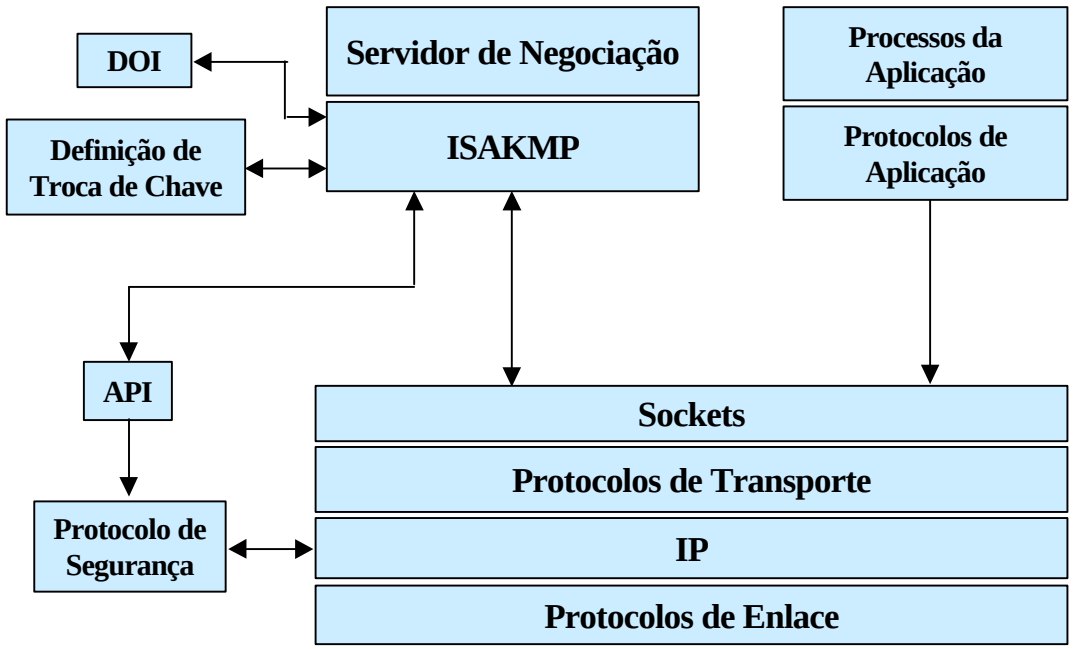

Figura 8. Arquitetura de Interação do Protocolo ISAKMP.

A criação de uma AS de controle, chamada de ISAKMP- AS é o início de toda comunicação segura entre os nós IPSec, estabelecendo um meio seguro de comunicação. Este meio seguro é obtido através de mecanismos de chave pública, cálculo de chaves simétricas, troca de *nonce* e *cookies* e a utilização do *perfect forward secrecy* que é um processo que garante que uma chave não é obtida a partir de outra.

2.3.3.1. ISAKMP-AS

O ISAKMP atua no estabelecimento, negociação, modificação e exclusão das AS, definindo o formato dos pacotes e os procedimentos de manutenção das associações. O mecanismo de troca de chaves utilizado pelo ISAKMP é uma variação mais segura do algoritmo Diffie-Hellman [29], podendo seu funcionamento ser realizado no modo de operação principal e agressivo[27].

2.3.3.1.1. Modo Principal

No modo principal os parceiros criam uma SA em 3 passos distintos, formando um total de 6 mensagens conforme a figura 9:

- No primeiro passo existe uma troca de mensagens com o objetivo de estabelecer os protocolos, algoritmos e *hashs* que serão utilizados durante esta comunicação. Este passo é chamado de proposta de comunicação;
- No segundo passo iniciado após os pares concordarem com uma proposta, os parceiros iniciam a troca de chaves, que tem como objetivo gerar chaves públicas Diffie-Hellman e *nonces* utilizados na proteção dos dados e na prevenção de ataques. A partir da troca de chaves é possível enviar mensagens criptografadas;
- No terceiro e último passo, os parceiros identificam-se mutuamente através de assinaturas digitais ou certificados de uma autoridade responsável por garantir a autenticação da comunicação.

Estes passos estabelecem um meio seguro para as mensagens trocadas nos acordos de requisitos de segurança da rede. Toda segurança depende desta primeira troca de mensagens, sendo responsável pelo acerto das chaves para a atuação dos protocolos de proteção que utilizam os algoritmos de criptografia.

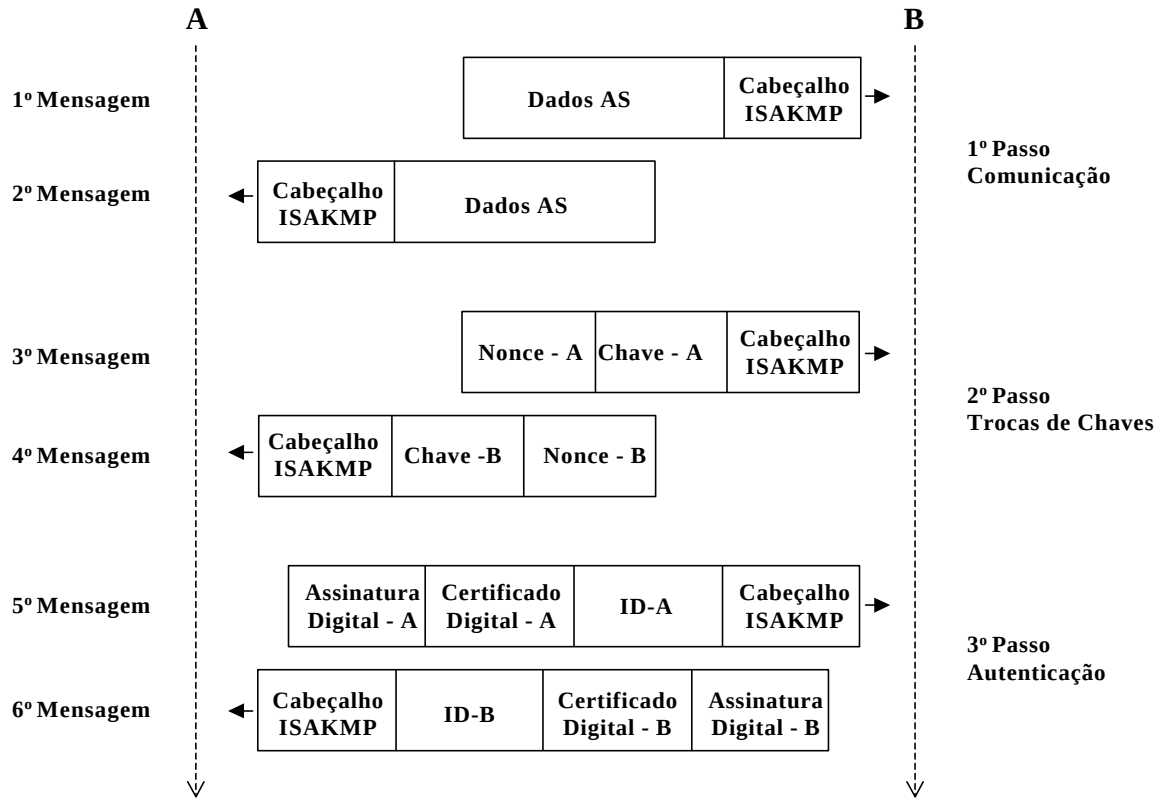

Figura 9. Mensagens da Fase Inicial em Modo Principal.

O cabeçalho ISAKMP tem formato fixo tendo um ou mais campos de dados AS. O cabeçalho armazena informações da associação segura necessárias ao processamento dos dados AS, manutenção dos seus estados e proteção contra DoS e *replay*. Os dados AS são utilizados para negociar os atributos de segurança e indicar o DOI e a situação da negociação. Em alguns casos estas trocas podem ser realizadas de forma mais simples o que é determinado pelo tipo de meio que se está utilizando. No próximo item poderemos verificar o estabelecimento da associação feita de forma mais simples através do modo agressivo.

2.3.3.1.2. Modo Agressivo

O modo agressivo tem como proposta a realização da associação segura em apenas 3 mensagens, melhorando a performance do sistema, mas com a degradação da segurança em relação ao modo principal. A figura 10 apresenta as três mensagens trocadas durante o processo de estabelecimento das associações.

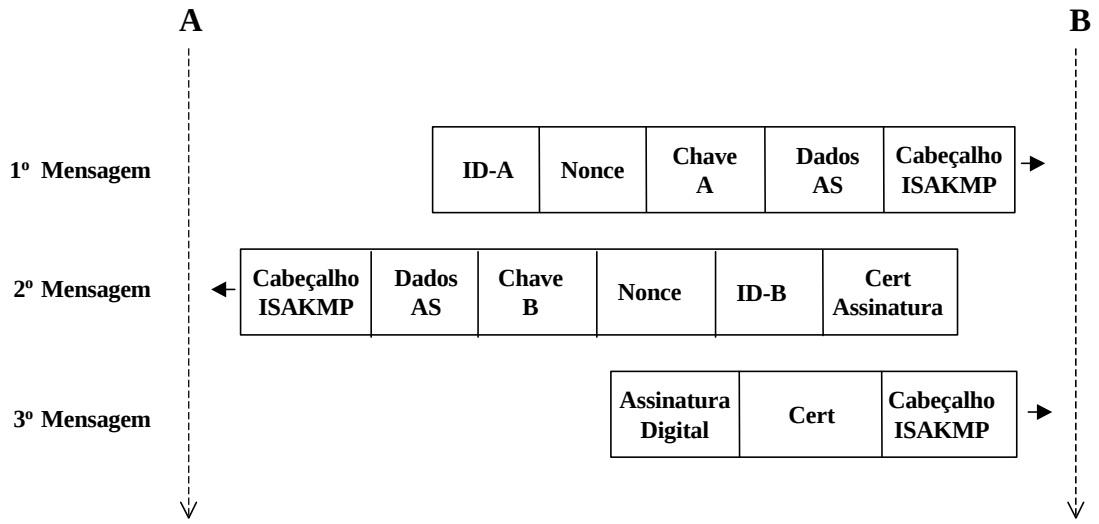

Figura 10. Mensagens da Fase Inicial em Modo Agressivo.

A primeira mensagem tem o objetivo de estabelecer os protocolos e algoritmos que serão utilizados durante esta comunicação. Na segunda mensagem há o envio da aceitação dos parâmetros já iniciando o processo Diffie-Hellman para estabelecer a chave criptográfica. A mensagem 3 é enviada a confirmação criptografada com a chave acordada. Podemos observar que no modo agressivo a agilidade do processo é priorizada em detrimento da segurança.

2.3.3.2. IPSec-AS

Após a conclusão desta primeira fase (ISAKMP-AS), podemos iniciar a troca de dados segura entre as entidades que irão solicitar a criação de uma associação AH ou ESP. Assim, serão enviadas uma ou mais propostas de transformação (algoritmos de criptografia a serem aplicados aos dados). As duas entidades devem concordar em pelo menos um conjunto de algoritmos para ser criada a AS.

A IPSec-AS trabalha em modo rápido (*quick mode*), apenas 3 mensagens são necessárias para o estabelecimento da associação segura, como apresentado na figura 11. A troca das mensagens é toda criptografada, tornando o meio totalmente seguro.

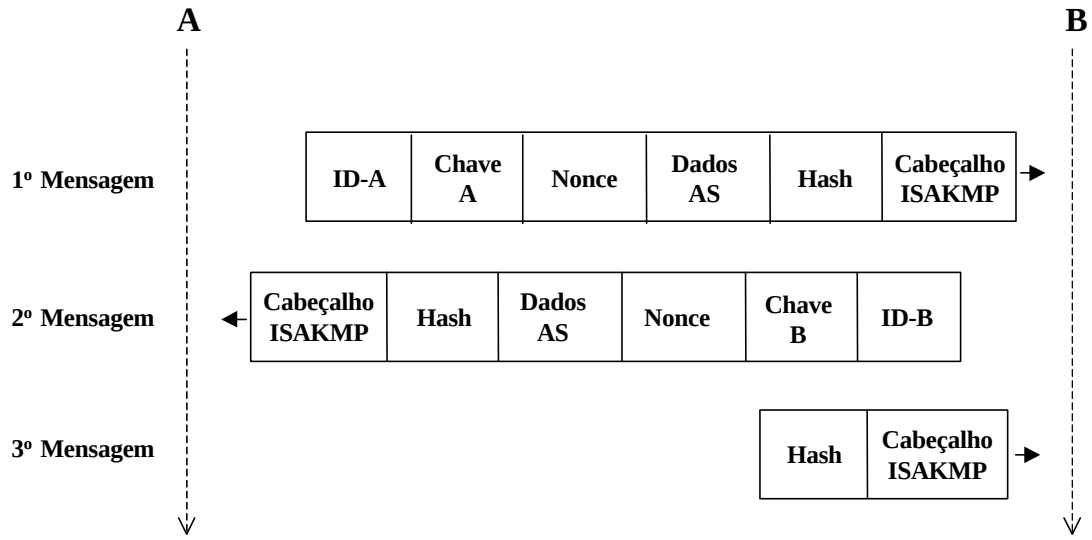

Figura 11. Mensagens da Fase Final.

A seção seguinte aborda a questão da segurança em redes de 3ª geração, apresentando a arquitetura de protocolos proposta e o modelo de comunicação utilizados para avaliar o correto funcionamento do protocolo de estabelecimento de associações seguras neste sistema.

2.4. A Arquitetura de Protocolos Proposta

Na especificação da arquitetura de segurança proposta podemos destacar o grande conjunto de alternativas, que dificulta a abordagem matemática das soluções e restringe a representação comportamental na linguagem formal. Este grande número de possibilidades torna impraticável à expressão de todas as funcionalidades, forçando a utilização de simplificações nas descrições dos protocolos da arquitetura.

As funcionalidades são representadas em cada processo pela troca de mensagens interagindo com os protocolos da arquitetura mostrados na figura 12. A modelagem dos processos respeita o funcionamento do protocolo, permitindo desta forma a correta representação dos eventos de uma seqüência comportamental. Portanto, se for garantido que o funcionamento básico do protocolo não seja violado, como por exemplo, a utilização de um processo iniciador no estabelecimento e na atualização de associações, pode-se realizar simplificações no funcionamento geral do protocolo.

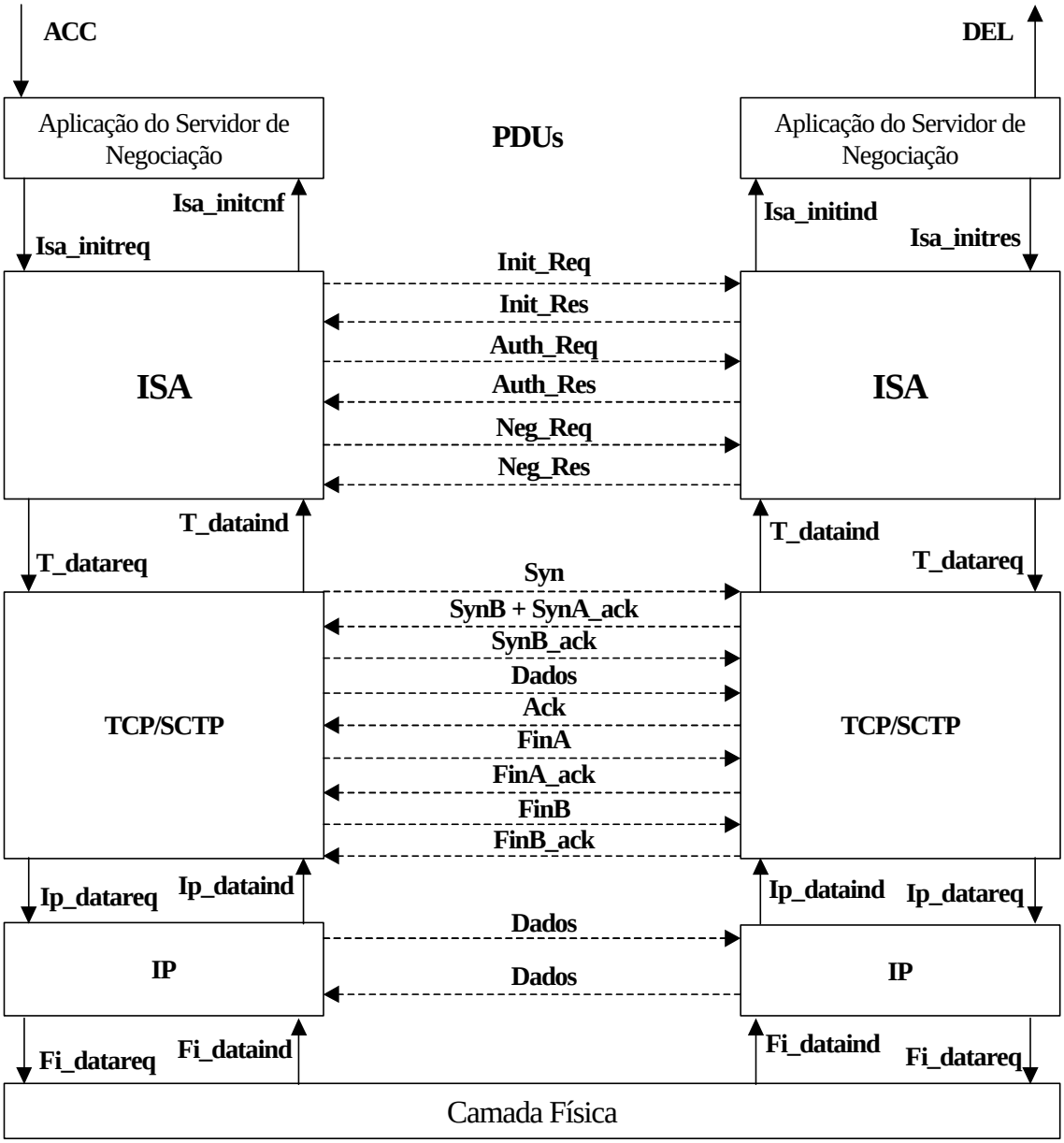

Figura 12. Estruturação dos Protocolos em Camadas

A pilha de protocolos da figura 12 representa todo o processo de troca de mensagens tanto entre camadas afins como entre camadas diferentes através de suas SDUs e PDUs. A representação da camada de aplicação destina-se a modelar a solicitação de algum "usuário", através das mensagens ACCEPT e DELIVER, que encadeiam o processo de estabelecimento das associações seguras realizado pelo protocolo ISA. As mensagens trocadas pelo protocolo ISA são enviadas através das camadas inferiores (TCP/SCTP e IP) até a camada de acesso ao meio. Todo este processo é repetido do lado do receptor.

É necessário lembrar que o diagrama da figura 12 foi apresentado para apenas duas estações, de forma a facilitar a visualização do mecanismo de funcionamento dos protocolos. Todavia no capítulo 3, na seção 3.6, mostraremos um modelo de análise comportamental baseado no número de dispositivos envolvidos na comunicação, o que permitirá verificar o comportamento dos protocolos com as trocas de mensagens simultâneas entre várias entidades.

2.5. O Protocolo ISA

O ISA é um protocolo responsável por estabelecer as associações seguras entre dispositivos de borda da rede no sistema de 3ª geração, realizando o acordo de parâmetros para a troca segura das mensagens. Este protocolo atua na camada entre a aplicação do servidor de negociação e o serviço de transporte, que pode ser provido pelos protocolos TCP ou SCTP.

A especificação do ISA leva em consideração a definição básica do ISAKMP [4], oferecendo os serviços estabelecimento, manutenção e finalização de associações seguras entre os SEGs (dispositivo de borda das rede). O ISA não está vinculado a nenhum algoritmo de criptografia, técnicas de geração de chaves ou mecanismos de segurança, tornando seu funcionamento muito flexível e de fácil adaptação a novos mecanismos e algoritmos. Esta capacidade é fundamental para a atuação rápida contra novas formas de ataques, que são desenvolvidas a todo momento.

A associação segura é um relacionamento entre duas ou mais entidades que descreve como elas utilizarão os serviços acordados para realizar uma comunicação segura. Este relacionamento é representado pela definição de parâmetros, que devem ser definidos e compartilhados de forma segura. Os atributos de cada associação são identificados pelo SPI (*Security Parameter Index*), que deve estar sempre atrelado a uma autenticação e troca de chaves.

O protocolo ISA define procedimentos para estabelecer, negociar e excluir associações seguras. As associações estabelecem todos os parâmetros de segurança necessários para a execução dos serviços seguros IP, como IP *Authentication Header* (AH) [25] *e* IP *Encapsulating Security Payload* (ESP) [26] e mecanismos de autenticação para protocolo de roteamento.

Na figura 13 apresentamos o formato do cabeçalho ISA.

Figura 13. Cabeçalho ISA.

Os campos do cabeçalho do ISA são:

- ***Cookie* - Iniciador**– *cookie* da entidade iniciadora do estabelecimento, modificação ou exclusão da AS;
- ***Cookie* - Respondedor**– *cookie* da entidade receptora do estabelecimento, modificação ou exclusão da AS;
- **Próximos dados AS** - identifica o tipo do próximo cabeçalho do pacote;
- **Versão** - atualmente tem valor 1;
- **Tipo da troca** - determina as mensagens e os dados seguintes;
- ***Flags*** – indicam as opções usadas na comunicação [4];
- **Identificação da Mensagem** – usado durante para identificar as mensagens;
- **Tamanho da mensagem** – tamanho total da mensagem (cabeçalho e dados).

O procedimento básico executado pelo protocolo ISA, pode ser definido em duas partes:

- Primeiro, uma troca inicial permite a definição básica dos atributos de segurança que serão acordados, protegendo as trocas de mensagens subsequentes. Isto indica que a autenticação e a troca de chaves foram realizadas como parte do protocolo ISA.

- Depois de acordado os atributos de segurança onde foi inicialmente autenticada a identidade e geradas as chaves requisitadas, outros atributos de segurança são trocados para estabelecer a associação completa.

Este processo é realizado por um conjunto de mensagens que define os acordos e as trocas de parâmetros. Estas mensagens estão apresentadas na tabela 2.

Tabela 2. Mensagens do Protocolo ISA

MENSAGENS DO PROTOCOLO ISA	
TIPO	**DESCRIÇÃO**
INICIAÇÃO	ISA_INIT_REQ
	ISA_INIT_RES
AUTENTICAÇÃO E TROCA DE CHAVES	ISA_AUTH_REQ
	ISA_AUTH_RES
NEGOCIAÇÃO	ISA_NEG_REQ
	ISA_NEG_RES
MODIFICAÇÃO	ISA_MODIFY_REQ
	ISA_MODIFY_RES
EXCLUSÃO	ISA_DEL
NOTIFICAÇÃO	ISA_NOTIFY

2.5.1. Estabelecimento das Associações Seguras

O estabelecimento das associações seguras é um processo de acordo e troca de parâmetros de segurança essenciais para a associação. Nas seções seguintes serão descritas as três fases que compõem o estabelecimento das associações seguras:

- Inicialização;
- Autenticação e troca de chaves;
- Negociação.

2.5.1.1. Inicialização da Associação Segura

A troca de inicialização é composta pelos pacotes ISA_INIT_REQ e ISA_INIT_RES, mostrado na figura 14. As mensagens ISA_INIT trocam *cookies*, opções de técnicas de geração de chaves, algoritmos de criptografia e mecanismos de autenticação.

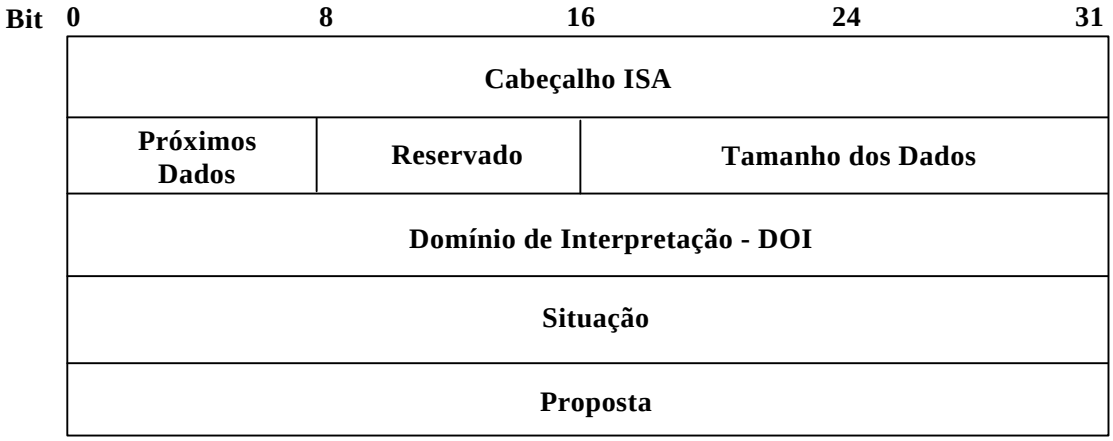

Figura 14. Pacote ISA_INIT.

Os campos do pacote ISA_INIT são:

- **Cabeçalho ISA** - apresentado na figura 13;
- **Próximos dados** - identifica o tipo dados que o pacote esta carregando;
- **Tamanho dos dados** - tamanho dos dados, incluindo propostas e transformações;
- **DOI ISAKMP** - identifica o domínio de interpretação da negociação, tendo valor 0 para negociações de ISAKMP-SA e valor 1 para IPSEC-DOI;
- **Situação** - contém as informações relevantes de segurança que o sistema considera necessárias para realizar a proteção da sessão que está sendo negociada;
- **Proposta** - contém a lista do conjunto de proteção proposta.

Os *cookies* são usados para prevenir ataques de *Replays* e *Denial of Service*. Os mecanismos de autenticação e algoritmo de criptografia são usados para autenticar e criptografar as trocas de mensagens ISA. As chaves geradas pelos mecanismos

acordados serão usadas nos algoritmos de criptografia. Estas chaves também podem ser utilizadas na troca de dados da sessão atual, na criação de novas chaves ou para proteger a troca de novas chaves para a associação.

2.5.1.1.1. Procedimentos para Inicialização das Associações Seguras

Ao receber do servidor de negociação um pedido de associação, o protocolo ISA deve enviar a mensagem de inicialização ISA_INIT_REQ. Para que isto aconteça a entidade iniciadora realiza os seguintes procedimentos:

- Cria o *cookie* do iniciador;
- Gera uma única negociação SPI pseudo-randômica;
- Determina as características relevantes de segurança para a sessão, chamada de situação;
- Gera a proposta para proteção da sessão naquela situação;
- Monta a mensagem ISA_INIT_REQ;
- Transmite a mensagem para a entidade receptora.

Quando uma mensagem ISA_INIT_REQ é recebida, a entidade receptora realiza os seguintes procedimentos:

- Checa o cabeçalho ISA para retirar os dados da mensagem;
- Determina se a situação pode ser protegida. Se não, o protocolo ISA envia a notificação de rejeição e retorna ao repouso;
- Determina se poderá usar algum dos protocolos do conjunto proposto para proteção na sessão. Se nenhum protocolo do conjunto for aceito, então o protocolo ISA envia uma notificação de rejeição, limpa todos os estados e retorna ao repouso;
- Cria o *cookie* do respondedor;
- Gera uma única negociação SPI pseudo-randômica;
- Monta a mensagem ISA_INIT_RES que contém a situação e o conjunto de

proteção escolhido;

- Transmite a mensagem a entidade iniciadora.

Quando a mensagem ISA_INIT_RES é recebida, a entidade iniciadora realiza os seguintes procedimentos:

- Checa o cabeçalho ISA para retirar os dados da mensagem;
- Verifica se a situação recebida é igual a que foi enviada. Se não, o protocolo ISA deve enviar a notificação de rejeição e possivelmente reenviar o ISA_INIT_REQ;
- Verifica se o conjunto de proteção recebida pertence as alternativas enviadas anteriormente. Se a proposta como um todo for rejeitada, é gravado o evento *PROPOSAL REJECTED* no arquivo de auditoria do sistema.
- Se for recebido um conjunto inválido de proteção, a entidade iniciadora:
  - Grava o evento *INVALID ATTRIBUTES* no arquivo de auditoria do sistema;
  - Limpa todos os estados e retorna ao repouso. Qualquer comunicação futura deverá começar o processo de inicialização do princípio.
- Se for recebido um conjunto válido de proteção, a entidade iniciadora:
  - Configura o protocolo ISA baseado no conjunto de proteção selecionado;
  - Inicia os procedimentos de autenticação e troca de chaves.

2.5.1.2. Autenticação e Troca de Chaves

A fase de autenticação e troca de chaves é realizada logo após a fase de iniciação, onde são trocadas as mensagens ISA_AUTH_REQ e ISA_AUTH_RES. Durante esta fase são definidas as informações necessárias para a identificação das entidades e estabelecimento das chaves criptográficas utilizadas na proteção da sessão. O pacote ISA_AUTH tem o formato mostrado na figura 15.

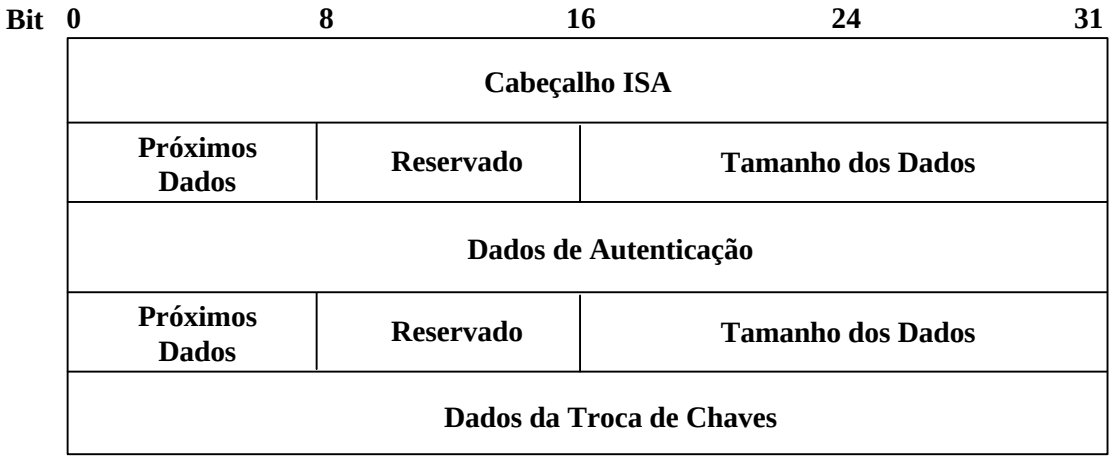

Figura 15. Pacote ISA_AUTH.

Os principais campos do pacote ISA_AUTH são:

- **Dados de Autenticação** - contém informações para a autenticação:
  - Tipo de autenticação – indica o formato dos dados de autenticação;
  - Autoridade de Autenticação – identifica o gerador dos certificados de autenticação;
  - Os certificados utilizados na autenticação;
  - Procedimento para Autenticação.
- **Dados da Troca de Chaves** - contém informações para a troca de chaves:
  - KEI – identificador de troca de chaves;
  - Algoritmo de estabelecimento de chaves;
  - Procedimento de derivação de chaves;
  - Chaves.

2.5.1.2.1. Procedimento para a Autenticação e Troca de Chaves

Após a fase de iniciação o protocolo ISA deve enviar a mensagem ISA_AUTH_REQ para autenticar as entidades envolvidas na associação e acordar as chaves que serão utilizadas na proteção das mensagens. Durante essa fase a entidade iniciadora realiza os seguintes procedimentos:

- Cria o cabeçalho ISA;
- Cria as informações de autenticação;
- Cria as informações de troca de chaves baseado no identificador de troca de chaves (KEI);
- Gera uma assinatura de autenticação, usando atributos e opções de autenticação que foram selecionadas na fase inicial;
- Transmite a mensagem para a entidade receptora.

Quando a mensagem ISA_AUTH_REQ é recebida, a entidade receptora:

- Checa o cabeçalho ISA;
- Verifica a assinatura do iniciador, processando e calculando a assinatura que é comparada com a que foi enviada na mensagem. Se estas assinaturas forem diferentes, a mensagem é descartada, o evento *INVALID SIGNATURE* é gravado no arquivo de auditoria do sistema. Nenhuma mensagem será enviada de volta, forçando a retransmissão;
- Cria as informações de autenticação;
- Cria as informações de troca de chaves baseado no identificador de troca de chaves;
- Gera uma assinatura de autenticação, usando atributos e opções de autenticação;
- Transmite a mensagem ISA_AUTH_RES para a entidade iniciadora;
- Pode-se iniciar o calculo das chaves em *background.*

Quando uma mensagem ISA_AUTH_RES é recebida, a entidade iniciadora:

- Checa o cabeçalho ISA;
- Verifica a assinatura do respondedor, processando e calculando a assinatura que é comparada com a que foi enviada na mensagem. Se estas assinaturas forem diferentes, a mensagem é descartada e o evento *INVALID SIGNATURE* é gravado no arquivo de auditoria do sistema. Nenhuma

mensagem será enviada de volta, forçando a retransmissão.

- Calcula a chave, se necessário;
- Inicia os procedimentos de negociação das associações seguras.

2.5.1.3. Negociação das Associações Seguras

A última fase do estabelecimento das associações permite a negociação de atributos de segurança entre as entidades envolvidas na comunicação. Estes atributos podem incluir opções adicionais para a realização do acordo e parâmetros para os mecanismos dos protocolos AH e ESP. Neste caso, são utilizadas as mensagens ISA_NEG_REQ e ISA_NEG_RES. O formato do pacote ISA_NEG é o mesmo do ISA_INIT, mostrado na figura 14.

2.5.1.3.1. Procedimento para Negociação das Associações Seguras

Com a autenticação e o acerto das chaves, o protocolo ISA inicia a fase final da associação através da mensagem ISA_NEG_REQ, que realiza a negociação dos parâmetros de segurança. Nesta fase a entidade iniciadora realiza os seguintes procedimentos de negociação:

- Determina os atributos que serão negociados.
- Dependendo dos atributos acordados na fase inicial, o protocolo irá aplicar os serviços de segurança acordados:
  - Se a associação necessitar de autenticação, a mensagem ISA_NEG_REQ enviará a assinatura;
  - Se a associação necessitar de criptografia, a mensagem ISA_NEG_REQ enviará a assinatura criptografada.
- Transmite a mensagem para a entidade receptora.

Quando a mensagem ISA_NEG_REQ é recebida, a entidade receptora:

- Checa o cabeçalho ISA para retirar os dados da mensagem;

- Dependendo dos atributos da associação, o protocolo irá aplicar os serviços de segurança acordados:
  - Se a associação utilizar criptografia, o protocolo irá decriptografar os dados e a assinatura. Se a decriptografia falhar, a mensagem é descartada e o evento *DECRYPTION FAILED* é gravado no arquivo de auditoria do sistema. Nenhuma mensagem será enviada de volta, forçando a retransmissão.
  - Se a associação utilizar autenticação, o protocolo irá calcular a assinatura que é comparada com a que foi enviada na mensagem. Se estas assinaturas forem diferentes, a mensagem é descartada e o evento *INVALID SIGNATURE* é gravado no arquivo de auditoria do sistema. Nenhuma mensagem será enviada de volta, forçando a retransmissão.
- Retira os dados e determina a prioridade dos atributos suportados na associação. Se nenhuma das opções dos atributos forem suportados, a mensagem ISA_NEG_RES terá valor zero e a associação não será estabelecida;
- Se a negociação da associação está requisitando uma troca de chaves ou um mecanismo de autenticação, então serão gerados as informações apropriadas e incluídas nos atributos da mensagem ISA_NEG_RES;
- Dependendo dos atributos da associação, o protocolo irá aplicar os serviços de segurança acordados;
  - Se a associação necessitar de autenticação, a mensagem ISA_NEG_REQ enviará a assinatura;
  - Se a associação necessitar de criptografia, a mensagem ISA_NEG_REQ enviará a assinatura criptografada.
- Transmite a mensagem para a entidade receptora;
- Se for necessário, começa o cálculo da nova chave de sessão em *background*;
- Retorna os dados apropriados (exemplo SPI, identificador da AS) para o servidor de negociação, limpa todos os estados e retorna ao repouso;

Quando uma mensagem ISA_NEG_RES é recebida, a entidade iniciadora:

- Checa o cabeçalho ISA para retirar os dados da mensagem;
- Dependendo dos atributos da associação, o protocolo irá aplicar os serviços de segurança acordados;
  - Se a associação utilizar criptografia, o protocolo irá decriptografar os dados e a assinatura. Se a decriptografia falhar, a mensagem é descartada e o evento *DECRYPTION FAILED* é gravado no arquivo de auditoria do sistema. Nenhuma mensagem é enviada de volta, forçando a retransmissão.
  - Se a associação utilizar autenticação, o protocolo irá calcular a assinatura que é comparada com a que foi enviada na mensagem. Se estas assinaturas forem diferentes, a mensagem é descartada e o evento *INVALID SIGNATURE*, é gravado no arquivo de auditoria do sistema. Nenhuma mensagem será enviada de volta, forçando a retransmissão.
- Retira os dados e verifica se os atributos selecionados são válidos. Se os atributos listados forem inválidos ou o receptor rejeitar todas as propostas o iniciador deve:
  - Gravar o evento *INVALID ATTRIBUTES* no arquivo de auditoria do sistema;
  - Limpar todos os estados e retorna ao repouso.
- Se for recebido atributos válidos a entidade deve configurar o protocolo ISA baseado nos atributos selecionados;
- Se for necessário, começa o cálculo da nova chave de sessão em *background*;
- Retorna os dados apropriados (exemplo SPI, identificador da AS) para o servidor de negociação, limpa todos os estados e retorna ao repouso.

2.5.2. Modificações nas Associações Seguras

A possibilidade de modificação nas associações seguras permite a atualização dos atributos e parâmetros de segurança sem a necessidade do estabelecimento de uma nova associação, provendo benefícios na performance sem o sacrifício da segurança na comunicação. O formato do pacote ISA_MODIFY é o mesmo do ISA_INIT, mostrado na figura 14, sendo os procedimentos de modificação similares aos de negociação.

2.5.3. Exclusão de Associações Seguras

A associação é excluída sempre que, por algum motivo, a comunicação esteja sendo prejudicada. Neste caso, se for realmente detectado qualquer ameaça a segurança do sistema, a associação será excluída e uma nova deverá ser estabelecida.

Antes de iniciar o processo de exclusão, a entidade informa ao par que associação segura será finalizada e enviar a mensagem ISA_DEL (vide figura 16). Esta única mensagem tem a capacidade de excluir qualquer quantidade de associações.

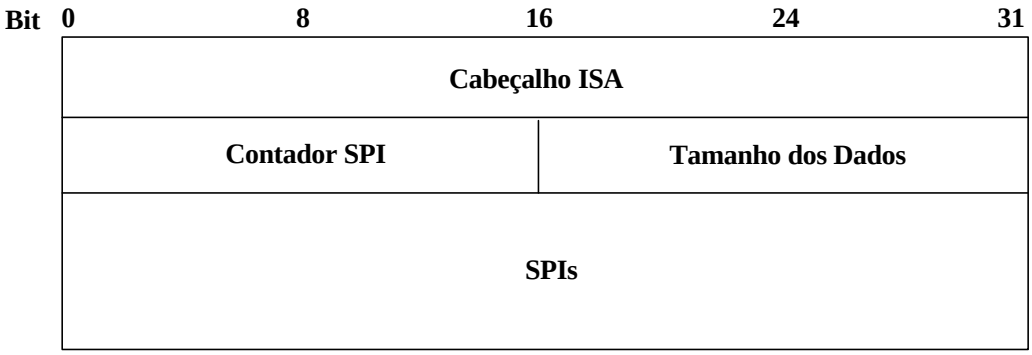

Figura 16. Formato do pacote ISA_DEL

Os campos do pacote ISA_DEL são:

- **Contador SPI** – número de associações seguras que serão excluídas;
- **Tamanho dos Dados** – tamanho dos dados em octetos;
- **SPIs** – SPIs do iniciador que serão excluídos.

2.5.3.1. Procedimentos de Exclusão

Quando é determinada a exclusão de uma associação segura a entidade iniciadora realiza os seguinte procedimentos:

- Cria o cookie iniciador;
- Determina o SPI da entidade receptora;
- Dependendo dos atributos da associação, o protocolo irá aplicar os serviços de segurança acordados:
  - Se a associação utilizar autenticação, o pacote ISA_DEL é processado e a assinatura é incluída;
  - Se a associação utilizar criptografia, os dados da mensagem ISA_DEL e a assinatura são criptografadas.
- Transmite a mensagem e atualiza o SPI no banco de dados local das associações.

Após receber um ISA_DEL, a entidade receptora:

- Checa o cabeçalho ISA para retirar os dados da mensagem;
- Dependendo dos atributos da associação, o protocolo irá aplicar os serviços de segurança acordados na seguinte ordem:
  - Se a associação necessitar de criptografia, o protocolo irá decriptografar os dados e a assinatura. Se a decriptografia falhar, a mensagem é descartada e o evento é gravado no arquivo de auditoria do sistema. A retransmissão não é realizada porque a mensagem ISA_DEL é unidirecional.
  - Se a associação necessitar de autenticação, o protocolo irá calcular a assinatura e comparar com a assinatura da mensagem. Se as assinaturas forem diferentes, a mensagem é descartada e o evento é gradado no arquivo de auditoria do sistema. A retransmissão não é realizada porque a mensagem ISA_DEL é unidirecional.
- Atualiza o SPI excluído no banco de dados local das associações.

2.5.4. Mensagem de Notificação

A mensagem ISA_NOTIFY (vide figura 17) é unidirecional e permite a comunicação entre as entidades na associação. Esta mensagem contém informações de erro que especificam o porque do não estabelecimento da associação e informações de *status* para a gerência da associação.

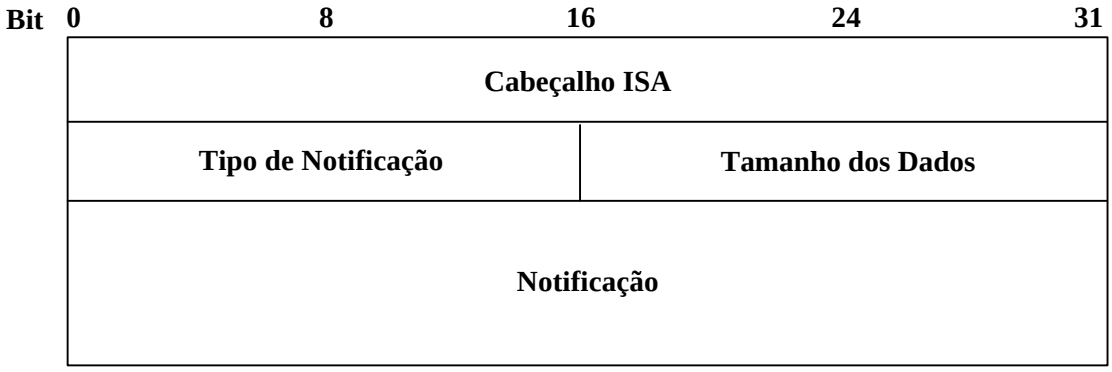

Figura 17. Formato da Mensagem ISA_NOTIFY

Os campos do pacote ISA_DEL são:

- **Tipo de Notificação** – contém o tipo de notificação;
- **Tamanho dos Dados** – tamanho dos dados em octetos;
- **Notificação** – valor dependente da notificação.

2.5.4.1. Procedimentos de Notificação

Quando for solicitado o envio de uma notificação, o protocolo utilizará a mensagem ISA_NOTIFY (vide figura 17). Para o enviar esta mensagem a entidade realiza os seguintes procedimentos:

- Cria o cookie iniciador;
- Determina o SPI do receptor;
- Dependendo dos atributos da associação, o protocolo irá aplicar os serviços de segurança acordados:
  - Se a associação utilizar autenticação, o pacote ISA_NOTIFY é

processado e a assinatura é incluída;

- Se a associação utilizar criptografia, os dados do ISA_NOTIFY e a assinatura é criptografada.

- Transmite a mensagem e atualiza o SPI no banco de dados local das associações.

Após receber um ISA_NOTIFY, a entidade receptora:

- Checa o cabeçalho ISA para retirar os dados da mensagem;
- Dependendo dos atributos da associação, o protocolo irá aplicar os serviços de segurança acordados da seguinte ordem:
  - Se a associação necessitar de criptografia, o protocolo irá decriptografar os dados e a assinatura. Se a decriptografia falhar, a mensagem é descartada e o evento é gravado no arquivo de auditoria do sistema. A retransmissão não é realizada porque a mensagem ISA_NOTIFY é unidirecional.
  - Se a associação necessitar de autenticação, o protocolo irá calcular a assinatura e comparar com a assinatura da mensagem. Se as assinaturas forem diferentes, a mensagem é descartada e o evento é gradado no arquivo de auditoria do sistema. A retransmissão não é realizada porque a mensagem ISA_NOTIFY é unidirecional.
- Dependendo da notificação, procedimentos adicionais podem ser necessários.

2.5.5. Proteção ISA

Além de serviços de segurança mencionados anteriormente o processo de funcionamento do protocolo ISA promove uma proteção contra alguns ataques como *denial of service*, conexões terroristas e ataques *Man-in-the-Middle*.

A negação de serviços pode ser evitada com a introdução de mecanismos que utilizem a troca de *cookies* para prover a disponibilidade do serviço. Os *cookies* também

podem ser utilizados para proteger recursos computacionais dos SEGs. Já as conexões com intuito terrorista podem ser evitadas pelos mecanismos de autenticação, troca de chaves e parâmetros de segurança, evitando o acesso ao sistema.

Os ataques *Man-in-the-Middle* realizam devolução de mensagens ao emissor, retransmissão de mensagens antigas e redirecionamento de mensagens com o intuito de realizarem a interceptação, inserção, modificação e exclusão de mensagens da comunicação. A máquina de estados do ISA previne este tipo de ataque pela exclusão de todas as mensagens que não sejam identificadas como de associação, limpando todos os estados e retornando à ociosidade após a exclusão.

2.6. Comentários

Neste capítulo, foram apresentadas as características principais da arquitetura de segurança de 3ª geração, respeitando os conceitos determinados pelo fórum 3GPP [3] e as funcionalidades da arquitetura IPSec [5], focando nos processos de proteção vinculados ao estabelecimento de associações seguras.

Apresentamos também a arquitetura de estudo e a descrição funcional do protocolo ISA, proposto para estabelecer as associações seguras no sistema de 3ª geração.

No próximo capítulo mostraremos a metodologia do projeto do protocolo ISA, as especificações em LOTOS das formas de estabelecimento das associações seguras e a análise do comportamento do protocolo com o aumento de associações simultâneas.

Capítulo 3

Projeto do Protocolo que Estabelece as Associações Seguras

3.1. Introdução

O projeto de um protocolo é composto de várias fases que compreendem estudo do serviço, adaptação das primitivas que interagem com protocolos de outras camadas e validação do seu comportamento. Em relação à validação, há muitas áreas do conhecimento científico onde é possível prever de forma razoável todas as hipóteses de funcionamento em um ambiente controlado. Portanto, haverá um conjunto de soluções composto de um número finito e razoável de possibilidades, levando em consideração o ponto de vista computacional.

Na verificação de qualquer protocolo o conjunto de estados e de soluções pode tender ao infinito, tornando a validação impossível. Todavia, com a realização de adaptações e simplificações podemos realizar a verificação comportamental de forma eficaz [30]. A simples verificação de propriedades essenciais, tal como *deadlocks*, é geralmente utilizada para garantir o correto funcionamento do protocolo.

Para a validação de um protocolo, antes mesmo de sua implementação, é necessário o uso de ferramentas que verifiquem as suas características em todas as hipóteses possíveis e até mesmo improváveis. Portanto, a utilização neste trabalho de técnicas de descrição baseada em métodos formais se dá pela flexibilidade alcançada na descrição das funcionalidades dos protocolos. O grau de abstração conseguido com a descrição formal é baseado em princípios matemáticos que permitem uma modelagem, verificação e análise do correto funcionamento do protocolo de forma genérica, precisa e sem ambigüidades.

No caso do protocolo de estabelecimento de associações seguras, onde não há como ter garantias dos meios de comunicação, é necessário garantir o correto funcionamento, mesmo quando ocorrem associações simultâneas. Na abordagem destes sistemas de alta complexidade, antes de tudo, é importante verificar se ele atende a propriedades básicas como, a não existência de *deadlocks*, se é vivo e inicializável, entre outras.

Neste capítulo, a seção 3.2. mostra a metodologia empregada no projeto; na seção 3.3 é apresentada a linguagem LOTOS com a biblioteca e a sua ferramenta de análise; a seção 3.4 mostra o procedimento de análise utilizado no processo de validação; a seção 3.5 apresenta o estudo do protocolo ISA em relação aos seus modos de operação; a seção 3.6 mostra a análise comportamental dos dispositivos de borda da rede no estabelecimento de associações seguras; a seção 3.7. apresenta a o processo utilizado nas simulações; na seção 3.8. são apresentados os comentários finais do capítulo, fazendo alusão aos resultados obtidos em relação as simulações do capítulo seguinte.

3.2. Metodologia

A definição da arquitetura foi o primeiro passo do estudo e levou em consideração a tendência de adaptação, para sistema sem fio, de protocolos consagrados nas redes convencionais. Já a escolha do protocolo foi determinada pela capacidade incontestável de prover, de uma forma segura, associações entre redes. Para o atendimento dos requisitos do sistema de 3ª geração, que tem como base o protocolo IP, foi escolhida a arquitetura de segurança baseada nos protocolos IPSec, que se encontra

descrita nas RFCs encontradas no grupo de segurança do IETF [12].

Após a definição dos protocolos, foi necessário encontrar uma linguagem de simulação que permitisse o uso de uma ferramentas de análise. Assim, feitas estas duas etapas foi possível partir para a modelagem da rede, das entidades e do ambiente de simulação, que dependiam da definição de parâmetros limitados pelas características do protocolo. A falha de execução do serviço prestado pelo protocolo implica no não atendimento dos requisitos de segurança da comunicação.

A descrição do protocolo em LOTOS é o ponto principal do estudo, utilizada como forma de desenvolvimento de uma arquitetura de segurança para a rede 3G. Isto só foi possível pelo grau de formalismo conseguido com esta linguagem. O protocolo foi especificado considerando que os protocolos das camadas imediatamente inferior e superior prestavam os serviços necessários e corretos, sem problemas ou erros, não comprometendo o correto funcionamento do sistema. Este processo permite que posteriormente os protocolos sejam interligados por troca de PDUs e SDUs, baseado no modelo de camadas. Desta forma, dividimos o problema em várias partes facilitando a validação.

Para os testes foi necessário que a modelagem das entidades envolvidas na comunicação fosse descrita corretamente, levando em consideração as funcionalidades básicas inerentes do protocolo, sendo capaz de garantir que seu funcionamento interno seja consistente com todos eles, simultaneamente.

Visando atender da melhor forma possível os requisitos da arquitetura foram realizadas algumas simplificações no protocolo, necessárias para melhorar a clareza do processo de descrição e simulação. Na modelagem do protocolo ISA foi considerada a análise do comportamento tendo como base a robustez de seu funcionamento para escolher a melhor forma de estabelecer uma associação segura num ambiente sem fio de 3ª geração.

No segundo passo deste estudo com a definição da melhor forma de realização de associações seguras, podemos verificar a evolução do comportamento das associações seguras com o aumento do número de entidades envolvidas. Este conjunto de análise nos dá a base para definirmos o grau de segurança alcançado por cada implementação, considerando que todos os serviços oferecidos pela camada inferior estão disponíveis e funcionam corretamente.

A seqüência das experiências seguida no processo de simulação é definida pela metodologia do projeto em 4 etapas:

- Definição da arquitetura de segurança com o protocolo adequado às características de segurança do ambiente sem fio de 3ª geração;
- Descrição do protocolo pela modelagem formal em LOTOS com passagem de parâmetros entre entidades, respeitando as características inerentes ao ambiente, para a escolha do melhor modo de estabelecimento das associações seguras;
- Aplicação do processo de análise comportamental de forma a incrementar a quantidade de associações e o número de entidades na comunicação, criando uma ambiente mais próximo do real;
- Validação do protocolo com auxílio da verificação das propriedades essenciais ao correto funcionamento, conseguindo uma modelagem final do protocolo que respeite todos os requisitos do ambiente.

3.3. Análise Formal nas Comunicações Seguras

A análise formal do protocolo nas comunicações seguras será feita utilizando a linguagem LOTOS e a ferramenta CADP. Antes de mais nada é necessária uma breve explicação da linguagem de especificação e da ferramenta de análise. Ambas utilizadas para especificações, verificação e validação aplicadas no intuito de gerar uma proposta de um protocolo adequado ao ambiente de 3ª geração.

A técnica de descrição formal torna possível a captura do comportamento funcional de sistemas [31] e, em particular, de protocolos de segurança [13]. A apropriada determinação das propriedades desejadas do sistema, bem como a sua especificação formal adequada, são essenciais à produção de documentação sem ambigüidades.

Nesse sentido, a especificação e a verificação do protocolo envolvido no processo de estabelecimento de associações seguras devem ser orientadas por técnicas de descrição formal, empregando mecanismos e linguagens apropriados. Estas técnicas formais, por serem métodos de definição do comportamento de um sistema com o uso

de uma sintaxe e de uma semântica, permitem uma implementação de protocolos sem ambigüidades, precisa e completa. Além disso, provêem uma base bem definida para a verificação e validação desses protocolos, entendidas como a avaliação de conformidade dos mesmos com relação ao comportamento esperado [6, 7].

3.3.1. A Linguagem LOTOS

LOTOS (*Language Of Temporal Ordering Specification*) é uma técnica de descrição formal padronizada pela ISO (*International Standards Organization*), utilizada na descrição formal de sistemas abertos [6]. Seu projeto foi motivado pela necessidade de uma linguagem que oferecesse alto nível de abstração sobre uma forte base matemática. Os modelos em LOTOS permitem o uso de várias técnicas de validação e verificação, e diversas ferramentas foram desenvolvidas para a automatização destes processos.

A linguagem LOTOS está baseada na premissa fundamental [6, 32] de que um sistema nada mais é do que um conjunto de processos, que trocam dados entre si e com o ambiente. A especificação do sistema deve conter a relação temporal entre as interações, descrevendo o comportamento externamente observável. A descrição na padronização LOTOS é composta por dois componentes básicos, que representam dados e os processos de interação.

Os componentes de dados de LOTOS são representados pelos domínios de valores *sorts*, que são funções matemáticas *operation*s e expressões algébricas *equations*. Estas funções agrupadas formam estruturas de dados *types*. A representação dos componentes de processos permite que sejam descritos conceitos primitivos de sistemas concorrentes (paralelo, seqüência, escolha, interrupção e etc.), podendo descrever as mais diversas interações, com base em processos fundamentados na álgebra. As interações permitem a troca de valores de dados entre os diferentes processos e o ambiente externo.

A parte de dados da linguagem é baseada na teoria de tipos de dados algébricos abstratos, mais especificamente na linguagem de especificação ACTONE [33]. Estes tipos de dados abstratos descrevem os valores que os dados podem assumir e as operações que sobre eles atuam, sem especificar como são representados e manipulados

na memória, o que contribui para a abstração inerente à linguagem [7]. Já a parte comportamental do LOTOS é baseada na álgebra de processos, combinando características das linguagens CCS, de Milner [34], e CSP, de Hoare [35].

Um sistema concorrente é descrito como uma coleção de processos paralelos interagindo por meio de *rendezvous* (pontos de encontro). O comportamento de cada processo é especificado com o uso de uma álgebra de operadores (vide tabela 1), e os processos podem manipular e trocar dados em pontos de interação denominados portas (*gates*) [6].

Tabela 3. Operadores LOTOS.

OPERADORES	INTERPRETAÇÃO
Exit	Encerramento do processo com sucesso.
P !V ?Y:T; A	Interação pela porta P, enviando um valor V e recebendo uma variável Y de valor T, com execução da ação.
Stop	Parada do processo.
A [ ] B	Executa A ou B.
A |[h,i,j]| B	Executa A e B em paralelo, com sincronização nas portas h,i,j.
Endproc	Delimitador de finalização para a estrutura do processo.
A ||| B	Executa A ou B em paralelo, sem sincronização.
Exit	Termina com sucesso.
A||B	Executa A ou B em paralelo, com sincronização.
Hide	Esconde eventos na apresentação de resultados.
P [h,i,j] (H,I,J)	Chamada do processo com parâmetros das portas h,i,j e parâmetros de valores H,I,J.

Esta linguagem foi destinada originalmente à descrição formal dos protocolos da arquitetura OSI, mas atualmente vem sendo aplicada em trabalhos de validação [20, 36, 37] de diversos sistemas complexos [38].

3.3.2. Bibliotecas em LOTOS

A biblioteca LOTOS deve definir os parâmetros que serão trocados pelas mensagens, sendo utilizada quando especificamos protocolos que necessitam de passagens de parâmetros entre as entidades. Para a sua construção devemos avaliar os tipos de dados que serão enviados nos PDUs e SDUs trocados no protocolo. O critério de escolha destes parâmetros é determinado pelo tipo de serviço prestado pelo protocolo especificado.

3.3.3. CADP (Pacote de Desenvolvimento Caesar/Aldebaran)

Para analisar o comportamento de um protocolo, empregando uma especificação LOTOS, utilizamos a ferramenta CADP (*Caesar/Aldebaran Development Package*) [7]. O CADP oferece um conjunto integrado de funcionalidades, que vão da simulação interativa até a verificação baseada em modelos. Este processo pode ser visualizado na figura 18.

As funcionalidades apresentadas pelo CADP podem ser reunidas sob três grandes grupos:

- Compilação de especificações descritas em LOTOS;
- Verificação de sistemas comunicantes;
- Validação e teste de protocolos.

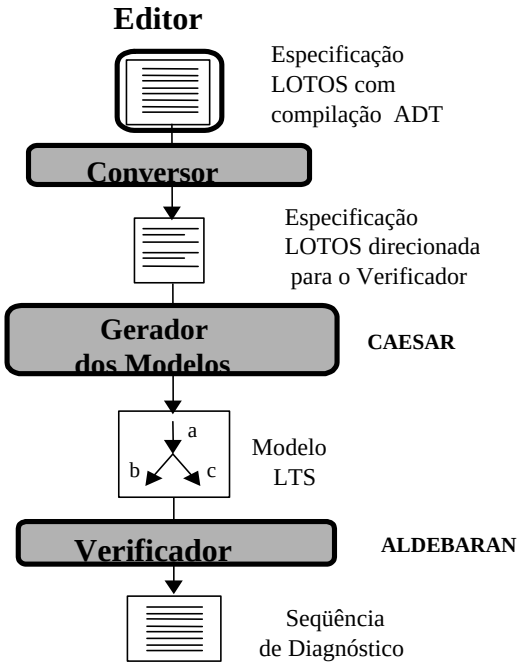

Figura 18. Processo de Análise Empregando as Ferramentas do Pacote CADP.

Na verificação de sistemas comunicantes, a partir da obtenção de sistemas de transições rotuladas, pode-se promover a verificação de várias relações de equivalência (equivalência forte, equivalência observacional, dentre outras). A validação e testes de protocolos permitem a inserção de operadores lógicos temporais. Tais operadores são aplicados sobre sistemas de transições rotuladas, possibilitando a verificação de propriedades nos protocolos especificados.

O pacote CADP [39] é um conjunto de ferramentas dedicadas a compilação, simulação, verificação e teste de descrição formal na linguagem LOTOS. Desenvolvido pela VASY, a INRIA Rhone-Alpes/Dyade e o laboratório Verimag é dividido em várias ferramentas que executam funções específicas durante a validação de um protocolo.

A ferramenta Caesar é um compilador da especificação em LOTOS para um programa em C ou em LTS para ser verificado usando as ferramentas de bissimulação e/ou avaliadores de lógica temporal. Utilizado com a ferramenta Aldebaran proporciona a verificação de sistemas de comunicação de transições de máquinas rotuladas (LTS - *Labelled Transition Systems*), possuindo ferramentas de bissimulação e avaliadores de lógica temporal que permitem comparar o LTS de um protocolo com seu respectivo serviço.

Quando o protocolo especificado em LOTOS deve trafegar dados, utilizamos a ferramenta Caesar.adt. Este compilador transforma a especificação da parte de dados, operações e parâmetros contidos na bibliotecas de tipos e funções em uma especificação em "C".

A sintaxe da descrição em LOTOS é verificada com o auxílio da ferramenta Caesar.ident, que é um programa que identifica por onde começar da forma mais fácil e coerente a leitura.

A parte gráfica da validação está reunida na ferramenta BCG (*Binary Code Graphic*). O Gráfico de Código Binário representa o sistema de transições rotuladas (LTS) criado após a verificação, através da representação binária com técnicas de compressão que pode reduzir o LTS em até 20 vezes, se comparado ao ASCII. Este pacote gráfico é composto por várias ferramentas:

- BCG_io - converte o formato BCG em diversos outros tipos;
- BCG_draw - cria o gráfico de estados e transições em duas dimensões;
- BCG_edit: - editor iterativo que permite a modificação do gráfico.

O CADP possui uma interface gráfica chamada Eucalyptus [40] que proporciona ao usuário a integração de diversas outras ferramentas, dentre elas o APERO [41] e ELUDO [42].

Durante o processo de validação as ferramentas mais utilizadas foram o Caesar, Caesar.adt e Aldebaran, possibilitando o teste das propriedades de equivalência observacional e forte [43]:

- Equivalência observacional: todo comportamento externamente observado de determinado processo pode ser igualmente realizado por uma ou mais ações de outro processo; esta relação foi utilizada para verificar se os protocolos especificados para cada camada representavam os serviços discriminados inicialmente como requisitos do projeto;
- Equivalência forte: toda ação interna de um processo deve ser igualmente realizada por uma ação interna de outro processo; esta relação foi utilizada para verificar a relação entre implementações incrementais dos diferentes protocolos.

Para o tratamento dos gráficos criados pela ferramenta BCG foi necessário utilizar o recurso de minimização que provoca a redução dos gráficos LTS e a verificação das equivalências observacional e forte, que foram utilizadas como método essencial para representar o comportamento do protocolo, retirando as redundâncias e permitindo uma análise mais simples da convergência.

3.3.4. Análise Comportamental

Uma análise do comportamento dos protocolos que promovem as garantias de segurança deve estar de acordo com o processo de validação dos requisitos determinados pelo sistema. A análise formal para protocolos adequa-se a este esforço de se atestar o comportamento dos protocolos de segurança dos sistemas de comunicação sem fio de 3ª geração. A especificação de um protocolo com o conceito de entidades confiáveis e não confiáveis torna-se viável devido à flexibilidade, em LOTOS, dos tipos de dados abstratos [20].

O processo de validação e a formalização do comportamento funcional do protocolo define uma ordem de estados que acarreta na execução da comunicação corretamente, sendo esta ordem avaliada através das propriedades que são capazes de expressar estes eventos de segurança. No entanto, este processo pode acarretar modelos infinitos, sendo necessário efetuar alguma simplificação, que torna-se viável pela limitação do número das entidades envolvidas.

A composição da especificação deve conter a modelagem dos processos de cada entidade envolvida na comunicação, descrevendo o seu exato e correto comportamento para permitir de um modo abstrato a obtenção de todos os detalhes dos mecanismos do protocolo. No caso dos protocolos de segurança, podemos modelar as funcionalidades que promovem segurança e verificar se todos os eventos relacionados com este serviço foram alcançados e executados corretamente sem *deadlocks.* Esta análise comportamental garantiria pelo menos que os mecanismos de segurança do protocolo tem o comportamento esperado e por isso conseguiriam prestar seus serviços corretamente.

3.4. Processo de Validação do Protocolo de Estabelecimento das Associações Seguras

O ponto fraco da maioria dos protocolos que promovem comunicações seguras encontra-se justamente no processo de estabelecimento e negociação das características de segurança, pois neste momento ainda não estão estabelecidos todos os serviços de segurança, tornando vulnerável todo o processo de comunicação.

A especificação dos modos de estabelecimento de associações seguras em LOTOS visa verificar o comportamento do protocolo nesta situação, através da modelagem do comportamento desejado do serviço, para que possa ser comparado com o comportamento obtido pelo protocolo especificado.

A relação entre diferentes descrições em LOTOS de um dado sistema [44] e, em particular, entre especificações do serviço e implementações do protocolo pode ser estudada usando a noção de equivalência, oriunda do CCS [34]. Essa equivalência, conhecida como observacional, é baseada na idéia de que o comportamento de um sistema é determinado pelo modo pelo qual ele interage com os observadores externos. A teoria de equivalência permite não somente provar que uma implementação está correta, com respeito a uma dada especificação, mas também substituir sistemas complexos por outros mais simples e de comportamento equivalente [6].

A outra equivalência analisada, a equivalência forte (ou bissimulação forte), também é uma ferramenta importante de análise, que é caracterizada por uma elegante definição de ponto fixo, que é muito forte na verificação de programas, não leva em consideração critérios de abstração, especialmente o conceito de ações internas ou não observáveis [7, 34]. A equivalência forte, em outras palavras, exige que toda ação interna de um processo seja igualmente realizada por uma ação interna de outro processo [31]. Como a especificação do serviço é uma modelagem do comportamento externamente observável do protocolo, não contém necessariamente as mesmas transições nem passa pelos mesmos estados internos que o protocolo modelado.

3.5. Estudo do Protocolo de Estabelecimento de Associações Seguras

O protocolo de associações seguras é a base de todo processo de acordo que determina os atributos para atendimento dos requisitos de troca de chaves criptográficas, necessário para a proteção da comunicação. Neste processo, os SEGs trocam informações para estabelecimento de associações seguras através das interfaces Za (vide figura 19).

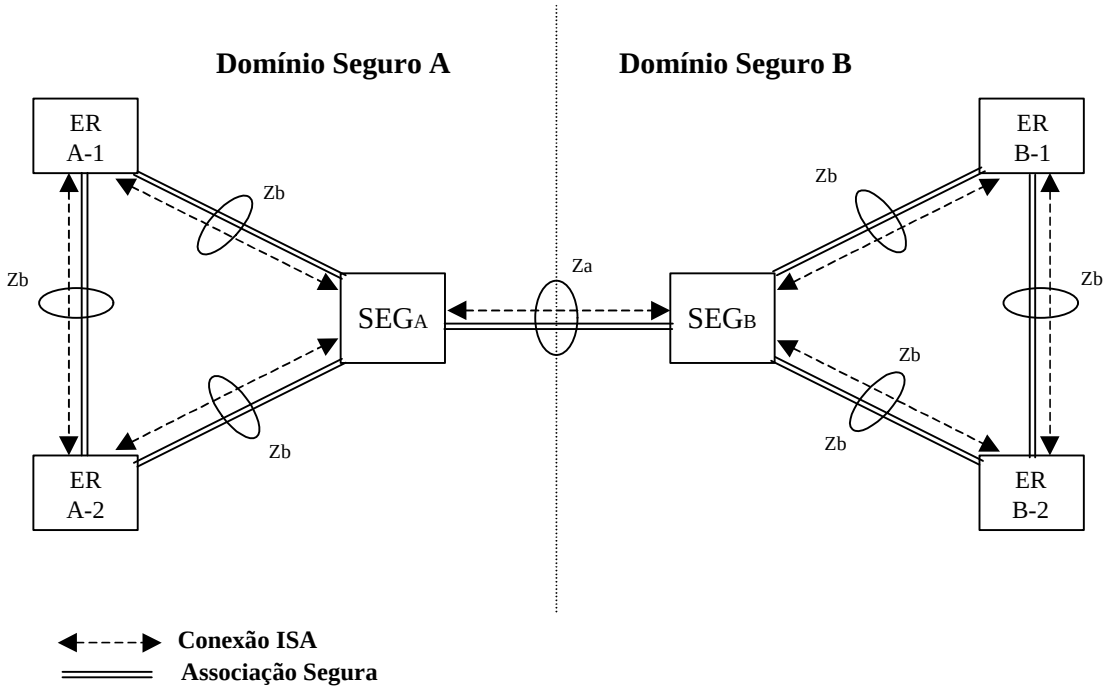

Figura 19. Entidades Envolvidas no Processo de Associação entre Domínios Seguros.

O número e a complexidade das mensagens compostas por cada estabelecimento das associações influenciam diretamente a ação de intrusos na interceptação, modificação e *replay* de mensagens. O grande problema encontrado é que, quanto maior a robustez de um sistema, mais complexa é sua implementação. Achar este equilíbrio é cada vez mais uma necessidade [13]. Neste sentido, uma análise da melhor forma de estabelecimento das associações seguras pode ser adotada, observando os dois modos de operação do protocolo ISA (agressivo e principal).

3.5.1. Especificação em LOTOS do Protocolo ISA em Modo Agressivo

Nas especificações elaboradas, foram definidos processos que implementam o protocolo, interligados por meio de comunicações simples e executados por dois roteadores de borda seguros (SEGs), estabelecendo associações seguras no modo agressivo.

Assim, conforme a figura 20, para iniciar uma negociação, o SEG_A envia na mensagem 1 o seu *cookie* C_{SEGA} e a identificação ID_{SEGA}. Inicia-se, então, a troca Diffie-Hellman [29] para estabelecer a chave criptográfica. O SEG_B aceita o tipo de criptografia, valida a identificação recebida e envia, na mensagem 2, os *cookies* C_{SEGB} e C_{SEGA} e as identificações ID_{SEGA} e ID_{SEGB}. Então o SEG_A valida a identificação e envia na mensagem 3 a confirmação criptografada com a chave acordada.

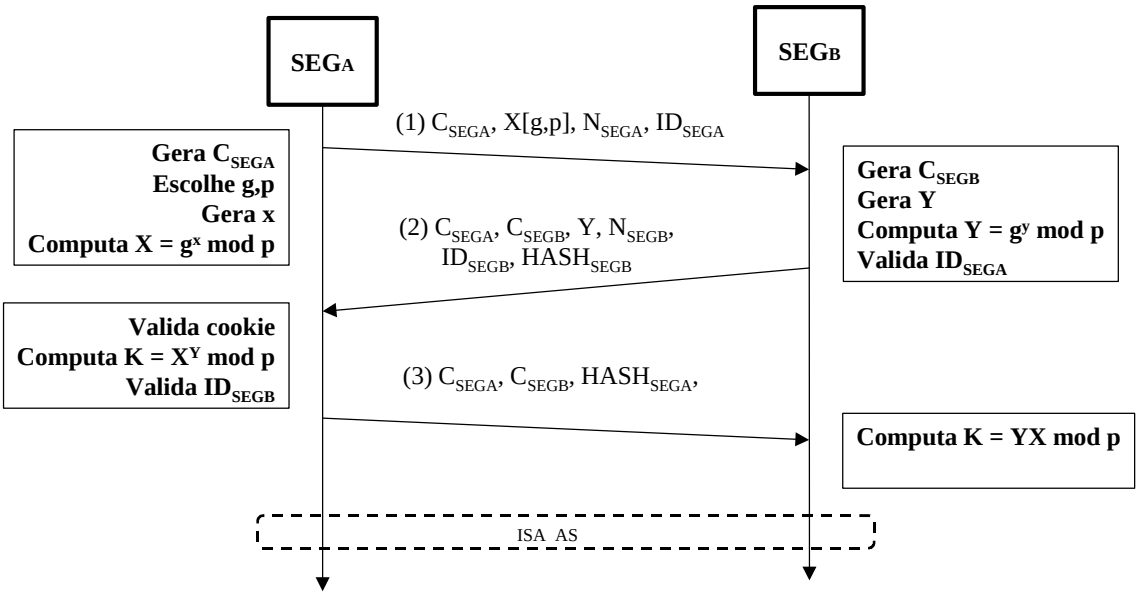

Figura 20. Modo Agressivo de Estabelecimento de Associações Seguras.

Estas mensagens foram especificadas em uma troca sincronizada que respeita a arquitetura de modelagem em LOTOS do protocolo de associação segura, descrevendo a expressão de comportamento esperado neste modo de operação. Podemos observar que a modelagem apresentada a seguir refere-se ao comportamento geral do protocolo. A especificação completa e descrição da biblioteca LOTOS encontram-se no apêndice

A e C respectivamente.

```
specification ISA_AGR [ACC, tp_datareq1, tp_dataind1, tp_datareq2,
tp_dataind2, isa_req, isa_cnf, isa_ind, isa_res, DEL] : noexit
library LIBSEG endlib
behaviour
  hide tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2, isa_req, isa_cnf,
isa_ind, isa_res   in
((USER_A [ACC, isa_req, isa_cnf, DEL
 |||  USER_B [isa_ind, DEL, ACC, isa_res])
|[isa_req, isa_cnf, isa_ind, isa_res]|
 (ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1
 |||  ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2])
|[tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]|
    TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2])
where
process USER_A [ACC, isa_req, isa_cnf, .....
```

A descrição da troca das mensagens é feita nos processos ISA_A e ISA_B onde podemos observar a especificação de cada mensagem com seu respectivo tipo de dado transmitido:

```
(* DESCRICAO TROCA DE MENSAGENS DO PROTOCOLO ISA - MODO
AGRESSIVO *)
process ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (DATA:data_type,
PDU: pdu_type) : exit :=
isa_req ?DATA: data_type;
tp_datareq1 !DATA !PDU;
   send_init_auth [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (RAS)
   where
   process send_init_auth  [isa_req, isa_cnf, tp_datareq1, tp_dataind1]
(DATA:data_type) : exit :=
   tp_dataind1  ?DATA:data_type ?PDU: pdu_type ;
   isa_cnf !DATA;
      send_auth_neg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (AUTH_NEG)
```

```
        where
        process send_auth_neg  [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PDU:
pdu_type) : exit :=
        tp_datareq1  !PDU;
        ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PAS, INIT_AUTH)
        endproc
    endproc
endproc

process ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (DATA:data_type) :
exit :=
tp_dataind2 ?DATA: data_type ?PDU: pdu_type ;
isa_ind !DATA;
    send_init_auth_res [isa_ind, isa_res, tp_datareq2, tp_dataind2](RAS,
INIT_AUTH_RES)
    where
    process send_init_auth_res  [isa_ind, isa_res, tp_datareq2, tp_dataind2]
(DATA:data_type, PDU: pdu_type) : exit :=
    isa_res ?DATA:data_type;
    tp_datareq2 !DATA !PDU ;
        send_auth_neg [isa_ind, isa_res, tp_datareq2, tp_dataind2]
        where
         process send_auth_neg  [isa_ind, isa_res, tp_datareq2, tp_dataind2] : exit
:=
         tp_dataind2 ?PDU: pdu_type;
         ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PAS)
        endproc
    endproc
endproc
process TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2] .....
```

A especificação em LOTOS utilizou o conjunto de mensagens, descritas para permitir a verificação do modelo. Foi utilizado os compiladores Caesar e Caesar.adt para gerar os LTS do protocolo e do serviço.

3.5.2. Especificação em LOTOS do Protocolo ISA em Modo Principal

Nas especificações elaboradas para o modo principal, foram definidos os mesmos processos básicos de implementação do protocolo, também interligados por meio de comunicações simples e executados por dois roteadores de borda seguros (SEGs) trocando as mensagens apresentadas na figura 21.

No modo principal, a negociação é iniciada pelo envio, no primeiro par de mensagens, dos *cookies* (C_{SEGA} e C_{SEGB}) e do tipo de criptografia suportado. Nas mensagens 3 e 4, é feita a troca Diffie-Hellman [29], que resulta em uma chave utilizada para criptografar as identidades (ID_{SEGA} e ID_{SEGB}). A confirmação é feita com o envio criptografado das identidades nas mensagens 5 e 6.

Como apresentado anteriormente, as mensagens foram sincronizadas respeitando a arquitetura de modelagem em LOTOS do protocolo de associação segura, na busca de descrever a expressão de comportamento esperado para este modo de operação.

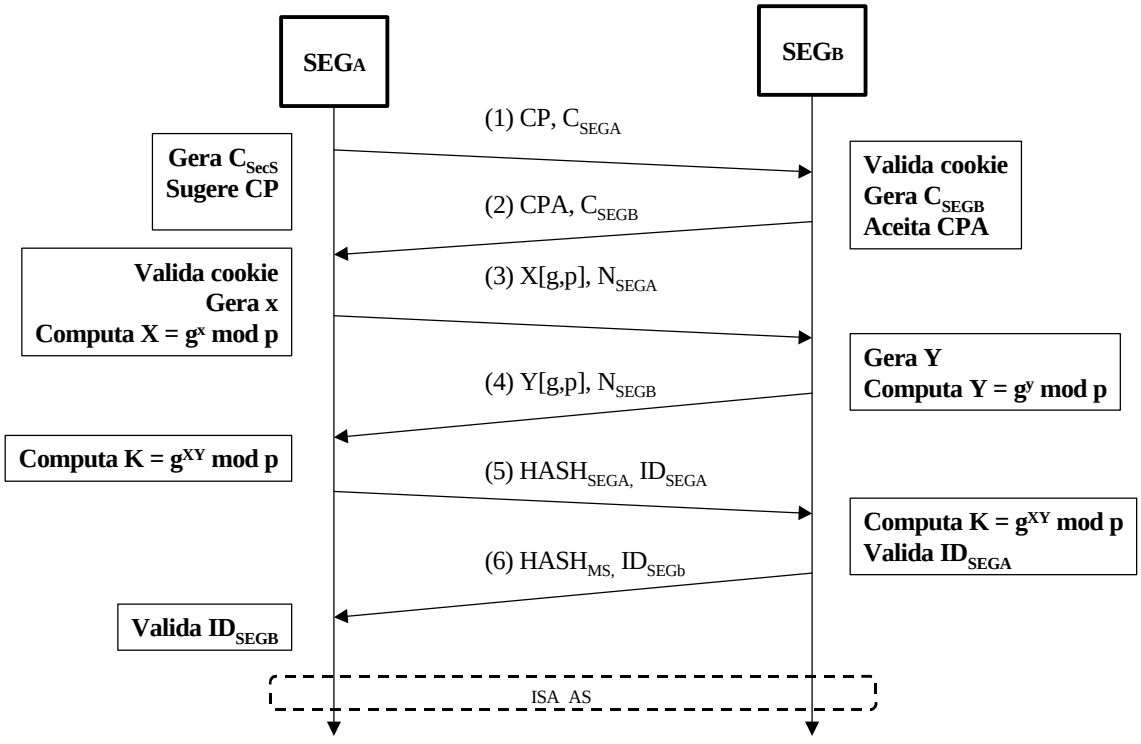

Figura 21. Modo Principal de Estabelecimento de Associações Seguras.

A especificação de interação dos protocolos nas camadas é a mesma nos dois

modos. Isto porque refere-se ao comportamento geral do protocolo. A especificação completa encontra- se no apêndice B.

```
specification ISA [ACC, tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2,
isa_req, isa_cnf, isa_ind, isa_res, DEL] : noexit
library LIBSEG endlib
behaviour
  hide tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2, isa_req, isa_cnf,
isa_ind, isa_res   in
((USER_A [ACC, isa_req, isa_cnf, DEL
 |||  USER_B [isa_ind, DEL, ACC, isa_res])
|[isa_req, isa_cnf, isa_ind, isa_res]|
 (ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1
 |||  ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2])
|[tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]|
    TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2])
where
process USER_A [ACC, isa_req, isa_cnf, .....
```

A descrição da troca de mensagens é feita nos processos ISA_A e ISA_B onde podemos observar especificação de cada mensagem com seu respectivo tipo de dado transmitido:

```
(* DESCRICAO TROCA DE MENSAGENS DO PROTOCOLO ISA - MODO
PRINCIPAL  *)
process ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PDU: pdu_type) :
exit :=
isa_req ?DATA: data_type;
tp_datareq1 !PDU;
sendauth [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PAS, AUTH)
where
  process sendauth [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (DATA:
data_type, PDU: pdu_type) : exit :=
  tp_dataind1  ?PDU: pdu_type;
  tp_datareq1 !DATA !PDU;
  sendneg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (RAS)
```

```
  where
    process sendneg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (DATA:
data_type) : exit :=
    tp_dataind1  ?DATA: data_type ?PDU: pdu_type;
    isa_cnf !DATA;
    fineg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (NEG)
    where
      process fineg [isa_req, isa_cnf, tp_datareq1, tp_dataind1](PDU: pdu_type)
: exit :=
      tp_datareq1 !PDU;
      tp_dataind1 ?PDU: pdu_type;
      ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (INIT)
      endproc
    endproc
  endproc
endproc
process ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PDU: pdu_type) :
exit :=
tp_dataind2 ?PDU: pdu_type;
tp_datareq2 !PDU;
sendinitres [isa_ind, isa_res, tp_datareq2, tp_dataind2](PAS)
where
  process sendinitres [isa_ind, isa_res, tp_datareq2, tp_dataind2]
(DATA:data_type) : exit :=
  tp_dataind2 ?DATA: data_type ?PDU: pdu_type;
  isa_ind !DATA;
  sendauthres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (RAS, AUTH_RES)
  where
    process sendauthres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (DATA:
data_type, PDU:pdu_type) : exit :=
    isa_res ?DATA:data_type;
    tp_datareq2 !DATA !PDU;
    sendnegres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (NEG_RES)
```

```
        where
          process sendnegres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PDU:
    pdu_type) : exit :=
          tp_dataind2 ?PDU: pdu_type;
          tp_datareq2 !PDU;
          ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (INIT_RES)
          endproc
        endproc
      endproc
    endproc
    process TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2] .....
```

Foram utilizados os compiladores Caesar e Caesar.adt para gerar os LTS do protocolo e do serviço. Verificando o comportamento do protocolo com o conjunto de mensagens especificadas.

3.5.3. Especificação do Serviço ISA

A arquitetura empregada para o estabelecimento de comunicações em modo seguro é descrita em LOTOS pela expressão comportamental do serviço, que é idêntica nos dois modos de operação. Servindo como base para verificação das propriedades observacionais.

```
specification isa_serv [ACC, DEL] : noexit
library LIBSEG endlib
behaviour
  Service [ACC, DEL]
where
  process Service [ACC, DEL] : noexit :=
    ACC;
      DEL;
        Service [ACC, DEL]
  endproc
endspec
```

3.6. Estudo do Comportamento do Protocolo ISA com o Aumento das Associações Seguras

Após uma análise do estabelecimento das associações seguras usando os modos agressivo e principal [45], foi necessário avaliar o comportamento dos SEGs com o aumento das associações no sistema. Para execução deste estudo foi utilizado o protocolo estabelecendo as associações seguras (ISA) no modo principal.

O processo utilizado nesta etapa visa verificar a variação do comportamento do protocolo modelado com o aumento do número de associações e entidades envolvidas. Este processo definirá se o serviço modelado será prestado mesmo em situações de alta complexidade comportamental. Durante este processo, os SEGs trocam informações simultâneas de estabelecimento de associações seguras através das interfaces Za (vide figura 22).

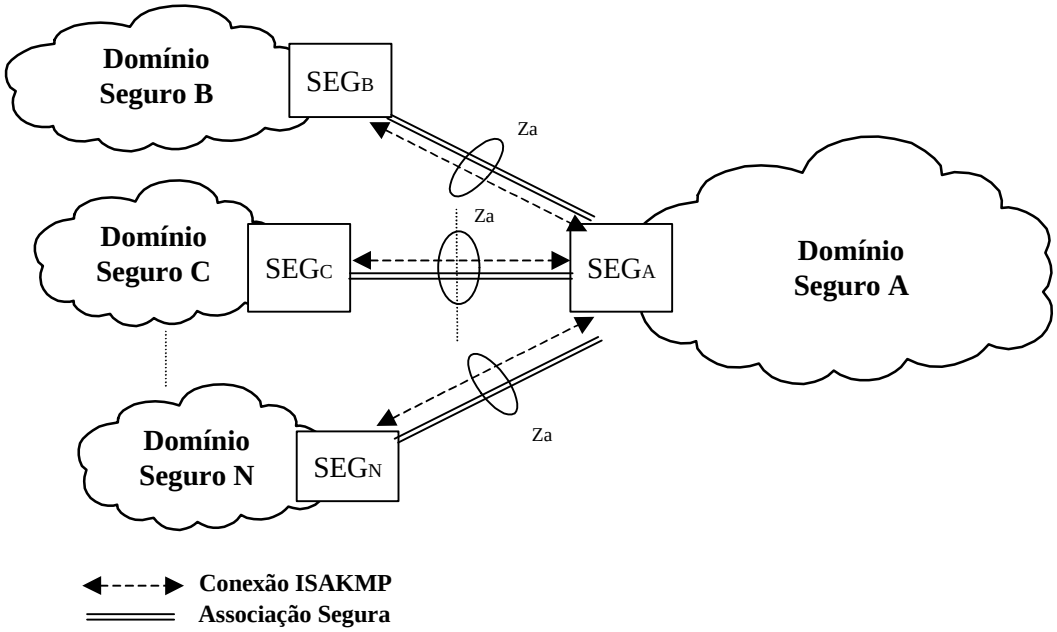

Figura 22. Entidades Envolvidas no Processo de Associação entre Domínios Seguros.

Cada SEG executará as trocas de mensagens apresentadas na figura 21, forçando a associação simultânea de vários SEGs, que é possível devido a modelagem de novas entidades para realizar simultaneamente as trocas de mensagens. Com este processo conseguimos analisar a evolução dos rótulos, estados e transições do comportamento do

protocolo modelado.

Na especificação elaborada, foram definidos processos que implementam o sincronismo entre as entidades modeladas. Todos os processos foram interligados por meio de comunicações simples e executados com 2 a 6 roteadores de borda seguros (SEGs).

A arquitetura empregada para o estabelecimento de comunicações é descrita em LOTOS pela expressão comportamental do serviço já apresentada na seção 3.5.3. A modelagem em LOTOS que representa o comportamento geral do protocolo com a interação de três SEG pode ser visto abaixo, estando a especificação completa no apêndice D.

```
specification ISA_1_2SEG [ACC, tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2,
isa_req, isa_cnf, isa_ind, isa_res, DEL] : noexit

    library LIBSEG endlib
    behaviour
      hide tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2, isa_req, isa_cnf,
    isa_ind, isa_res   in

    (((( USER_A [ACC, isa_req, isa_cnf, DEL] (PAS of data_type)
          |||
          USER_B [isa_ind, DEL, ACC, isa_res] ({} of data_type)    )
          |||
          (USER_A [ACC, isa_req, isa_cnf, DEL] (PAS of data_type)
          |||
          USER_C [isa_ind, DEL, ACC, isa_res] ({} of data_type)
          ))
         |[isa_req, isa_cnf, isa_ind, isa_res]|
        (ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (INIT of pdu_type)
          |||
          ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (INIT_RES of pdu_type)
    ))
        |[tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]|
         TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]....
```

A seguir, apresentamos um pequeno fragmento da especificação LOTOS, contendo a troca de mensagens entre os SEGs na execução do protocolo:

```
process SEG_A [ACC, isa_req, isa_cnf, DEL] (DATA: data_type) : noexit :=
    ACC; (* Aplicacao recebe um pedido de iniciacao da AS *)
    isa_req !DATA; (* ISA recebe um pedido de iniciacao da AS da aplicacao *)
    isa_cnf ?DATA: data_type; (* ISA envia a confirmacao da AS para a
aplicacao*)
    DEL; (* Aplicacao recebe uma confirmacao da AS *)
    SEG_A [ACC, isa_req, isa_cnf, DEL] (PAS) (* retorna ao estado inicial *)
endproc

process USER_N [isa_ind, DEL, ACC, isa_res] (DATA: data_type)  : noexit :=
    isa_ind ?DATA: data_type; (* Aplicacao recebe a solicitacao de iniciacao da
AS do ISA *)
    DEL; (* Aplicacao envia a solicitacao de iniciacao *)
    ACC; (* Aplicacao recebe a solicitacao de iniciacao *)
    isa_res !DATA; (* ISA recebe a resposta da AS *)
    SEG_N [isa_ind, DEL, ACC, isa_res] (RAS)(* retorna ao estado inicial *)
endproc
```

3.7. Processo de Simulação do Protocolo ISA

Nas primeiras simulações do protocolo, onde buscávamos a melhor forma de implementar o protocolo de associações seguras, conseguimos verificar que o protocolo ISA, trabalhando nos modos agressivo e principal obteve modelos de serviço adequados em relação ao comportamento básico esperado. Nos dois modelos foi possível verificar o atendimento das equivalências observacionais. Atestando o correto funcionamento do protocolo especificados.

Os modelos também foram submetidos à verificação de equivalência forte, através da ferramenta Aldebaran, o que resultou no não atendimento desta equivalência. Este resultado já era esperado, devido a não descrição das ações internas na

especificação do serviço.

Todos testes tinham o objetivo de verificar a não existência de *deadlocks* no protocolo e garantir que era vivo e reinicializável. Estas condições são essenciais para o desenvolvimento do projeto.

O procedimento utilizado nos testes seguiu a mesma linha em todas as fases do processo de validação. Foram testadas todas as propriedades e condições comportamentais baseadas no serviço oferecido, visando o atendimento gradual dos requisitos do ambiente 3G. Modificações nas mensagens foram sendo feitas a cada condição satisfatória alcançada. Este processo dinâmico de remodelagem do protocolo foi importante para criar uma variedade de modelos adaptados a cada nova situação testada.

Com as simulações conseguimos um novo fator para atestar a escolha do modo de operação do protocolo, possibilitando constatar com a análise das transições e estados que a complexidade das trocas de mensagens no modo principal supera a do modo agressivo, o que deve acontecer em um protocolo dito seguro, devido ao grau de dificuldade que ele exerce sobre tentativas de interceptação e retransmissão de suas mensagens [45].

A otimização alcançada na proposta do estabelecimento de associações seguras em modo agressivo tende a ser mais vulnerável aos ataques de *replays* (retransmissões de mensagens antigas) e a ataques de *Denial of Service* [46], prejudicando a sua utilização em ambientes com risco agregado de invasões como o 3G. Com a utilização das trocas de mensagens de associação em modo principal entre os SEGs obtém-se um acréscimo na complexidade das mensagens do protocolo, justamente o que se busca nesta análise.

Para a arquitetura proposta neste trabalho, a idéia de se utilizar o protocolo de associações seguras em modo principal atende aos requisitos de segurança definidos em [9], tornando a troca de mensagens mais complexas. Portanto a troca dos parâmetros na primeira fase em modo agressivo só seria utilizada em ambientes mais seguros, o que não acontece no sistema 3ª geração.

Após a determinação do modo de operação do protocolo ISA podemos nos concentrar na validação do seu comportamento real. Este processo foi realizado com o aumento do número de associações simultâneas e entidades envolvidas nas simulações

do protocolo ISA.

3.8. Comentários

O capítulo explicou o processo utilizado para verificar a melhor forma de funcionamento do protocolo de associações seguras, apresentando as especificações formais do protocolo, seus respectivos serviços conforme os modos de operação e a influência do número de associações no seu comportamento. Estes modelos foram submetidos a testes para verificar as propriedades de equivalência observacional, equivalência forte e garantir se eram vivos e reinicializáveis, em relação à possibilidade de haver *deadlocks*, etc.

A validação destas condições essenciais de funcionamento do protocolo foram descritas no prosseguimento dos experimentos. Estes testes e procedimentos foram feitos em todas as fases do projeto do protocolo. Em cada experimento, quando havia um incremento de complexidade, eram testadas todas as propriedades e condições anteriormente mencionadas, de forma a garantir o aumento da complexidade apenas a partir de tais condições satisfeitas.

Com o decorrer das diversas simulações, foram feitas melhorias que eram visualizadas através dos resultados alcançados no comportamento de seus gráficos de estados e transições. Em cada versão terminada, os atributos do protocolo, parâmetros e tratamento de exceções foram criando um modelo completo de situações reais de validação [47]. Com isto, esta série de experimentos culminou com a especificação e verificação de modelos completos de nós, que apresentam transparência em relação ao meio, independência diante dos demais nós da rede, sendo dotados de todas as funcionalidades de transmissão e recepção requerido por um dispositivo de borda da rede 3G. No próximo capítulo, destinado aos resultados, poderemos verificar as conclusões obtidas com as simulações.

Capítulo 4

Verificação e Simulação do Protocolo ISA

4.1. Introdução

A verificação e simulação do protocolo responsável pelo estabelecimento das associações seguras no sistema de 3ª geração serão apresentadas neste capítulo. Descreveremos os experimentos realizados com o protocolo ISA e suas especificações mostradas no capítulo anterior.

A busca de uma especificação que refletisse da forma mais real o funcionamento do protocolo obrigou à utilização de um processo gradual de descrição destas funcionalidades. Este processo de validação foi organizado de forma a obter, primeiramente uma descrição simples e genérica, contando somente com as características básicas do funcionamento, possibilitando posteriormente a inclusão de funcionalidades mais específicas.

A obtenção dos resultados de cada simulação proporcionou mecanismos mais eficazes para implementação do aumento da complexidade de funcionamento. Esta complexidade refere-se aos processos internos de descrição das troca de mensagens executadas nas especificações dos nós (SEGs) e ao aumento do número destes na rede.

O processo de aumento do detalhamento funcional tornou as especificações cada vez mais completas, culminando em uma validação eficaz do protocolo com relação ao sistema de 3ª geração.

Neste capítulo, a seção 4.2. apresenta a verificação e simulação do protocolo de associações seguras com a escolha da forma de funcionamento; na seção 4.3 apresentamos a verificação e simulação do comportamento do protocolo com o aumento do número de associações seguras estabelecidas; a seção 4.4 apresenta os comentários finais sobre as simulações.

4.2. Simulações do Protocolo ISA

As simulações com o protocolo ISA seguem uma seqüência de lógica incremental de funcionalidade, apresentando primeiramente a descrição básica do protocolo levando em consideração os mecanismos originais [4], e a partir daí entrando em um processo de incremento gradual de funcionalidades específicas, alcançando uma especificação completa e coerente para o sistema de 3ª geração.

O protocolo ISA foi descrito em LOTOS, com dois usuários, apresentando alternativas quanto ao modo de operação executado durante o processo de estabelecimento das associações. As descrições das fases de projeto foram executadas numa seqüência lógica e cronológica dentro da filosofia apresentada anteriormente.

A modelagem na linguagem LOTOS permitiu que as trocas de mensagens entre os dispositivos em um meio de comunicação fossem alteradas de forma eficaz, graças ao grau de abstração conseguido com o seu uso.

A ferramenta CADP [39] foi de grande importância para a produção dos resultados que possibilitaram definir a forma ideal de funcionamento do protocolo modelado. As mensagens e campos necessários para uma comunicação eficiente foram simuladas, tornando este processo de incremento gradual e dinâmico. Estas simulações se detiveram em determinar a melhor forma de estabelecimento sem se preocupar neste momento com variações de comportamento derivadas da mudança do número de dispositivos na rede. Este estudo será tratado na seção 4.3.

4.2.1. Simulação com o Protocolo ISA em Modo Agressivo

A simulação do protocolo ISA em modo agressivo segue o processo de determinação do funcionamento do protocolo. A evolução do comportamento nesta modalidade serviu como base às fases de especificação do protocolo. As conclusões serão apresentadas no decorrer do texto com auxílio de diagramas e gráficos sendo evidenciadas no final do capítulo.

As simulações começaram com troca de mensagens básicas utilizando parte de sua estrutura original. A primeira simulação foi feita com dois SEGs, um transmissor e um receptor. Através do estudo e desta simulação foi possível verificar a viabilidade do funcionamento do protocolo e a necessidade de mais simulações para determinar a eficácia da prestação do serviço neste modo de operação. Podemos observar na figura 23 a rede de *Petri* do modo de operação agressivo do protocolo.

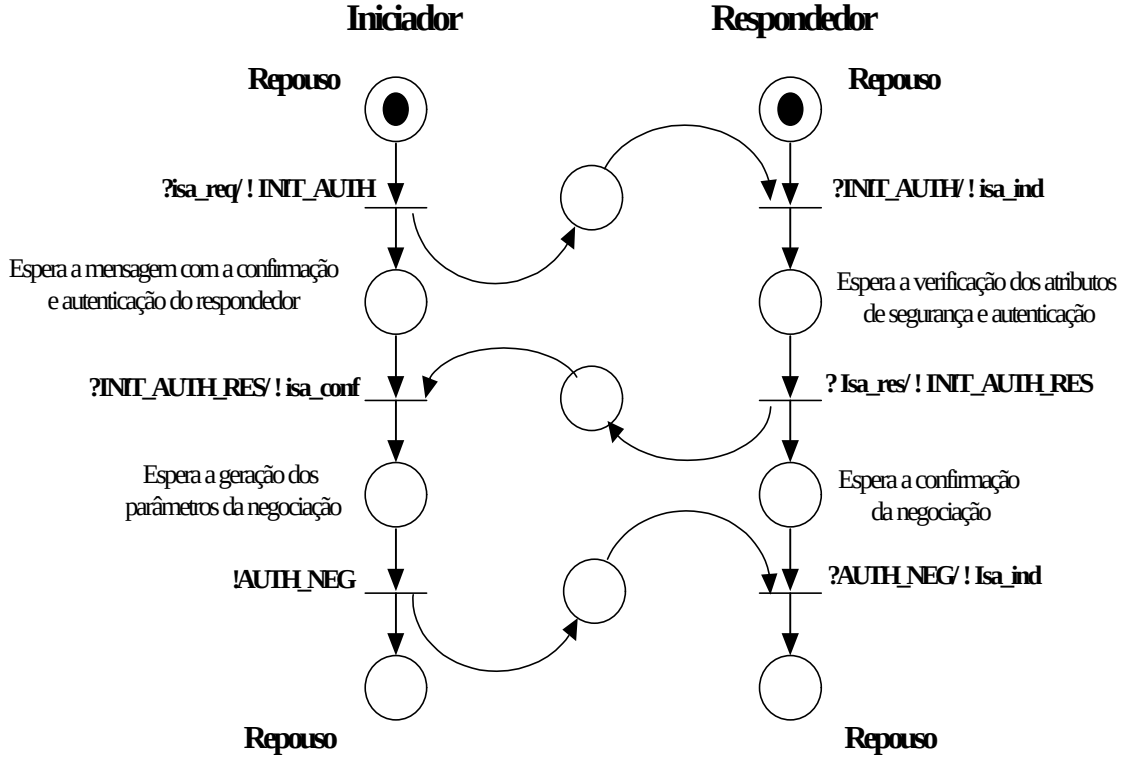

Figura 23. Rede de Petri do Protocolo ISA em Modo Agressivo.

Podemos observar os vários processos na figura 24, com suas respectivas trocas de mensagens. Estes processos mostram a relação das mensagens com o comportamento do protocolo. As mensagens são trocadas entre os SEGs através do processo TP, que representa a camada inferior do protocolo de associação segura. Os processos $USER_A$ e $USER_B$, representam os usuários das camadas superiores, que utilizam os serviços do protocolo ISA_A e ISA_B. Esta interação pode também ser observada na rede de *Petri* apresentada anteriormente.

O protocolo foi modelado em LOTOS seguindo a processo de comunicação a seguir:

- Ao receber uma mensagem da camada superior, o processo $USER_A$ solicita os serviços da camada ISA;
- Os processos ISA_A e ISA_B representam o protocolo de associações seguras dentro de cada SEG, funcionando de forma seqüencial e sincronizada;
- Os processos internos ao ISA_A e ISA_B são encadeados e interagem seqüencialmente até a última mensagem;
- Quando ISA_A recebe DATA, ele envia a ISA_B os parâmetros de negociação suportados, sua identificação e a iniciação do acordo Diffie Hellman.
- O ISA_B recebe os parâmetros de ISA_A e envia o DATA para o $USER_B$
- Quando o ISA_B receber a resposta de confirmação de negociação do $USER_B$, confirma os componentes da troca Diffie Hellman e envia as informação de confirmação e a sua identificação para o ISA_A;
- O ISA_A valida a identificação e envia a confirmação e os parâmetros acordados na a negociação criptografados para o ISA_B;
- A comunicação entre os SEGs se encerra, ISA_B volta a seu estado inicial.
- O ISA_A, após confirmar o êxito da comunicação para o $USER_A$, também volta a seu estado inicial e o canal está livre para uma nova comunicação.

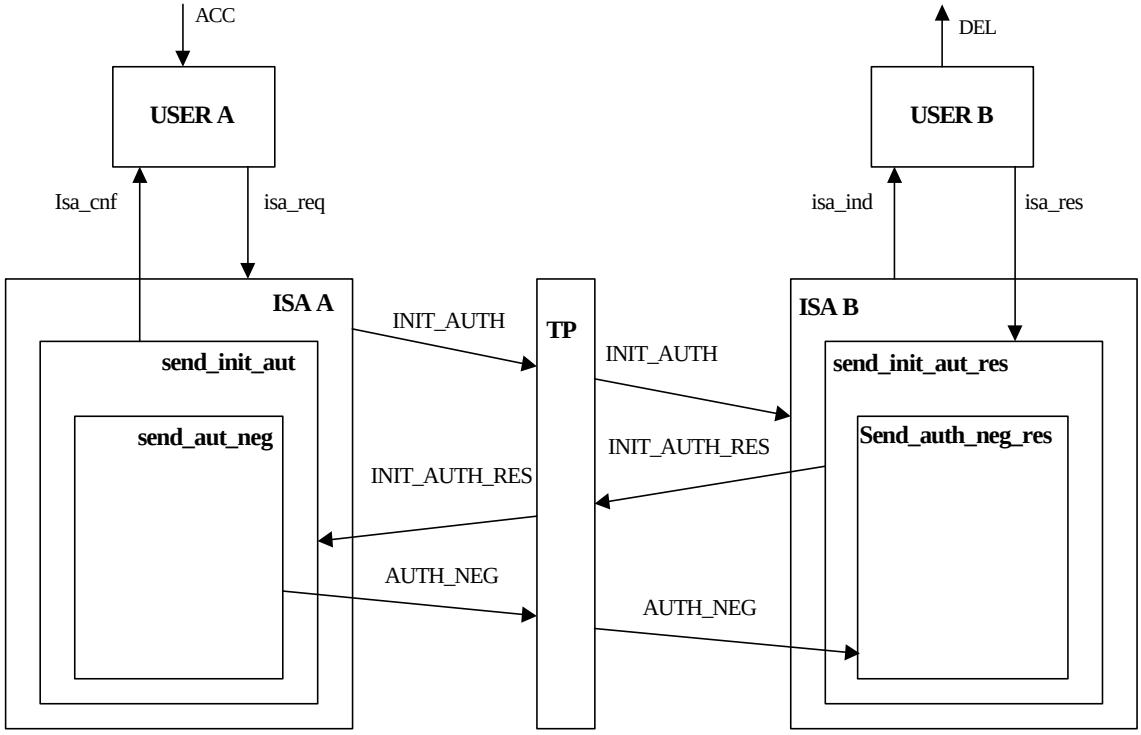

Figura 24. Processos do Protocolo ISA em Modo Agressivo.

Os resultados das simulações e alterações do protocolo nesta modalidade de estabelecimento levaram à criação do gráfico de estados e transições da figura 25. Este gráfico foi obtido pelas ferramentas do CADP, através da especificação formal em LOTOS do protocolo de associações seguras em modo agressivo, contendo 20 estados e 25 transições.

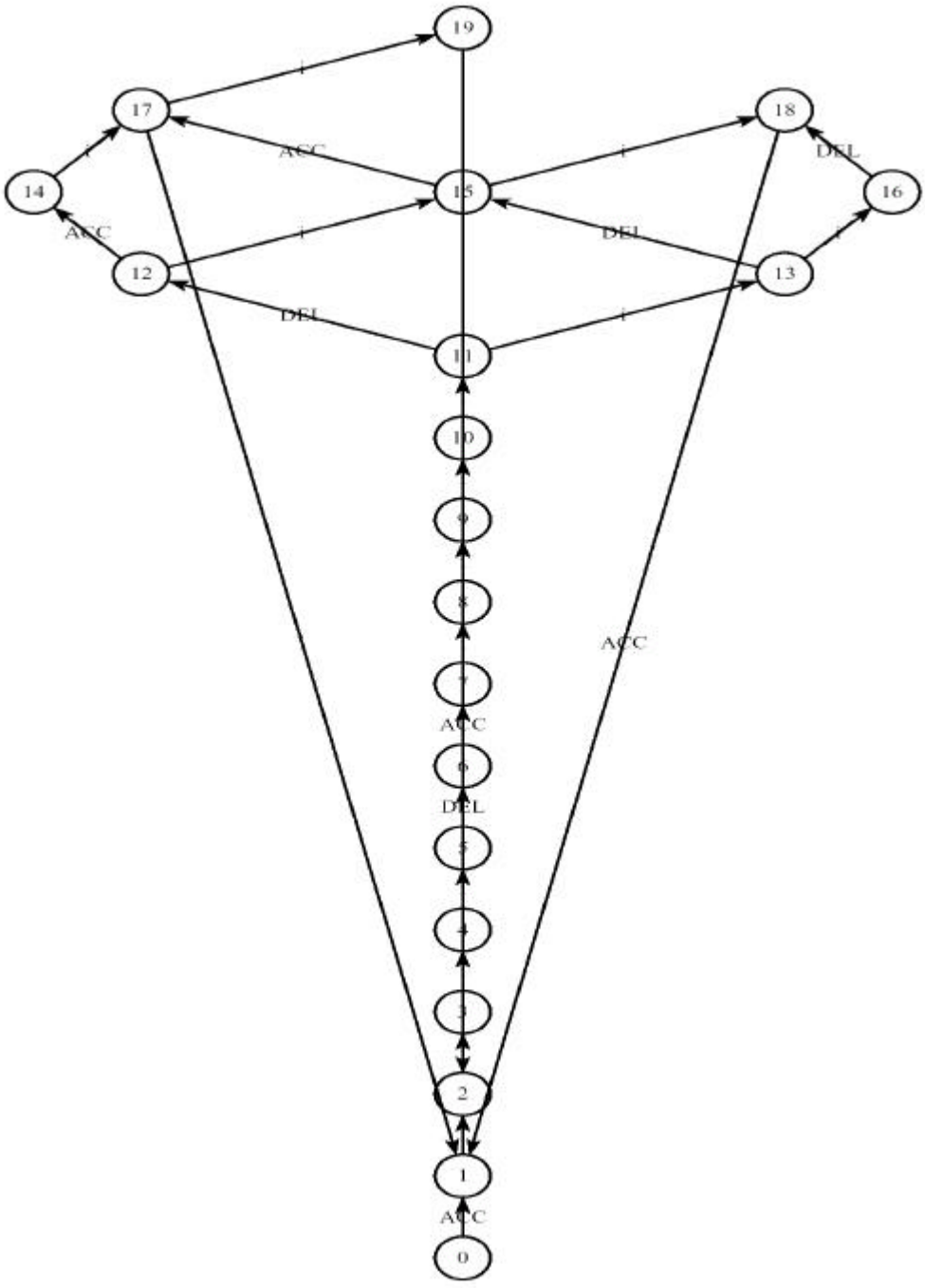

Figura 25. Gráfico de Estados e Transições do Protocolo ISA em Modo Agressivo.

O gráfico 26 apresenta o número de estados e transições minimizado para o protocolo em modo agressivo. Este recurso é utilizado para retirar os estados e transições redundantes que possuem mesmo comportamento, mesma origem e mesmo destino. A utilização desta minimização possibilitou uma análise mais qualitativa dos resultados, deixando o gráfico com 18 estados e 22 transições.

Os gráficos demonstram a coerência da descrição formal com a existência de poucos estados redundantes e a confirmação das propriedades observacionais.

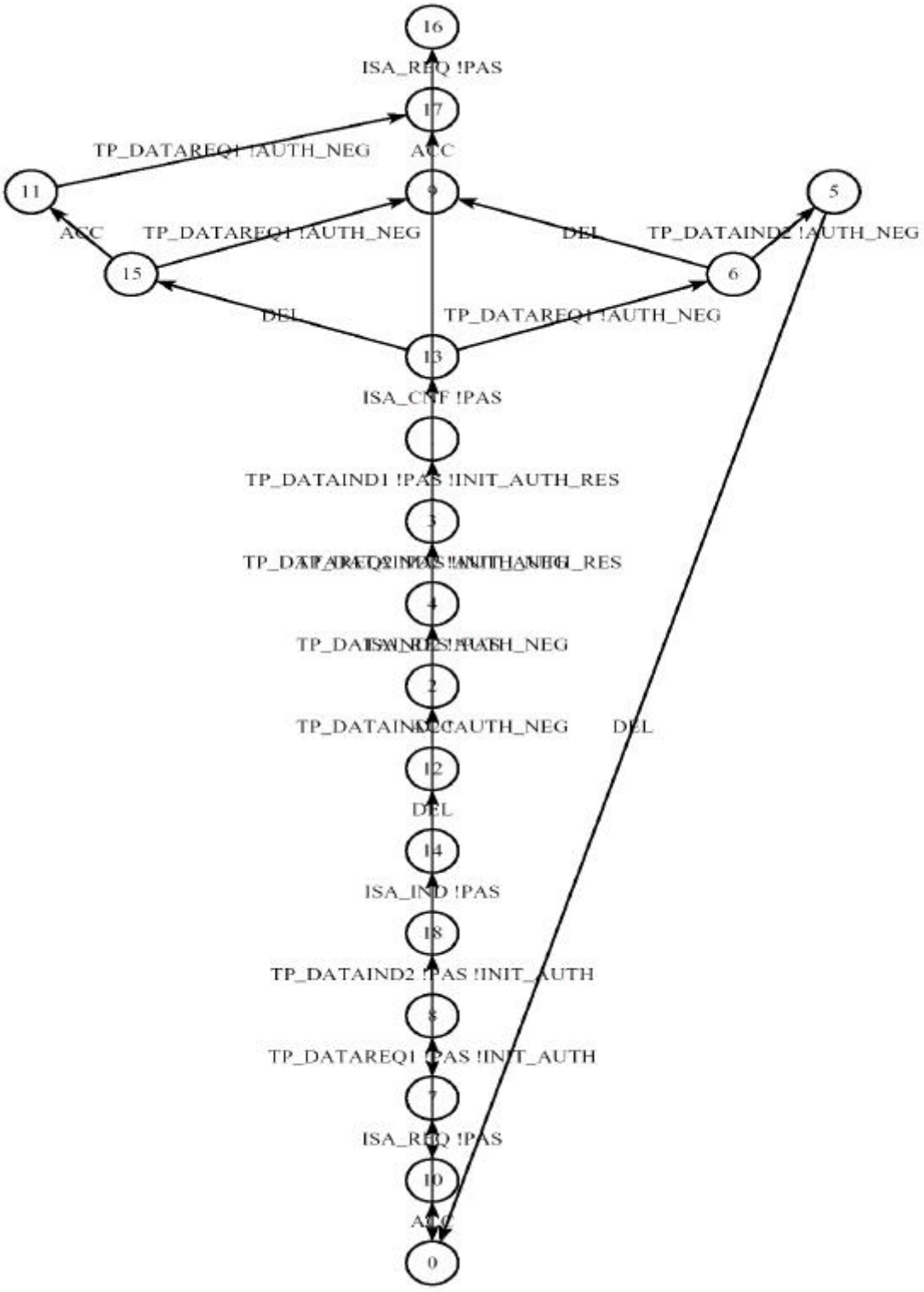

Figura 26. Gráfico Minimizado de Estados e Transições do Protocolo ISA em Modo Agressivo.

Nas simulações que resultaram nesta última especificação foram feitos testes de lógica inter processos, entre as mensagens. Estas estruturas de decisão de envio e recebimento de mensagens foram testadas tendo como objetivo verificar a possibilidade de haver falhas como, por exemplo, "*deadlocks*".

4.2.2. Simulação com o Protocolo ISA em Modo Principal

Na primeira simulação do protocolo trabalhando em modo agressivo são apresentados, sucintamente, os mecanismos de estabelecimento das associações, através dos processos INIT_AUTH e AUTH_NEG, respectivamente. Já em modo principal estes processos são divididos e originam novas mensagens deixando a estrutura de comunicação mais clara e segura. As trocas agora são monitoradas de acordo com a mensagem recebida. Podemos observar na figura 27 a rede de *Petri* do modo de operação principal do protocolo.

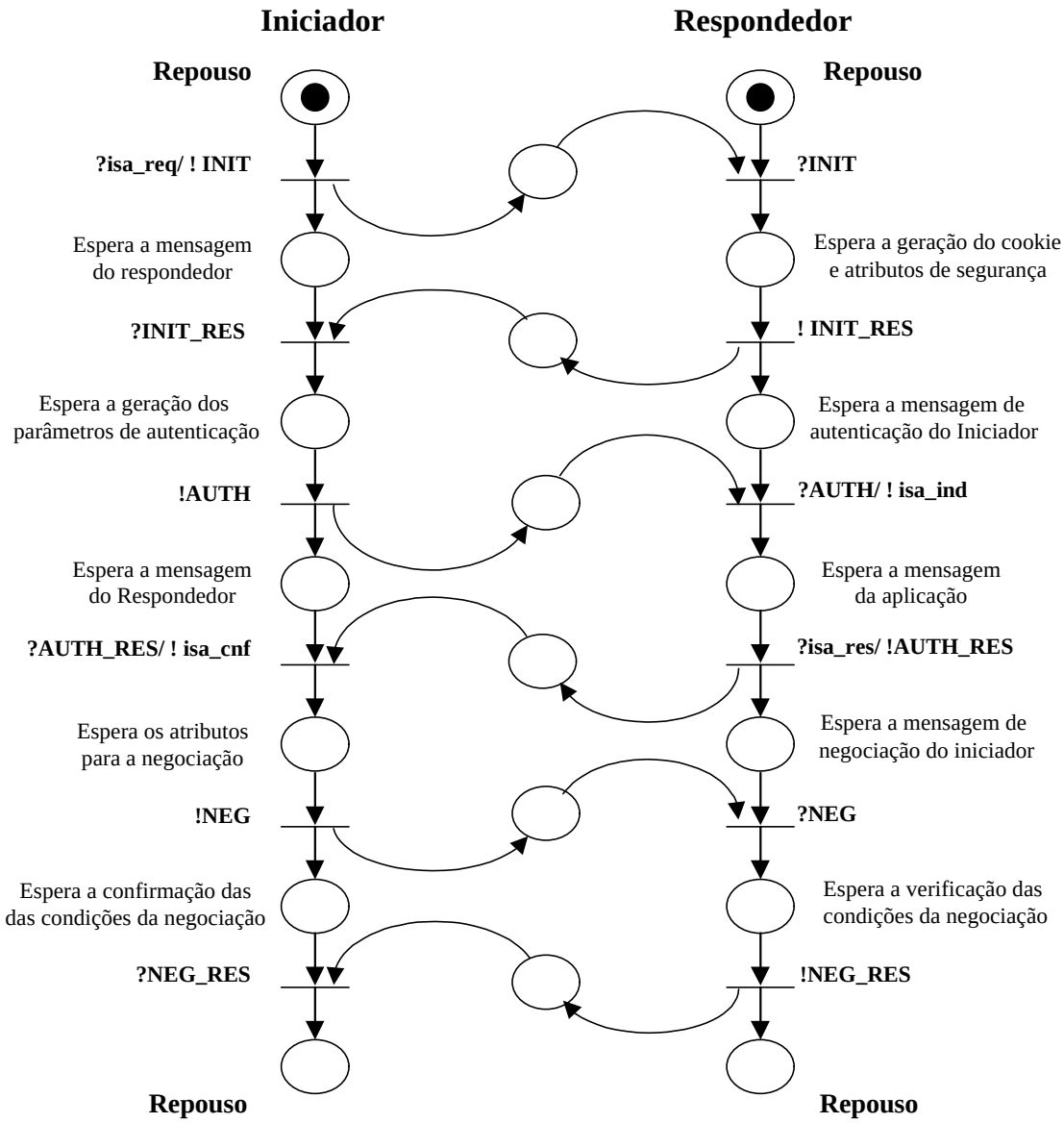

Figura 27. Rede de Petri do Protocolo ISA em Modo Principal.

Podemos observar também na figura 28 as interações com dois usuários $USER_A$ e $USER_B$ entre os protocolos ISA_A e ISA_B, trocando mensagens através do protocolo TP. Os protocolos interagem entre si, considerando que o protocolo ISA lhes dá total suporte para o acordo da associação segura.

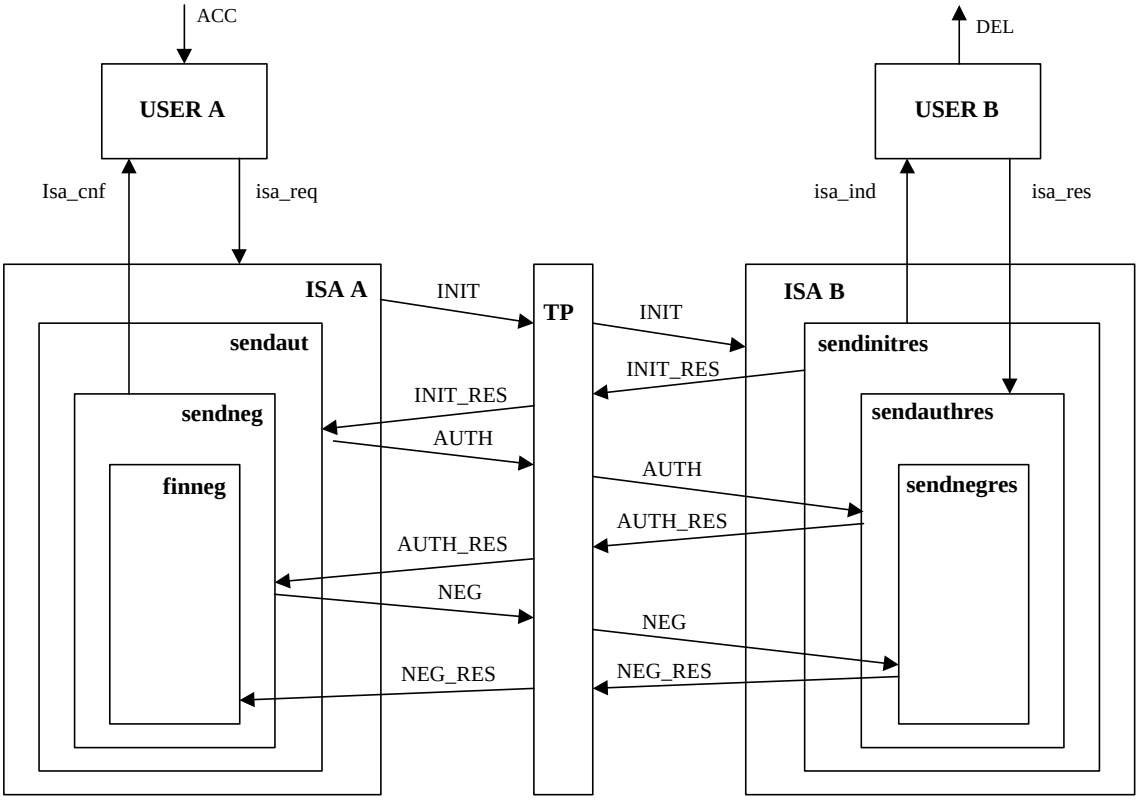

Figura 28. Processos do Protocolo ISA em Modo Principal.

Nesta fase do projeto há troca de mensagens com parâmetros entre dois nós, sendo um iniciador e o outro o respondedor, visando facilitar a visualização dos mecanismos do protocolo.

Após a implementação deste modo de operação principal verificamos que através de sua utilização estaríamos garantindo um estabelecimento mais eficaz, atendendo assim os requisitos do projeto. A partir desta determinação iniciou-se toda a metodologia incremental, de onde foram feitos modelos mais elaborados para análise e correção. A construção de estruturas simples com incrementos contínuos em complexidade é a maneira mais prática de uma descrição estruturada para a especificação formal de protocolos extensos. A descrição desta maneira tem por objetivo testar a maioria das propriedades observacionais.

O protocolo foi modelado em LOTOS seguindo a processo de comunicação a seguir:

- Ao receber uma mensagem da camada superior, o processo $USER_A$ solicita os serviços da camada ISA;

- Os processos ISA_A e ISA_B representam o protocolo de associações seguras dentro de cada SEG, funcionando de forma seqüencial e sincronizada;
- Os processos internos ao ISA_A e ISA_B são encadeados e interagem seqüencialmente até a última mensagem;
- Quando ISA_A recebe DATA, ele envia ao ISA_B os parâmetros de negociação suportados e sua identificação, que confirma o recebimento, enviando os seus parâmetros da negociação e sua identificação para o outro ISA_A;
- O ISA_A confirma a negociação verificando os parâmetros acordados e envia os componentes da troca Diffie Hellman, de onde serão geradas as chaves para o ISA_B;
- O ISA_B recebe esta informações e envia o DATA para o $USER_B$
- Ao receber a resposta de confirmação de negociação do $USER_B$, o ISA_B envia a confirmação dos componentes da troca Diffie Hellman;
- O ISA_A recebe os componentes da troca Diffie Hellman, gera a chave criptográfica e envia a confirmação da negociação e identidade criptografados para o ISA_B;
- O ISA_B envia a confirmação da negociação e identidade criptografados para o ISA_A;
- Após esta confirmação a comunicação entre os SEGs se encerra, ISA_B volta a seu estado inicial.
- O ISA_A, após confirmar o êxito da comunicação para $USER_A$, também volta a seu estado inicial e o canal está livre para uma nova comunicação.

O comportamento deste estabelecimento pode ser verificado através do gráfico de estados e transições da figura 29. A ferramentas do CADP, também foi usada na análise da especificação formal em LOTOS para gerar o gráfico comportamental representado pelo número de estados e transições minimizados para o protocolo. Pelo mesmo motivo apresentado anteriormente, foi necessário minimizar o gráfico para facilitar a visualização.

Os resultados mostraram que para esta modalidade de funcionamento se obteve o número de 29 estados e 37 transições sendo reduzido para 28 estados e 36 transições,

descartando um estado e uma transição redundante no comportamento, demostrando a eficácia da especificação. Os gráficos demonstram a coerência funcional da descrição e a confirmação das propriedades observacionais.

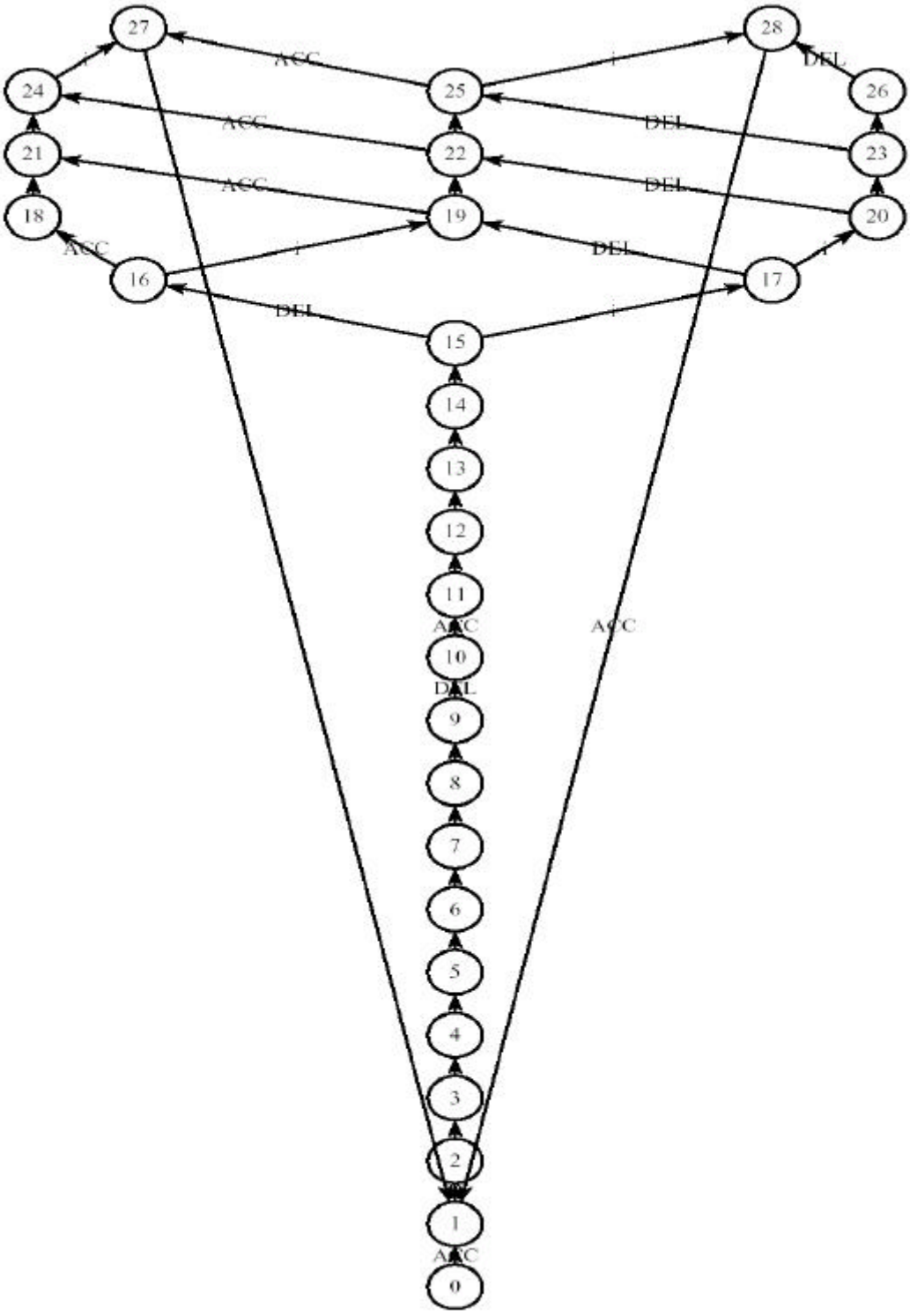

Figura 29. Gráfico de Estados e Transições do Protocolo ISA em Modo Principal.

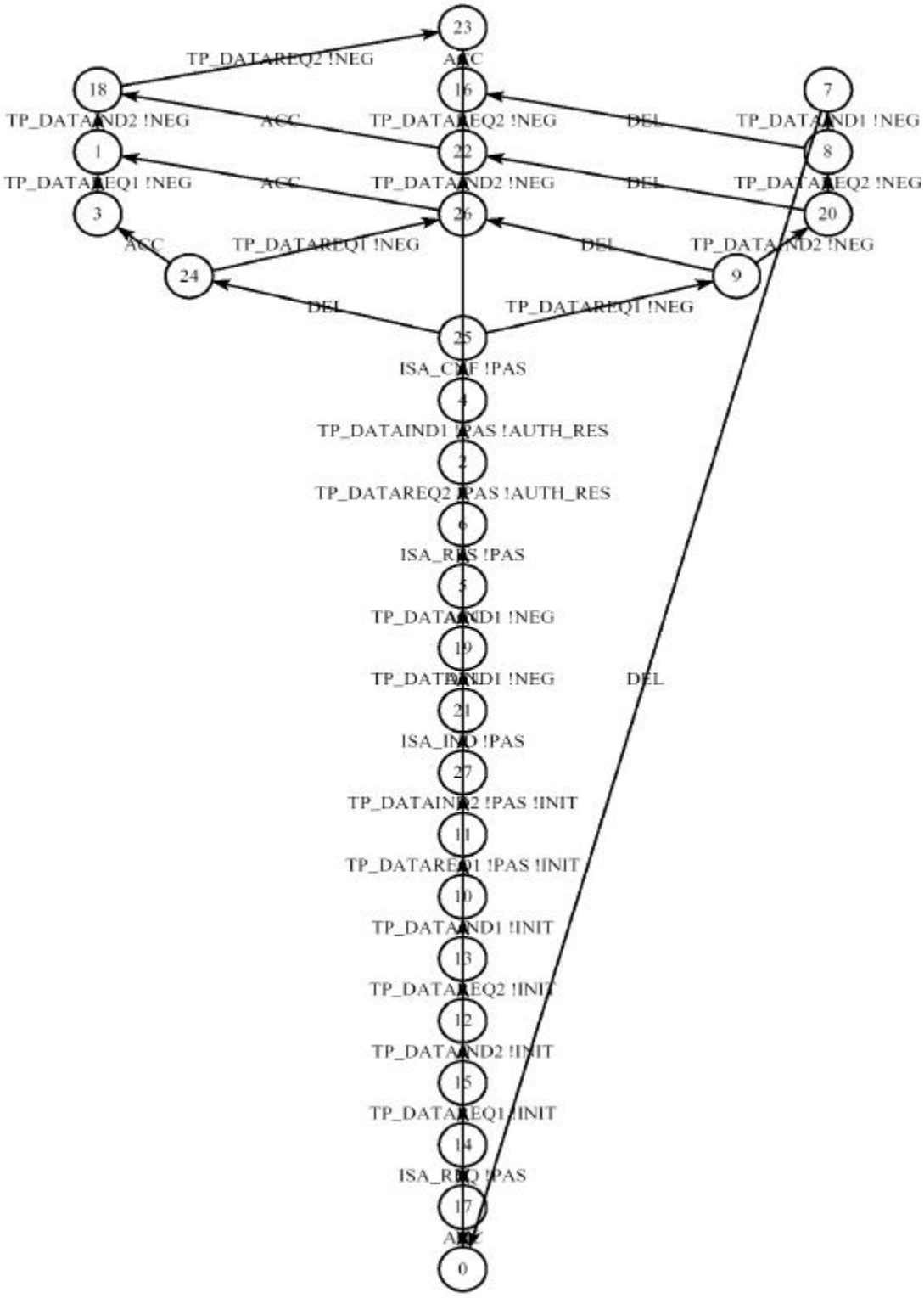

Figura 30. Gráfico Minimizado de Estados e Transições do Protocolo ISA em Modo Principal.

Nas simulações que resultaram nesta última especificação foram feitos testes de lógica inter processos, entre as mensagens. Cada ramo da arvore comportamental que é

apresentada pelo gráfico apresenta uma alternativa de rota para as mensagens entre os SEGs, todas com origem no SEG emissor. A ocorrência de poucos estados e transições redundantes prova que a especificação final se tornou muito eficaz na prestação do serviço. Estas estruturas de decisão de envio e recebimento de mensagens foram testadas tendo como objetivo verificar a possibilidade de haver falhas como, por exemplo, "*deadlocks*". Algumas simulações não geraram resultados que contribuíssem de forma significativa para o projeto e por isso não foram citadas.

Com o comportamento apresentado pelo protocolo neste ponto do projeto, podemos afirmar que já está bem estabelecido e consolidado. Os mecanismos de troca de mensagens estão bastante robustos para garantir que ele convergirá mesmo para uma quantidade maior de nós interagindo nas redes. Para isto, seria necessária realizar simulações com o aumento das associações entre os nós das redes.

4.2.3. Resumo dos Resultados Obtidos

Os resultados obtidos com a verificação do comportamento do protocolo, em relação ao modelo do serviço, podem ser vistos na tabela 4, com o número de estados e transições do protocolo.

Por se tratar de modos diferentes de execução de um mesmo protocolo, a grande diferença encontrada é a expansão do comportamento dos estados e rótulos que determinam o grau de vulnerabilidade dos modos em relação ao comportamento das mensagens. Podemos perceber que o modo principal apresenta um número maior de estados e transições, mas com menos redundância no comportamento, realizando o serviço proposto pelo protocolo de forma mais eficaz em relação ao modo agressivo.

Na tabela 4 é apresentado o resumo das simulações para o protocolo de associações seguras em modo agressivo e principal. A evolução das simulações retrata o aumento das funcionalidades do protocolo em ambos os modos de operação. A tabela contém informações de cada simulação.

Os campos da tabela 4 estão descritos a seguir:

- Número da simulação;
- Modo de operação;

- Característica da simulação, que indica a minimização do grafo LTS;
- Número de SEGs interagindo, quer sejam emissores, ou receptores ou ambos;
- Número de estados gerados pela ferramenta responsável pela obtenção do grafo LTS;
- Número de transições geradas pela ferramenta responsável pela obtenção do grafo LTS;

Tabela 4. Números de Estados e Transições do Protocolo ISA nos Modos de Estabelecimento de Associações Seguras Principal e Agressivo.

Nº	Modo	Característica	SEGs	Estados	Transições
1	Agressivo	Normal	2	20	25
2	Agressivo	Minimizado	2	18	22
3	Principal	Normal	2	29	37
4	Principal	Minimizado	2	28	36

A observação das equivalências, executada com a ferramenta Aldebaran, foi realizada com relação ao comportamento do protocolo. A equivalência forte, pelo motivo exposto na seção 3.7, não foi verificada integralmente. De fato, existem diferenças entre a evolução do diagrama de estados do protocolo e o do serviço modelado. Diversos estados internos surgem no modelo mais completo do protocolo, que tornam a evolução diferente.

Todas as especificações foram testadas com relação a existência de *deadlocks*, se o protocolo era vivo, reinicializável e se respeitava as propriedades observacionais. Como as simulações convergiram para este tipo de abordagem, cabe aqui salientar a necessidade da passagem para a outra fase do projeto que seria a verificação do comportamento do protocolo com o aumento dos dispositivos que trocam informações entre si para o estabelecimento das associações.

4.2.4. Conclusões sobre as Simulações

Como conclusão das simulações podemos destacar a escolha do modo de operação principal para o estabelecimento das associações seguras, através dos resultados obtidos com a verificação do funcionamento do protocolo ISA.

A verificação das estruturas de troca de mensagens nos dois modos apresenta na parte comportamental as mesmas características funcionais básicas, já que nos dois modos o serviço era prestado de forma correta. Mas, se tratando de um protocolo que trabalha com acertos de parâmetros que são primordiais para a troca segura de informações, optou-se pelo processo mais robusto encontrado no modo principal. Apesar de executar mais mensagens, sua árvore de decisões era mais eficaz, pois o número de estados e transições redundantes era menor. Desta forma, para uma especificação completa devemos agora verificar o comportamento do protocolo com a iniciação de várias associações simultâneas, o que poderemos ver na seção a seguir.

4.3. Simulações com o Aumento do Número de Associações Seguras

Para completar a verificação da especificação do protocolo de estabelecimento das associações seguras (ISA) foi necessária a verificação do comportamento em relação à variação do número de dispositivos estabelecendo associações. Como na seção anterior todas as simulações foram feitas com apenas dois SEGs, esta seção descreve em LOTOS vários SEGs interagindo entre si de forma simultânea. Esta verificação apresenta alternativas quanto ao número de SEGs da rede, sendo executada em uma seqüência lógica e cronológica dentro da filosofia de evolução incremental de complexidade do protocolo.

A ferramenta CADP forneceu conclusões quanto às mensagens e campos necessários para uma comunicação eficiente. Os SEGs têm todas as funcionalidades tanto para a transmissão quanto para a recepção das mensagens. Esta característica se dá devido à linguagem LOTOS, que possibilita a simulação da comunicação entre vários dispositivos ao mesmo tempo, como num caso real.

A linguagem LOTOS funciona como uma máquina seqüencial, permitindo a troca de mensagens entre os dispositivos em um meio de comunicação, desde que estes estejam livres para estabelecer esta conexão. Conseguindo assim uma simulação do protocolo num caso real, onde dispositivos concorrem para obtenção do meio para a comunicação.

Após a correção da escrita do protocolo, realizada com as simulações do estabelecimento de associações seguras entre dois SEGs, foi possível verificar que o aumento dos SEGs gera um aumento suave de estados e transições no primeiro momento, mas se formos acrescentando mais SEGs, os números de estados e transições tendem ao infinito. Isto acontece devido às estruturas de funcionamento do LOTOS, que necessitam de sincronizadores durante a troca de mensagens.

Para conseguir realizar um processo gradual de aumento dos dispositivos interagindo entre as redes foi necessário aumentar um a um o número de SEGs na especificação. Começaremos as simulações com três SEGs até chegarmos ao máximo alcançado neste estudo de seis SEGs estabelecendo associações simultâneas.

4.3.1. Simulações com a Interação de Três SEGs

Para simulações com três SEGs foram modelados três usuários $USER_A$, $USER_B$ e $USER_C$ em três SEGs com seus protocolos ISA_A, ISA_B e ISA_C. Estas três entidades são compostas por processos iniciados por mensagens de INIT durante o estabelecimento de uma associação.

A estrutura iniciada nos usuários $USER_A$, $USER_B$, e $USER_C$ são compostos por processos encadeados, inicializáveis pelos seus sucessores, em resposta as mensagens recebidas sincronamente ao processo ISA_A, ISA_B e ISA_C, contendo o processo AUTH de autenticação que precede o processo de negociação NEG.

Nas simulações, o protocolo possui três nós, todos receptores e emissores, neste caso, há a possibilidade de troca de mensagens entre os nós. Nesta forma o protocolo já cobre todas as funcionalidades de troca de parâmetros. A figura 31 mostra a estruturação de dados e do fluxo de mensagens para o incremento das novas possibilidades descritas anteriormente.

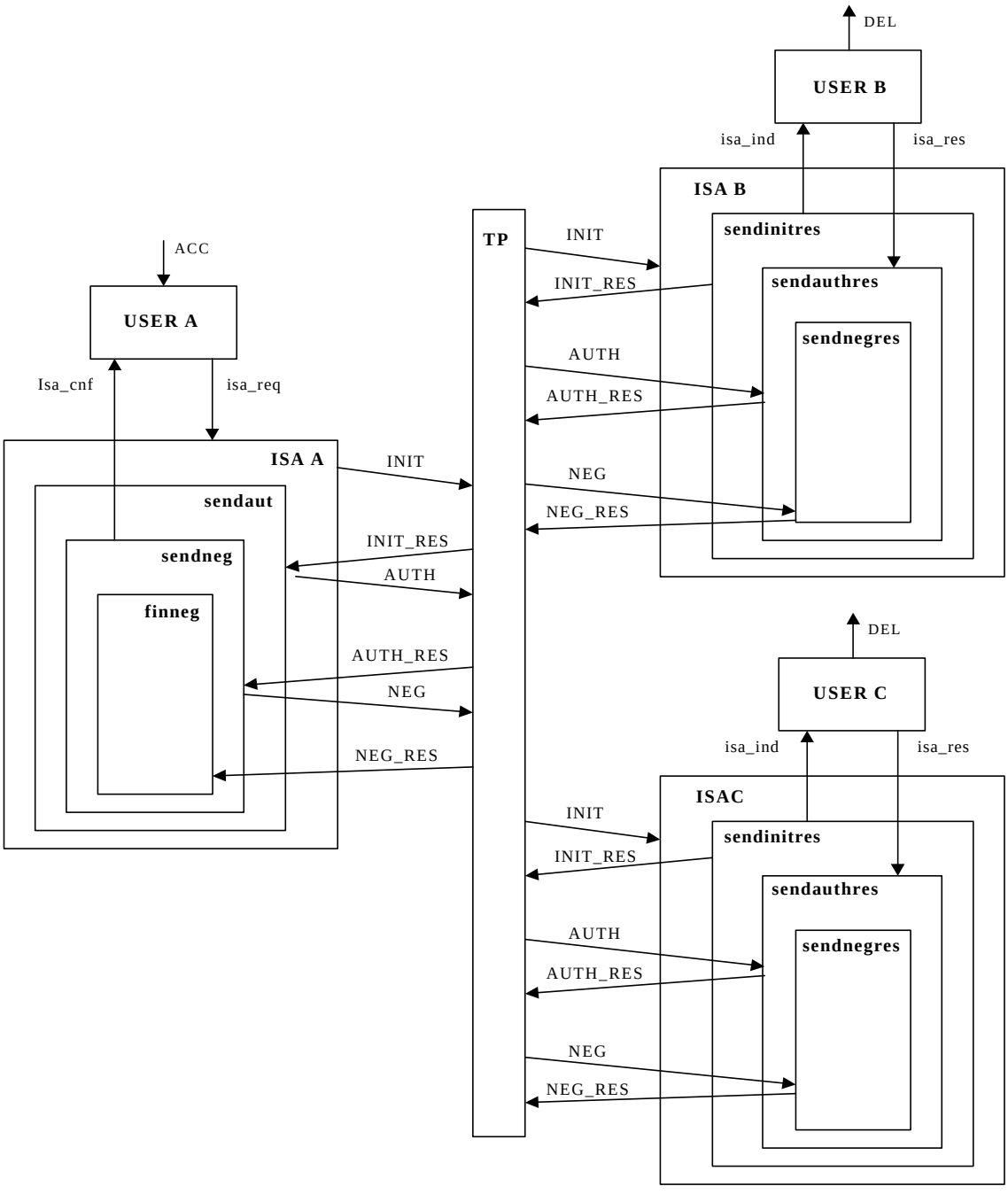

Figura 31 - Processos do Protocolo ISA com Três SEGs.

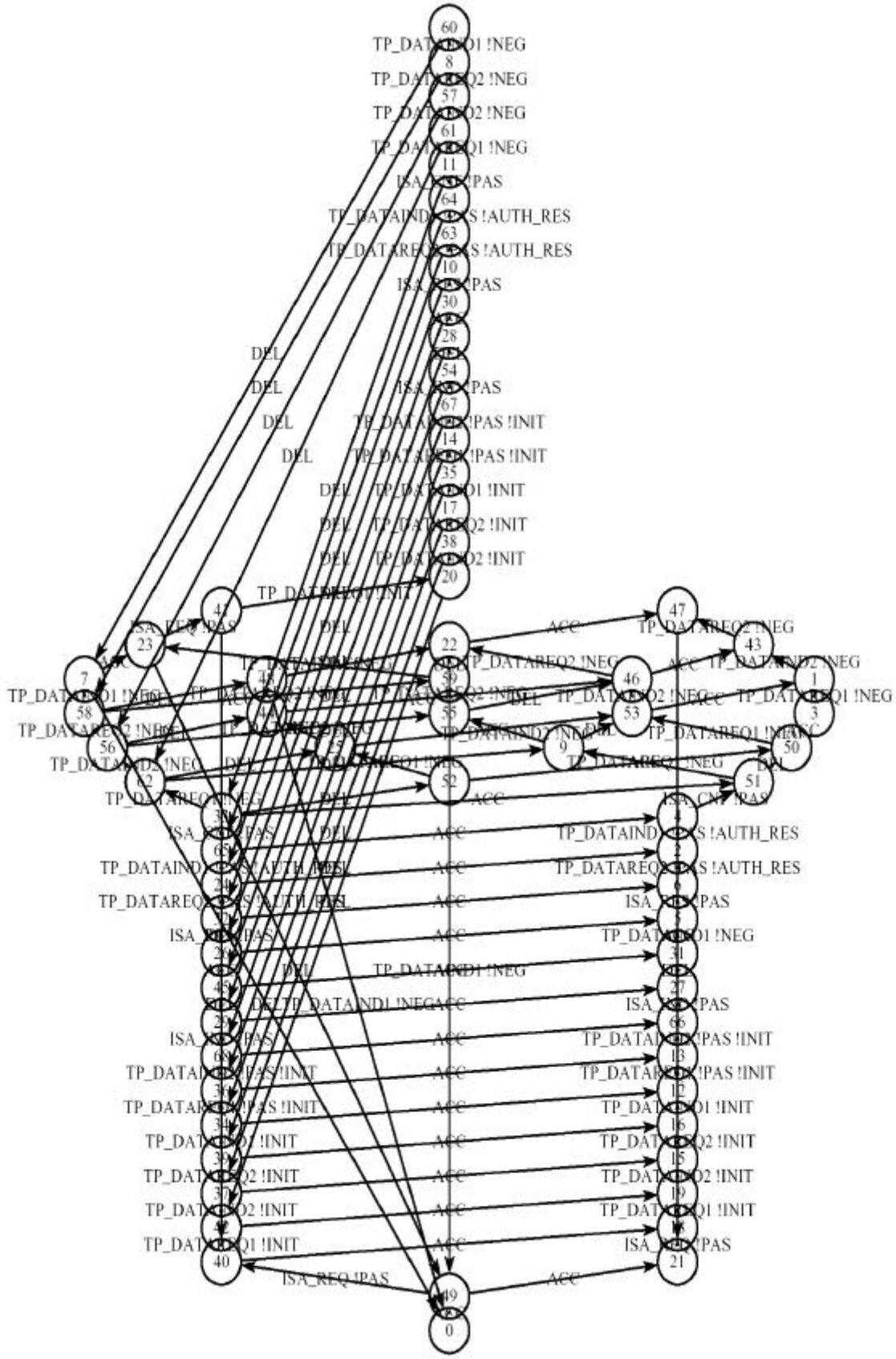

Figura 32 - Gráfico Minimizado de Estados e Transições do Protocolo ISA com Três SEGs.

Neste ponto do projeto o protocolo ISA já encontra-se com todas as funcionalidades necessárias para o correto estabelecimento de associações seguras entre dispositivos de borda da rede 3G. Verificamos o comportamento de várias possibilidades de troca de mensagens entre 3 SEGs, incluindo através da linguagem LOTOS a possibilidade de envio e recepção de mensagens simultâneas de iniciação de associação. O gráfico LTS normal é composto por 283 estados e 538 transições, o que impossibilitou sua apresentação. Já o gráfico minimizado com 69 estados e 122 transições, apresentado na figura 32, mostra todas as implementações descritas anteriormente, confirmando a convergência da simulação, além de ser viva, inicializável e não possuir *deadlocks*.

4.3.2. Simulações com o Aumento Gradual dos SEGs

Para que fique completa a validação do protocolo ISA será realizada a verificação do comportamento do protocolo com o aumento número de SEGs. Este processo tem como função apresentar a evolução dos estados e transições obtidos nas simulações com a inclusão de novas entidades, estabelecendo associações seguras entre si, possibilitando observar a convergência do protocolo, mesmo executando grande números de associações.

Nas simulações ficou constatado que a partir das interações com quatro SEGs não é mais possível a visualização total dos grafos, mesmo minimizados, pois nesta situação o grafo LTS torna-se um borrão na tela devido ao número muito grande de estados e transições. Isto se deve às estruturas de programação do LOTOS, que necessitam de sincronização do meio para impossibilitarem a geração de infinitas alternativas.

Verificamos que a ferramenta de simulação não conseguiu gerar o grafo LTS para especificações com mais de sete SEGs. Porém pelo menos o aplicativo CADP informou que o protocolo estava livre de *deadlocks*. Isto acontece pois a capacidade de funcionamento do software está na ordem de dezenas de milhões de estados e transições e para minimizar este grafo a ferramenta precisava gerar outros estados e transições para testes de minimização. Este mecanismo de funcionamento fazem com que estas quantidades ultrapassassem a capacidade de análise da ferramenta. Esta característica de

resposta, com excessiva quantidade de estados e transições, limita a análise dos resultados até a interação de seis SEGs.

4.3.3. Resumo dos Resultados Obtidos

Os resultados obtidos com a verificação do comportamento do protocolo modelado, em relação à variação do número de entidades envolvidas (SEGs), podem ser vistos na tabela 5, junto com o número de estados e transições.

Por se tratar de execução de um mesmo protocolo com a variação do número de entidades envolvidas no processo, a expansão do comportamento dos estados e transições demonstrou que mesmo com a variação de associações, o protocolo continuava convergindo e mantendo-se compatível com as propriedades observacionais.

A seguir é mostrada a tabela 5 com o resumo das simulações para a verificação do comportamento do protocolo com associações simultâneas. A evolução das simulações retrata o aumento das funcionalidades do protocolo com a inclusão gradual dos SEGs. Esta tabela contém as informações de cada simulação.

Os campos da tabela 5 estão descritos a seguir:

- Número da simulação;
- Número de SEGs interagindo,
- Número de SEGs emissores;
- Número de SEGs receptores;
- Número de estados gerados pela ferramenta responsável pela obtenção do grafo LTS;
- Número de transições geradas pela ferramenta responsável pela obtenção do grafo LTS;
- Número de estados minimizados gerados pela ferramenta responsável pela obtenção do grafo LTS, sem seus estados e transições redundantes;
- Número de transições minimizadas geradas pela ferramenta responsável pela obtenção do grafo LTS, sem seus estados e transições redundantes.

Tabela 5. Números de Estados e Transições do Protocolo ISA no Estabelecimento de Associações Seguras com o Aumento do Número de Associações.

SIMULAÇÕES						
Item	**Nº SEGs**	**SEGs (emissores)**	**SEGs (receptores)**	**Estados**	**Transições**	**Características**
1ª	2	1	1	29	37	normal
2ª	2	1	1	28	36	minimizado
3ª	3	1	2	142	270	normal
4ª	3	1	2	69	122	minimizado
5ª	4	1	3	649	1659	normal
6ª	4	1	3	128	262	minimizado
7ª	4	2	2	1405	3294	normal
8ª	4	2	2	393	8867	minimizado
9ª	5	1	4	2782	8968	normal
10ª	5	1	4	205	405	minimizado
11ª	6	1	5	11341	44177	normal
12ª	6	1	5	300	704	minimizado
13ª	6	3	3	90262	302103	normal
14ª	6	3	3	3910	11809	minimizado

As simulações foram feitas com o intuito de verificar o comportamento do protocolo ISA com o aumento de entidades envolvidas nas associações. Este processo foi conduzido conforme apresentado abaixo:

- Na primeira simulação foram descritos apenas dois SEGs cujas estruturas de recepção e transmissão foram implementadas para realizar as associações não simultâneas, sendo um transmissor e o outro receptor;
- Na segunda simulação três SEGs foram descritos sendo um transmissor e

dois receptores, trocando mensagens de associação simultaneamente entre três redes distintas;

- Na terceira simulação foram descritos quatro SEGs sendo dois transmissores e o dois receptores realizando associações entre as redes;
- Na quarta simulação foram descritos cinco SEGs realizando associações simultâneas com quatro redes distintas, sendo um transmissor e quatro receptores;
- Na sexta simulação seis SEGs foram descritos sendo que um transmissor e cinco receptores, podendo realizar associações entre cinco redes simultaneamente;
- Na sétima simulação foram descritos seis SEGs sendo que três transmissores e três receptores, realizando associações entre as redes.

Até este ponto todos as simulações convergiram e foram comprovadas todas as propriedades observacionais, provando que em todas as etapas do projeto, o protocolo era convergente, vivo, reinicializável e isento de *deadlock*s. A partir daí, como já havíamos comentado, a ferramenta não gera mais os grafos LTS.

A equivalência observacional, executada com a ferramenta Aldebaran, foi verificada inteiramente. A equivalência forte, pelo motivo já comentado anteriormente, não foi verificada integralmente.

Com as simulações foi possível verificar que o número de estados e transições cresce bruscamente com a entrada de mais SEGs no processo de associação, o que eleva a complexidade do processo de troca de mensagens no protocolo. Observamos também que, pela modelagem formal de um protocolo, podemos aferir seu comportamento através dos estados ocorridos e verificar eventuais falhas nos procedimentos executados por ele.

4.3.4. Conclusões sobre as Simulações

A conclusão que se chega com todos os testes e simulações feitas é que o protocolo de associações seguras (ISA) realmente funciona e converge. A contribuição deste estudo foi especificar o protocolo ISA em LOTOS e complementá-lo com as

estruturas de decisões e análises do ambiente 3ª geração. Tendo como base este princípio, foram feitas formalizações em LOTOS destas estruturas, possibilitando o teste e aperfeiçoamento das especificações do protocolo através da metodologia de projeto e das fases das simulações mostradas anteriormente.

Alguns dos resultados obtidos nas diversas fases da simulação estão descritos na tabela 5, mostrando a seqüência lógica e cronológica dentro da filosofia de evolução incremental em complexidade do protocolo descrito e mostrado anteriormente.

Pela análise dos experimentos observa-se que a seqüência lógica e cronológica dentro da filosofia de evolução incremental em complexidade foi ampliada para acrescentar as características de emissão e transmissão em cada SEG. A inclusão destas característica, como passagem de parâmetros, é fundamental, principalmente quando se amplia o número de SEGs no sistema.

Por estes motivos à evolução dos experimentos seguiu um caminho natural de complexidade, passando primeiro pelos processos simples, seguido do aumento de estações e da complexidade destas; numa fase seguinte foi realizada a inclusão de todas as características em conjunto com o incremento gradual de SEGs e associações, possibilitando ao protocolo ter um máximo de funcionalidades.

A afirmação de não haver *deadlocks* é consistente, porque a metodologia de simulação utilizada testou todas as características, possibilidades e peculiaridades do protocolo, através de simulações anteriores que sempre convergiram.

4.4. Comentários

O capítulo apresentou as simulações e os resultados alcançados neste estudo, ressaltando que os resultados obtidos foram conseguidos passo a passo e em várias simulações dentro de cada processo de validação. As funcionalidades descritas no protocolo foram abordadas em várias etapas, onde cada uma delas simulava parte do mesmo problema.

Normalmente as especificações foram refeitas diversas vezes, culminando em uma sólida abordagem do protocolo, produzindo alterações descritivas que foram frutos destas simulações. A partir destas especificações foram estudadas novas formas de

validação, que representavam as associações com mais realismo.

O estudo do comportamento com o aumento dos participantes só fez sentido a partir de uma especificação madura do protocolo. Esta especificação foi fruto do ganho de experiência no manuseio da linguagem, que foi sendo agregado a cada simulação.

Capítulo 5

Conclusão

Este trabalho analisou o protocolo de estabelecimento de associações seguras do sistema móvel de 3ª geração, utilizando um processo de verificação formal através da linguagem LOTOS e da ferramenta CADP. Este processo possibilitou a correta validação da especificação do protocolo, com passagem de dados, sem a necessidade da implementação do código.

O protocolo foi analisado através de sua especificação formal, a fim de verificar e validar as exigências de funcionalidade para o estabelecimento de uma conexão segura entre as redes 3G, garantindo assim os aspectos fundamentais para a segurança do sistema. As redes 3G tiveram descrições genéricas para uma abordagem mais geral quanto possível. Sua estrutura foi mostrada fisicamente e sua arquitetura em camadas mostrou as ligações dos protocolos por meio de PDUs e SDUs.

O processo de validação do protocolo seguiu uma metodologia que se propunha o aumento gradual de funcionalidades, permitindo a utilização de uma estrutura de segurança pertencente a arquitetura de sistemas móveis de 3ª geração.

Toda validação foi realizada com o auxílio de métodos formais porque estes são baseadas em princípios matemáticos, permitindo uma boa modelagem e posterior verificação, análise e validação de forma genérica, precisa e sem ambigüidades de todos os protocolos e das interligações existentes entre eles. A proposta de analisar as associações seguras tem como objetivo estabelecer futuras implementações através de uma arquitetura segura, em multicamadas, desde o meio físico até o nível de aplicação.

Foi utilizado um processo de verificação das propriedades comportamentais do protocolo, tendo como base a verificação da quantidade dos seus estados, confirmação das propriedades observacionais e equivalências observacionais. A equivalência forte não foi observada devido à simplificação ocorrida durante o processo de minimização, que resultou num diagrama com número de estados muito menor que o original, além de ter sua numeração alterada.

As equivalências foram definidas para garantir que o protocolo modelado apresenta, em termos observacionais, o mesmo comportamento que se espera do serviço de segurança modelado.

Outra análise foi a escolha do melhor modo de operação para o estabelecimento das associações seguras, que levou em conta a quantidade de estados redundantes encontrados e o comportamento do protocolo nestes modos. Neste caso, foi constatado que o protocolo trabalhando em modo principal apresenta o funcionamento mais eficaz na fase de iniciação e possui maior complexidade das mensagens, o que pode ser interpretado como uma segurança a mais contra interceptações e retransmissões.

A análise da quantidade de estados e transições a cada inclusão de SEGs nas simulações nos permite afirmar que, mesmo com o número elevado de estados encontrados, o protocolo funcionou corretamente conforme especificado, sem a possibilidade de *deadlocks,* sendo vivo e inicializável.

A apresentação das especificações formais em LOTOS e a verificação de propriedades essenciais de funcionamento, mostrou-se viável devido aos resultados obtidos com a validação do protocolo frente ao modelo requerido para o serviço. Pode-se afirmar que a análise da evolução do comportamento com o aumento do número de entidades envolvidas no processo de associação permite a verificação do comportamento mais próximo do real. Assim, algumas conclusões e contribuições podem ser destacadas:

- A inclusão do processo completo de validação do comportamento, desde o comportamento básico entre dois dispositivos e indo até vários dispositivos interagindo simultaneamente;
- A metodologia empregada que auxilia numa análise qualitativa e quantitativa dos protocolos, facilitando a tomada de decisão sobre a que parte do protocolo se deve dar atenção em relação ao teste de determinadas características que se queira colocar em evidência no protocolo;
- Destaca-se também a implementação de estruturas de testes e decisões para implementação dos processos de troca de mensagem, que foram construídas tendo como referência descrições sucintas contidas no IETF.

Todos os testes foram feitos levando em consideração a estrutura do protocolo e a arquitetura em camadas das ligações dos protocolos por meio de PDUs e SDUs. Como passos futuros, seriam necessárias as especificações dos novos protocolos de segurança da arquitetura IPSec, como o protocolo IKEv2, sendo descrito e submetido à mesma abordagem empregada no protocolo estudado.

Referencias Bibliográficas

[1] International Telecommunication Union (ITU) - http://www.itu.int

[2] European Telecommunications Standards Institute (ETSI) – http://www.etsi.org

[3] Third Generation Partnership Project Forum (3GPP Forum) - http://www.3gpp.org

[4] MAUGHAN, D., SCHERTLER, M., SCHNEIDER, M. e TURNER, J., “Internet Security Association and Key Management Protocol (ISAKMP)”, RFC 2408, 1998.

[5] KENT, S. e ATKINSON, R, “Security Architecture for the Internet Protocol”, IETF RFC 2401, 1998.

[6] BOLOGNESI, T., BRINKSMA, E., "Introduction to the ISO Specification Language LOTOS", Computer Networks and ISDN Systems, v. 14, n. 1, pp. 25-29, 1987.

[7] FERNANDEZ, J. C., GARAVEL, H., KERBRAT, A., MATEESCU, R., MOUNIER, L. e SIGHIREANU, M., “CAESAR/ALDEBARAN Development Package: a protocol validation and verification toolbox”, Proceedings of the Eighth Conference on Computer-Aided Verification, (CAV), pp. 437-440, New Brunswick, Aug. 1996.

[8] 3GPP TS 23.101: , “3rd Generation Partnership Project; Technical Specification Group (TSG); "General UMTS Architecture".

[9] 3G TS 33.102, “3rd Generation Partnership Project; Technical Specification Group; 3G Security Architecture”, 2003.

[10] QIAN, T. e CAMPBELL, R., Dynamic Agent-based Security Architecture for Mobile Computers. In Proceedings of the International Conference on Parallel and Distributed Computing and Networks (PDCN'98), Australia, December 1998.

[11] CLAESSENS, J., PRENEEL B. e VANDEWALLE, J., Combining World Wide Web and wireless security. In Proceedings of IFIP TC11 WG11.4 First Annual Working Conference on Network Security, pages 153--171, Boston, 2001.

[12] IPSec Group "The Internet Engineering Task Force in Security group - IP Security Protocol", http://www.ietf.org/html.charters/ipsec-charter.html.

[13] RIBEIRO, F.J.L., LOPES, J.C.R., PEDROZA, A.C.P., "Análise dos Processos de Segurança em Sistemas Móveis de 3ª Geração". In: Anais da I Escola Regional de Redes de Computadores, pp. 59-64, Porto Alegre, Set., 2003.

[14] 3GPP TS 33.120, "3rd Generation Partnership Project; Technical Speci_cation Group (TSG); 3G Security; Security Principles and Objectives", 1999.

[15] 3GPP TS 29.060, "3rd Generation Partnership Project; Technical Specification Group Core Network; General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp Interface", 2003.

[16] LIN, Y. B., RAO, H. C. H. e CHLAMTAC I., "General packet radio service (GPRS): architecture, interfaces, and deployment," Journal on Wireless Communications and Mobile Computing, vol. 1, pp. 77-92, 2001.

[17] RAUTPALO, J.,"GPRS Security - Security Remote Connections over GPRS", Helsinki University of Technology Department of Computer Science, http://www.hut.fi/~jrautpal/gprs/gprs_sec.html, 2000.

[18] 3GPP TS 33.102, "3rd Generation Partnership Project; Technical Specification Group (TSG); 3G Security; Security Architecure", 2003.

[19] PECHEUR, C., LEDUC, G., BONAVENTURE, O., LÉONARD, L. e KOERNER, E., "Model-based verification of a security protocol for conditional access to services" , 1990.

[20] GERMEAU, F., LEDUC, G., "Model-based Design and Verification of Security Protocols using LOTOS". In: Proceedings of the DIMACS Workshop on

Design and Formal Verification of Security Protocols, New Jersey, Sep., 1997.

[21] LEDUC, G., “Verification of two versions of the Challenge Handshake Authentication Protocol”, 2001.

[22] 3GPP TS 33.210, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Network domain security; IP network layer security”, 2003.

[23] 3GPP TS 33.203: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Access security for IP-based services",2003.

[24] HARKINS, D. e CARREL, D., “The Internet Key Exchange (IKE)”, IETF RFC 2409, 1998.

[25] KENT, S. e ATKINSON, R., “IP Authentication Header”, IETF RFC 2402, 1998.

[26] KENT, S. e ATKINSON, R., “IP Encapsulating Security Payload (ESP)”, IETF RFC 2406, 1998.

[27] KAUFMAN, C., PERLMAN, R. e SPENCINER, M., Network Security Private Communication in a Public World. 2nd ed. New Jersey, Prentice Hall, 2002.

[28] ORMAN, H., “The OAKLEY Key Determination Protocol”, IETF RFC 2412, 1998.

[29] DIFFIE, W. e HELLMAN, M. E., 1976, “New Directions in Cryptography,” Transactions on Information Theory, v 22.

[30] GARAVEL, H. e SIFAKIS, J., “Compilation and Verification of LOTOS Specifications”, VERILOG Rhône-Alpes, 1990.

[31] BAGATELLI, R., MOURA, D. F. C. e PEDROZA, A. C. P., “Especificação Formal de uma Arquitetura de Suporte à Descoberta de Serviços em Redes Móveis Ad Hoc”, in: Anais do V Workshop de Métodos Formais (WMF'2002), Gramado, RS, Brasil, 2002.

[32] ISO/IEC, "IS 8807: Information Processing Systems -- Open Systems Interconnection -- LOTOS -- A Formal Description Technique based on the Temporal Ordering of Observational Behaviour", Geneva, Switzerland, 1998.

[33] MEER, J., ROTH, R. e VONG, S., “Introduction to Algebraic Specifications Based

on the Language ACT ONE". Computer Networks and ISDN Systems, 23(5): 363-392, 1992.

[34] MILNER, R., Communication and Concurrency. Englewood Cliffs, Prentice-Hall, 1989.

[35] HOARE, C.A.R., Communicating Sequential Processes. Englewood Cliffs, Prentice-Hall, 1985.

[36] GERMEAU, F., LEDUC, G., "Verification of Security Protocols Using LOTOS-method and Appplication", Computer Communication 23, páginas 1089-1103, Elsevier Science B. V., "http://www.elsevier.com/locate/comcom, 2000.

[37] GARAVEL, H.,"An Overview of the Eucalyptus Toolbox", INRIA Rhône- Alpes/ VERIMAG, Zirst, 655, avenue de l'Europe, F-38330, Montbonnot Saint Martin, France, 1997.

[38] VASY - INRIA Rhône-Alpes, Case Studies Achieved using the CADP Toolset, Siège de l'INRIA (moyens d'accès), Domaine de Voluceau, Rocquencourt - B.P. 105, 78153 Le Chesnay Cedex - France, http://www.inrialpes.fr/vasy/cadp/case-studies/, 2001.

[39] CADP (Caesar/Aldebaran Development Package) "A Software Engineering Toolbox for Protocols and Distributed Systems", Version 1.163, http://www.inrialpes.fr/vasy/cadp/, 2003.

[40] EUCALYPTUS, "University of Liège (Sart-Tilman Campus)", Institut d'Electricité Montefiore (Parking P 32, Building B 28), B-4000, Liège 1, Belgium http://www.run.montefiore.ulg.ac.be/Projects/Presentation/index.php?project=Eucalyptus, 2003.

[41] APERO, "Université de Liège, Institut Montefiore (B28)", B-4000 Liège, BELGIUM, http://www.run.montefiore.ulg.ac.be/Projects/Presentation/Apero, 2003.

[42] ELUDO, The UofO LOTOS Research Group, University of Ottawa, Canada, http://lotos.site.uottawa.ca/eludo/, 2003.

[43] EHRIG, H., MAHR, B., "Fundamentals of Algebraic Specifications", livro Monographs on Theoretical Computer Science 1 (2), volume 6 (21), editora

Springer-Verlag, 1990.

[44] MARRERO, W., CLARKE, E. e JHA, S., “A Model Checker for Authentication Protocols”. Proc. of the DIMACS Workshop on Design and Formal Verification of Security Protocols, Rutgers University, 1997.

[45] RIBEIRO, F.J.L., LOPES, J.C.R., PEDROZA, A.C.P., “Análise do Estabelecimento de Associações Seguras em Sistemas Móveis de 3ª Geração”. In: Proceedings of I2TS’2003 – 2nd International Information and Telecommunication Technologies Symposium, Florianópolis, Nov., 2003.

[46] LAU, F., RUBIN, S. H., SMITH, M.H. e TRAJKOIC, L., Distributed denial of service attacks. In Proc. 2000 IEEE Int. Conf. on Systems, Man, and Cybernetics, Nashville, TN, volume 3, pages 2275{2280. IEEE Press, 2000.

[47] GARAVEL, H., HERMANNS, H., 2002, “On Combining Functional Verification and Performance Evaluation using CADP”, Théme, Rapport de Recherche, número 4492.

Apêndice A - Especificação do Protocolo ISA em Modo Agressivo

specification ISA_AGR [ACC, tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2, isa_req, isa_cnf, isa_ind, isa_res, DEL] : noexit

library

LIBSEG

endlib

behaviour

hide tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2, isa_req, isa_cnf, isa_ind, isa_res in

(

(

(((

USER_A [ACC, isa_req, isa_cnf, DEL] (PAS of data_type)

|||

USER_B [isa_ind, DEL, ACC, isa_res] ({} of data_type)

)

```
   |[isa_req, isa_cnf, isa_ind, isa_res]|
   (
    ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PAS of data_type, INIT_AUTH
of pdu_type)
    |||
    ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PAS of data_type)
    ))
    )
    )
    |[tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]|
     TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
    )

where
(* DESCRICAO DO PROTOCOLO ISA EM MODO AGRESSIVO *)

process USER_A [ACC, isa_req, isa_cnf, DEL] (DATA: data_type) : noexit :=
     ACC; (* Aplicacao recebe um pedido de iniciacao da AS *)
     isa_req !DATA; (* ISA recebe um pedido de iniciacao da AS da aplicacao *)
     isa_cnf ?DATA: data_type; (* ISA envia a confirmacao da AS para a aplicacao*)
     DEL; (* Aplicacao recebe uma confirmacao da AS *)
     USER_A [ACC, isa_req, isa_cnf, DEL] (PAS) (* retorna ao estado inicial *)
endproc

process USER_B [isa_ind, DEL, ACC, isa_res] (DATA: data_type)  : noexit :=
```

```
    isa_ind ?DATA: data_type; (* Aplicacao recebe a solicitacao de iniciacao da AS do
ISA *)
    DEL; (* Aplicacao envia a solicitacao de iniciacao *)
    ACC; (* Aplicacao recebe a solicitacao de iniciacao *)
    isa_res !DATA; (* ISA recebe a resposta da AS *)
    USER_B [isa_ind, DEL, ACC, isa_res] (RAS)(* retorna ao estado inicial *)
endproc

process ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (DATA:data_type, PDU:
pdu_type) : exit :=
isa_req ?DATA: data_type;
tp_datareq1 !DATA !PDU;
send_init_auth [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (RAS)
where
  process send_init_auth [isa_req, isa_cnf, tp_datareq1, tp_dataind1]
(DATA:data_type) : exit :=
  tp_dataind1  ?DATA:data_type ?PDU: pdu_type ;
  isa_cnf !DATA;
  send_auth_neg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (AUTH_NEG)
  where
    process send_auth_neg  [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PDU:
pdu_type) : exit :=
    tp_datareq1  !PDU;
ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PAS, INIT_AUTH)
    endproc
  endproc
endproc
```

```
process ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (DATA:data_type) : exit :=
tp_dataind2 ?DATA: data_type ?PDU: pdu_type ;
isa_ind !DATA;
send_init_auth_res [isa_ind, isa_res, tp_datareq2, tp_dataind2](RAS, INIT_AUTH_RES)
where
  process send_init_auth_res [isa_ind, isa_res, tp_datareq2, tp_dataind2] (DATA:data_type, PDU: pdu_type) : exit :=
  isa_res ?DATA:data_type;
    tp_datareq2 !DATA !PDU ;
    send_auth_neg [isa_ind, isa_res, tp_datareq2, tp_dataind2]
  where
    process send_auth_neg  [isa_ind, isa_res, tp_datareq2, tp_dataind2] : exit :=
    tp_dataind2 ?PDU: pdu_type;
    ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PAS)
    endproc
  endproc
endproc

process TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2] : noexit :=
  (tp_datareq1 ?DATA: data_type ?PDU: pdu_type;
  (( tp_dataind2 !DATA !PDU;
   TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
  ) ))
  []
```

```
  (tp_datareq2 ?DATA: data_type ?PDU: pdu_type;
  (( tp_dataind1 !DATA !PDU;
   TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
  ) ))
  []
  (tp_datareq1 ?PDU: pdu_type;
  (( tp_dataind2 !PDU;
   TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
  ) ))

endproc
endspec
```

Apêndice B - Especificação do Protocolo ISA em Modo Principal

```
specification ISA_1_1SEG [ACC, tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2,
isa_req, isa_cnf, isa_ind, isa_res, DEL] : noexit

library
LIBSEG
endlib

behaviour
  hide tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2, isa_req, isa_cnf, isa_ind,
isa_res   in
  (
   (
    (((
    USER_A [ACC, isa_req, isa_cnf, DEL] (PAS of data_type)
    |||
    USER_B [isa_ind, DEL, ACC, isa_res] ({} of data_type)
```

```
   )

|[isa_req, isa_cnf, isa_ind, isa_res]|

   (
    ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (INIT of pdu_type)
    |||
    ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (INIT_RES of pdu_type)
    ))
    )
    )

    |[tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]|
     TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
    )

where

(* DESCRICAO DO PROTOCOLO ISA EM MODO PRINCIPAL  *)

process USER_A [ACC, isa_req, isa_cnf, DEL] (DATA: data_type) : noexit :=

      ACC; (* Aplicacao recebe um pedido de iniciacao da AS *)
      isa_req !DATA; (* ISA recebe um pedido de iniciacao da AS da aplicacao *)
      isa_cnf ?DATA: data_type; (* ISA envia a confirmacao da AS para a aplicacao*)
      DEL; (* Aplicacao recebe uma confirmacao da AS *)
      USER_A [ACC, isa_req, isa_cnf, DEL] (PAS) (* retorna ao estado inicial *)
endproc

process USER_B [isa_ind, DEL, ACC, isa_res] (DATA: data_type)  : noexit :=

      isa_ind ?DATA: data_type; (* Aplicacao recebe a solicitacao de iniciacao da AS do
ISA *)
```

```
    DEL; (* Aplicacao envia a solicitacao de iniciacao *)
    ACC; (* Aplicacao recebe a solicitacao de iniciacao *)
    isa_res !DATA; (* ISA recebe a resposta da AS *)
    USER_B [isa_ind, DEL, ACC, isa_res] (RAS)(* retorna ao estado inicial *)
endproc

process ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PDU: pdu_type) : exit :=
isa_req ?DATA: data_type;
tp_datareq1 !PDU;
sendauth [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PAS, AUTH)
where
  process sendauth [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (DATA: data_type,
PDU: pdu_type) : exit :=
  tp_dataind1  ?PDU: pdu_type;
  tp_datareq1 !DATA !PDU;
  sendneg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (RAS)
  where
    process sendneg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (DATA: data_type) :
exit :=
    tp_dataind1  ?DATA: data_type ?PDU: pdu_type;
    isa_cnf !DATA;
    fineg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (NEG)
    where
      process fineg [isa_req, isa_cnf, tp_datareq1, tp_dataind1](PDU: pdu_type) : exit
:=
      tp_datareq1 !PDU;
      tp_dataind1 ?PDU: pdu_type;
      ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (INIT)
      endproc
    endproc
  endproc
endproc
```

```
process ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PDU: pdu_type) : exit :=
tp_dataind2 ?PDU: pdu_type;
tp_datareq2 !PDU;
sendinitres [isa_ind, isa_res, tp_datareq2, tp_dataind2](PAS)
where
  process sendinitres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (DATA:data_type) :
exit :=
  tp_dataind2 ?DATA: data_type ?PDU: pdu_type;
  isa_ind !DATA;
  sendauthres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (RAS, AUTH_RES)
  where
    process sendauthres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (DATA: data_type,
PDU:pdu_type) : exit :=
    isa_res ?DATA:data_type;
    tp_datareq2 !DATA !PDU;
    sendnegres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (NEG_RES)
    where
      process sendnegres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PDU: pdu_type) :
exit :=
      tp_dataind2 ?PDU: pdu_type;
      tp_datareq2 !PDU;
      ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (INIT_RES)
      endproc
    endproc
  endproc
endproc

process TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2] : noexit :=
  (tp_datareq1 ?Data: data_type ?PDU: pdu_type;
  (( tp_dataind2 !Data !PDU;
   TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
  ) ))
  []
```

```
  (tp_datareq2 ?Data: data_type ?PDU: pdu_type;
  (( tp_dataind1 !Data !PDU;
   TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
  ) ))
  []
  (tp_datareq1 ?PDU: pdu_type;
  (( tp_dataind2 !PDU;
   TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
  ) ))
  []
  (tp_datareq2 ?PDU: pdu_type;
  (( tp_dataind1 !PDU;
   TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
    ))
  )
endproc
endspec
```

Apêndice C – Biblioteca LOTOS

```
(* Obs. 01 : Nao colocar como saida de um operador um sort que seja renomeacao de
outro sort *)
(* Obs. 02 : Se O1 eh um operador definido em um sort S1 entao nao se deve inserir o
comentario de construcao em O1 se sua saida nao for S1 *)
(* Obs. 03 : Se O1 eh um operador definido em um sort S1, deve-se definir os
construtores para operadores que possuirem saida igual a S1, desde que nao sejam da
forma S1->S1 *)

 type PDU is

   sorts PDU_TYPE  (*! implementedby PDUTYPE comparedby CMP_PDUTYPE
                printedby PRINT_PDUTYPE *)
   opns
     INIT (*! implementedby INIT constructor *),
     INIT_RES (*! implementedby INIT_RES constructor *),
     AUTH (*! implementedby AUTH constructor *),
     AUTH_RES (*! implementedby AUTH_RES constructor *),
     NEG (*! implementedby NEG constructor *),
     NEG_RES (*! implementedby NEG_RES constructor *),
     INIT_AUTH (*! implementedby INIT_AUTH constructor *),
     INIT_AUTH_RES (*! implementedby INIT_AUTH_RES constructor *),
     AUTH_NEG(*! implementedby AUTH_NEG constructor *):  -> PDU_TYPE

 endtype
```

```
type DATA is

  sorts DATA_TYPE  (*! implementedby DATATYPE comparedby
CMP_DATATYPE
                printedby PRINT_DATATYPE *)
  opns
    PAS (*! implementedby PPP constructor *),
    RAS (*! implementedby RRR constructor *),
    {} (*! implementedby VVV constructor *):       -> DATA_TYPE
endtype
```

Apêndice D - Especificação do Protocolo ISA com Três SEGs

```
specification ISA_1_2SEG [ACC, tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2,
isa_req, isa_cnf, isa_ind, isa_res, DEL] : noexit

library
LIBSEG
endlib

behaviour
  hide tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2, isa_req, isa_cnf, isa_ind,
isa_res   in

  (
   (
    (

     (
      USER_A [ACC, isa_req, isa_cnf, DEL] (PAS of data_type)
      |||
      USER_B [isa_ind, DEL, ACC, isa_res] ({} of data_type)
      )
      |||
      (
```

```
      USER_A [ACC, isa_req, isa_cnf, DEL] (PAS of data_type)
      |||
      USER_C [isa_ind, DEL, ACC, isa_res] ({} of data_type)
        )
     )
     |[isa_req, isa_cnf, isa_ind, isa_res]|
     (
      ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (INIT of pdu_type)
      |||
      ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (INIT_RES of pdu_type)
     )
    )
    |[tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]|
     TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
   )

where

(* DESCRICAO DA INTERACAO ENTRE USUÁRIOS*)


process USER_A [ACC, isa_req, isa_cnf, DEL] (DATA: data_type) : noexit :=
     ACC; (* Aplicacao recebe um pedido de iniciacao da AS *)
     isa_req !DATA; (* ISA recebe um pedido de iniciacao da AS da aplicacao *)
     isa_cnf ?DATA: data_type; (* ISA envia a confirmacao da AS para a aplicacao*)
     DEL; (* Aplicacao recebe uma confirmacao da AS *)
     USER_A [ACC, isa_req, isa_cnf, DEL] (PAS) (* retorna ao estado inicial *)
endproc


process USER_B [isa_ind, DEL, ACC, isa_res] (DATA: data_type)  : noexit :=
     isa_ind ?DATA: data_type; (* Aplicacao recebe a solicitacao de iniciacao da AS do
```

```
ISA *)
    DEL; (* Aplicacao envia a solicitacao de iniciacao *)
    ACC; (* Aplicacao recebe a solicitacao de iniciacao *)
    isa_res !DATA; (* ISA recebe a resposta da AS *)
    USER_B [isa_ind, DEL, ACC, isa_res] (RAS)(* retorna ao estado inicial *)
endproc


process USER_C [isa_ind, DEL, ACC, isa_res] (DATA: data_type)  : noexit :=
    isa_ind ?DATA: data_type; (* Aplicacao recebe a solicitacao de iniciacao da AS do
ISA *)
    DEL; (* Aplicacao envia a solicitacao de iniciacao *)
    ACC; (* Aplicacao recebe a solicitacao de iniciacao *)
    isa_res !DATA; (* ISA recebe a resposta da AS *)
    USER_C [isa_ind, DEL, ACC, isa_res] (RAS)(* retorna ao estado inicial *)
endproc


process ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PDU: pdu_type) : exit :=
isa_req ?DATA: data_type;
tp_datareq1 !PDU;
sendauth [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (PAS, AUTH)
where
  process sendauth [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (DATA: data_type,
PDU: pdu_type) : exit :=
  tp_dataind1  ?PDU: pdu_type;
  tp_datareq1 !DATA !PDU;
  sendneg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (RAS)
  where
    process sendneg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (DATA: data_type) :
exit :=
    tp_dataind1  ?DATA: data_type ?PDU: pdu_type;
    isa_cnf !DATA;
```

```
    fineg [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (NEG)
    where
      process fineg [isa_req, isa_cnf, tp_datareq1, tp_dataind1](PDU: pdu_type) : exit
:=
      tp_datareq1 !PDU;
      tp_dataind1 ?PDU: pdu_type;
      ISA_A [isa_req, isa_cnf, tp_datareq1, tp_dataind1] (INIT)
      endproc
    endproc
  endproc
endproc


process ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PDU: pdu_type) : exit :=
tp_dataind2 ?PDU: pdu_type;
tp_datareq2 !PDU;
sendinitres [isa_ind, isa_res, tp_datareq2, tp_dataind2](PAS)
where
  process sendinitres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (DATA:data_type) :
exit :=
  tp_dataind2 ?DATA: data_type ?PDU: pdu_type;
  isa_ind !DATA;
  sendauthres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (RAS, AUTH_RES)
  where
    process sendauthres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (DATA: data_type,
PDU:pdu_type) : exit :=
    isa_res ?DATA:data_type;
    tp_datareq2 !DATA !PDU;
    sendnegres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (NEG_RES)
    where
      process sendnegres [isa_ind, isa_res, tp_datareq2, tp_dataind2] (PDU: pdu_type) :
exit :=
      tp_dataind2 ?PDU: pdu_type;
```

```
        tp_datareq2 !PDU;
        ISA_B [isa_ind, isa_res, tp_datareq2, tp_dataind2] (INIT_RES)
        endproc
      endproc
   endproc
endproc


process TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2] : noexit :=
   (tp_datareq1 ?Data: data_type ?PDU: pdu_type;
   (( tp_dataind2 !Data !PDU;
    TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
   ) ))
   []
   (tp_datareq2 ?Data: data_type ?PDU: pdu_type;
   (( tp_dataind1 !Data !PDU;
    TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
   ) ))
   []
   (tp_datareq1 ?PDU: pdu_type;
   (( tp_dataind2 !PDU;
    TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
   ) ))
   []
   (tp_datareq2 ?PDU: pdu_type;
   (( tp_dataind1 !PDU;
    TP [tp_datareq1, tp_dataind1, tp_datareq2, tp_dataind2]
     ))
   )
endproc

endspec
```