UM MODELO DE ARQUITETURA DE SEGURANÇA PARA TINA APLICADO A CONTABILIDADE

Morvan D. Müller, Desire Nguessan, Carlos B. Westphall, Abderrahim Sekkaki
morvan, desire, westphal, sekkaki@lrg.ufsc.br

Laboratório de Redes e Gerência (LRG) - Centro Tecnológico (CTC)
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476, CEP 88040-970, Tel: +55.48.3319739
Florianópolis, SC, Brasil

ABSTRACT

TINA has been thought in the goal to offer an universal vision of Telecommunications Services, it answers to the increasing needs of fast developments of news services (e.g., multimedia, multi-party conference, etc.) in heterogeneous networks environments (e.g., fixe, mobile and multimedia) with interaction of several actors. However, the management context functionality (FCAPS management functions) are still an initial state of research and development.

In this paper we discuss requirements and security services for TINA management context, in particularly for TINA accounting management architecture. Then we propose and implement a model of security management middleware architecture, based on secure objects, cryptography and ticket distributed. To provide security services, secure objects which have security functionality like authentication and access control, have been defined in security domain. Secure objects in our model are CORBA objects. Security domain, which is also called SBS (Security Base Server), provides security services and has SMIB (Security Management Information Base) that contains security policies, cryptographic algorithms and other relevant information.

RESUMO

TINA foi desenvolvida com a meta de oferecer uma visão universal de serviços de telecomunicações, ela responde ao incremento de rápidos desenvolvimentos de novos serviços (multimídia, conferencia multiponto e etc.) em ambientes de redes heterogêneos (fixo, móvel e multimídia) com interações diversas. Todavia, a funcionalidade do gerenciamento de contextos (funções de gerenciamento FCAPS) são ainda um estado inicial de pesquisa e desenvolvimento.

Neste paper discutiremos os requisitos e serviços de segurança para o contexto de gerenciamento na TINA, em particular no gerenciamento da contabilidade. Propomos e implementamos um modelo de arquitetura de gerenciamento de segurança, baseado em objetos seguros, criptografia e distribuição de etiquetas. Para prover serviços com segurança, objetos seguros os quais tem funcionalidades de segurança, como autenticação e controle de acesso, são definidos como um domínio seguro, representados no nosso modelo por objetos CORBA. Um domínio seguro, também chamado SBS - Security Base Server (Servidor da Base de Segurança), provê serviços seguros e a SMIB - Security Management Information Base (Base de Informações do Gerenciamento da Segurança) contém políticas de segurança, algorítmos de criptografia e outras informações relevantes.

Palavras Chave : Segurança, Contabilidade, TINA e CORBA.

1. INTRODUÇÃO

Em geral, os sistemas que são baseados em ambientes de computação distribuídos aperfeiçoam eficiência e a reusabilidade de componentes, porém eles são mais vulneráveis a violação da segurança do que sistemas tradicionais porque as informações estão abertas e distribuídas. Por isso, sistemas que rodam em ambientes distribuídos requerem frameworks seguros ou mecanismos que implementem um ambiente distribuído seguro dentro da contabilidade [01]. Alguns tipos de frameworks seguros suportam plataformas abertas e distribuídas como Kerberos, POSIX, ISO7498-2, todavia, estes elementos não suportam objetos distribuídos como CORBA (Common Object Request Broker Architecture) e TINA (Telecommunication Information Networking Architecture)[13].

CORBA, é uma especificação para uma arquitetura padrão orientada a objetos que é definida pelo OMG (Object Management Group). CORBA possibilita objetos a interagir com outros independente da sua localização, tipo de computador hospedeiro ou linguagem de programação. Ela aperfeiçoa a escalabilidade do sistema, incrementando o reuso de software, facilidade de distribuição, linguagem de implementação independente e combina a tecnologia de computação distribuída com orientação a objetos. As vantagens dos serviços CORBA como segurança e notificação de mensagem podem também ajudar a acelerar o desenvolvimento das aplicações [08]. CORBA ainda tem falhas quanto a operar em ambientes de tempo real com alta tolerância a falhas, conforme é esperado em sistemas de telecomunicações.

Por outro lado, TINA define uma arquitetura global para telecomunicações baseadas no avanço da tecnologia de software. Ela apresenta independência da tecnologia de transporte: serviço de conectividade. Os serviços que rodam nesta plataforma (conferências, multimídia ou recuperação de informações) podem, sem diferença, usar ATM (Asynchronous Transfer Mode) ou IP (Internet Protocol) para a conectividade, baseados em qualidade de serviço (QoS) requerido pelo usuário (por exemplo, imagem com qualidade para streams de vídeo).

Sistemas CORBA podem ser adaptados aos serviços da arquitetura TINA. Neste sentido, nós propomos um modelo de

gerenciamento de segurança baseado em objetos CORBA e integrado a TINA. Nós validamos o modelo aplicando-o ao gerenciamento da contabilidade dentro da TINA [06].

Inicialmente apresentaremos os requisitos de segurança no domínio TINA, discutiremos sobre serviços e mecanismos de segurança e apresentaremos considerações de segurança na contabilidade. Depois vamos descrever o modelo de segurança baseado em objetos integrados a contabilidade da TINA e por final apresentaremos a conclusão deste trabalho.

2. DOMÍNIO SEGURO NA TINA

Na TINA, requisitos de segurança são definidos para quatro áreas: segurança de serviços e sistemas de telecomunicações , segurança de sistemas de gerenciamento, segurança de ambientes e serviços DPE (Distributed Process Environment) e ainda gerenciamento da segurança [02].

Segurança de Serviços e Sistemas de Telecomunicações envolve a proteção de elementos de rede, tráfego de rede, sinalização e serviços. Mecanismos de segurança em redes e serviços asseguram contra fraudes, revelação de informação e negação de serviços.

Segurança no Gerenciamento de Sistemas envolve a proteção de processos que são parte do sistema gerenciador para prover administração, operações de telecomunicações e manutenção da funcionalidade. Isto assegura contra ataques ao sistema de gerenciamento e contra as informações transferidas entre o sistema de gerenciamento e os elementos da rede.

Segurança de Ambientes e Serviços DPE envolve a proteção de objetos, desenvolvimento, interoperabilidade e funcionalidade provida pelo ambiente DPE. Isto inclui proteção de interface de objetos, interação, ciclo de vida das operações, persistência e QoS. Pode ser implementada nas diferentes camadas como aplicação, rede ou transporte [12].

Gerenciamento da Segurança preocupa-se com o gerenciamento de serviços de segurança e mecanismos para ambas as áreas. Isso envolve a administração das informações de autenticação, relatórios, auditoria e mecanismos de recuperação. Para prover esse gerenciamento da segurança, serviços e mecanismos de segurança precisam ser definidos.

3. SERVIÇOS E MECANISMOS DE SEGURANÇA

Para prover segurança é necessário especificar serviços e mecanismos de segurança. Os seguintes serviços de segurança são necessários [01]:

- Identificação e autenticação dos usuários para verificar se eles são quem dizem ser.
- Autorização e controle de acesso para decidir se um usuário pode acessar um objeto.
- Segurança de comunicação entre objetos, os quais requerem o estabelecimento de associações seguras, integridade e confidencialidade de mensagens.
- Não repúdio para provar a origem e recepção de dados.
- Auditoria de segurança para fazer contabilidade de usuários e de suas ações relacionadas com segurança.
- Distribuição de chaves para suportar serviços de segurança usando mecanismos especiais de segurança [01][11].

Estes serviços de segurança podem ser implementados com uma combinação de mecanismos de segurança como criptografia que provê confidencialidade de dados ou controle de tráfego de informações, mecanismos de assinatura digital para prevenção de negação de serviços (deny of services) ou autenticação de usuários, mecanismos de controle de acesso para garantir os direitos de acesso do usuário na SMIB (Security Managment Information Base) e mecanismos de integridade de dados que podem garantir a integridade comparando os dados gerados pelo remetente com os dados recebidos pelo receptor.

4. CONSIDERAÇÕES DE SEGURANÇA EM CONTABILIDADE

Em Sistemas distribuídos a segurança é essencial e parte natural do contexto, obviamente a contabilidade não é uma exceção. A segurança na contabilidade implica em garantia, confiabilidade e integridade nas informações da contabilidade e outras.

- *Garantida na contabilidade* refere-se a transação de serviços que no ideal deveriam oferecer um mecanismo para garantir a integridade do serviço. Isto significa que "você não paga se não concorda com o serviço".
- *Confiabilidade na Contabilidade* significa que a informação da contabilidade deve ser confiável. Este requisito vem do fato da TINA ser um padrão aberto. Um usuário e um servidor, os quais são totalmente desconhecidos, podem se conectar usando um serviço de páginas amarelas on-line ou usando um trader (situação onde mecanismo de segurança é urgentemente necessário). Como pode o usuário confiar no serviço provido e como pode o serviço provido confiar no usuário? A primeira questão é relacionada com gerenciamento na contabilidade e a segunda é relacionada mais com gerenciamento da segurança. Ela é similar ao caso onde a companhia telefônica manda um telefone desconhecido na cobrança baseando-se no seus serviços. Plataforma aberta não é necessariamente uma coisa boa, a menos que ambos usuário e provedor de serviço estejam protegidos. A informação na contabilidade deve ser confiável e deve estar apta para ser gravada em ambos os lados, de maneira não modificável e não refutável.
- A integridade das informação na contabilidade deve ser preservada mesmo sob falhas da rede e corrompimento de serviços, considerando que o serviço possa atravessar diferentes domínios de gerenciamento.

5. MODELO DE SEGURANÇA BASEADO EM OBJETOS CORBA

5.1 Componentes Seguros

No modelo sendo proposto as funções de segurança são executadas pelo Security Base Server (SBS). Ele provê o serviço

de autenticação, distribuição de etiquetas, controle de acessos e a SMIB (Security Management Information Base). Estes serviços são implementados por objetos CORBA. Quando um usuário invoca um serviço de um objeto, há uma requisição e uma resposta interativa entre eles. Assim para prover serviços de segurança, objetos seguros com funcionalidades de segurança (no nosso modelo Interceptor) devem interceptar a requisição e a resposta e ao mesmo tempo o mecanismo deve ser transparente ao usuário. A maioria das funções de segurança são feitas pelo SBS durante a invocação do objeto alvo. A figura 1 apresenta a arquitetura geral do modelo com os componentes do SBS.

O serviço de Autenticação provê identificação e autenticação do usuário através da informação de autenticação fornecida pela aplicação do usuário e da SMIB. A autenticação é encriptada com uma chave que é gerada no usuário, no SBS e no SI (Security Interceptor), então o usuário é mapeado na SMIB e uma etiqueta encriptada é gerada e enviada ao usuário.

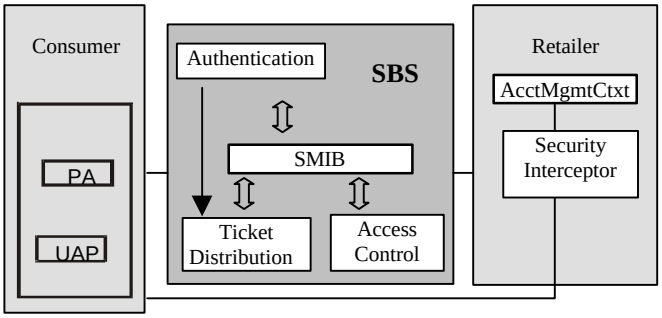


Figura 1. Arquitetura do SBS (Security Base Server)

O serviço de controle de acesso fornecido pelo SBS provê autorização e controle de acesso para decidir se um usuário pode acessar um objeto. O processo é baseado nas permissões do cliente (atributos de segurança do usuário, campos de controle, serviços e outras informações relevantes que o usuário pode acessar). Veja a figura 2.

A figura 2 dá uma visão dos processos, interações e controle da autenticação, autorização e distribuição de etiquetas.

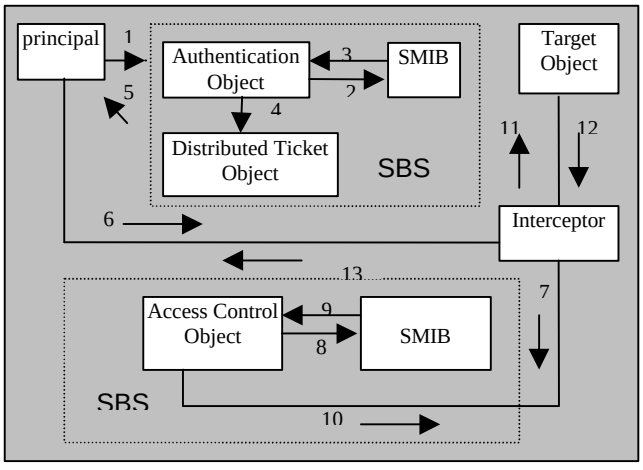


Figura 2. Fluxo das informação de autenticação, autorização e distribuição de etiquetas.

Descrevemos abaixo o cenário dos processos entre os componentes de segurança:

1. O domínio do usuário requisita ao objeto Authentication Object (Objeto de Autenticação) para identificar e autenticar o usuário (principal), encripta as informações, como identificação e senha do usuário e envia para ele.

2. O objeto Authentication Object decripta a informação recebida e acessa a SMIB para obter dados sobre o usuário.

3. Os dados do usuário gravados na SMIB são retornados ao objeto Authentication Object.

4. O objeto Authentication Object compara a informação recebida do domínio do usuário (principal) com os dados do usuário gravados na SMIB. Se eles são iguais, então o usuário é quem diz ser, e neste caso, contata o objeto Distributed Ticket Object (Objeto Distribuidor de Etiquetas).

5. Uma transação de etiqueta é gerada pelo objeto Distributed Ticket Object e enviada ao usuário. Esta etiqueta é a identidade do usuário, a qual é encriptada com uma chave conhecida pelo usuário.

6. O usuário usa esta etiqueta como sua identificação para invocar os objetos (serviços implementados pelos objetos). Ela é necessária para realizar qualquer transação, sendo que a requisição sempre estará encriptada.

7. No domínio do retailer (terceirizado do provedor), a requisição do usuário é interceptada pelo objeto Interceptor (Interceptador) o qual envia uma cópia da requisição para o objeto Access Control Object (Objeto de Controle de Acesso).

8. O objeto Access Control Object decripta a informação recebida e acessa a SMIB para obter informações e dados a respeito do usuário, como as permissões do mesmo.

9. Os dados do usuário gravados na SMIB são enviados ao objeto Access Control Object.

10. O objeto Access Control Object verifica se o usuário tem permissão de invocar o serviço fornecido pelo objeto, em caso positivo, ele envia uma confirmação (acknowledge) ao objeto Interceptor.

11. O objeto Interceptor envia a requisição do usuário ao objeto Target Object (objeto alvo).

12. O objeto Target Object processa a requisição, encripta a resposta e a envia ao objeto Interceptor.

13. Finalmente, o objeto Interceptor envia a resposta encriptada ao usuário requisitante.

No nosso modelo, o SBS atua como um gerenciador de segurança que controla e gerencia o contexto da segurança e as interações entre o consumidor (consumer) e o retailer (terceirizado do provedor). O Security Interceptor (SI) é um agente de segurança que ajuda no controle de acesso dos serviços dentro do domínio do retailer, incluindo o contexto de gerenciamento da contabilidade (AcctMgmtCtxt - Accounting Management Context).

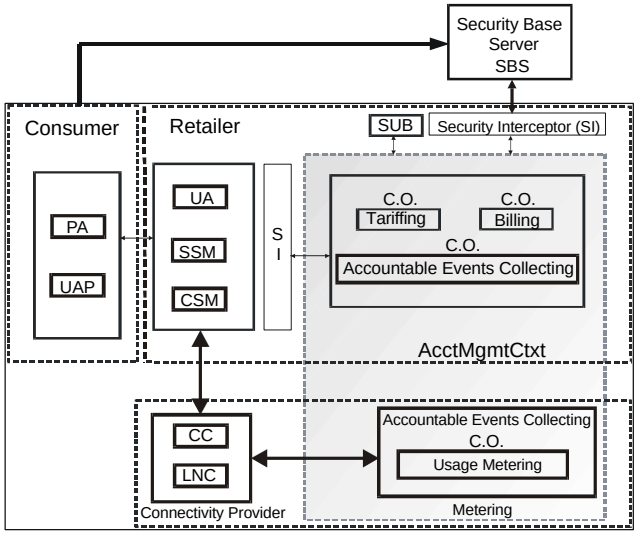
5.2 Security Base Server (SBS) e a contabilidade

O AcctMgmtCtxt (Contexto de Gerenciamento da Contabilidade) tem os seguintes componentes: **Accountable Event Collecting** (Coletor de Eventos de Contabilidade) – recebe e coleta eventos de contabilidade e coleta eventos de contabilidade associados com o estado da sessão. **Tariffing** (Tarifador) – ele fornece mudança de taxas (charging-rate) em função do estado corrente da sessão, e acumula as mudanças correntes em função dos eventos de contabilidade. **Billing** (Cobrador) – ele atua no final do período de cobrança conforme negociado e definido no contrato dos usuários. **Usage Metering** (Medidor de Uso) – durante o tempo de vida da conexão o componente medidor de uso coleta e controla a aquisição de dados pelo uso dos recursos da rede [06].

Esses componentes são vistos pelo SBS como serviços de objetos, dessa forma eles precisam de transações seguras. O SI presente no domínio do retailer intercepta as chamadas ao AcctMgmtCtxt e provê segurança.

A figura 3 ilustra a relação entre o SBS e o AcctMgmtCtxt definido em [06] e a arquitetura de serviço da TINA.

A descrição de cada um dos componentes da TINA podem ser encontrados em TINA Service Component Specification [07].



CC – Connection Coordinator
C.O. – Context Object
CSM – Communication Session Manager
LNC – Layer Network Coordinator
PA – Provider Agent
SBS – Security Base Server
SI – Security Interceptor
SSM – Service Session Manager
SUB – Subscription Management Component
UA – User Agent
UAP – User Application

Figura 3. Relação entre AcctMgmtCtxt e Security Base Server (SBS)

6. VISÃO GERAL DO SBS

Todas as requisições enviadas ao provedor (provider) são interceptadas transparentemente pelo Security Interceptor (SI) que as redireciona ao SBS para proceder a autorização do usuário através de verificação de sua identidade e permissões na SMIB. Dados trocados entre o provedor e o usuário serão encriptados para garantir a segurança num ambiente distribuído (DPE - Distributed Process Environment) como a TINA. O SBS tem cinco módulos principais:

- Authentication Management (AuthMgmt)
- Access Control Management (AccessContMgmt)
- Ticket Distribution (TicketDistrib)
- Security Management Information Base (SMIB)
- Security Interceptor (SecInterceptor)

Cada um desses módulos tem funções específicas, conforme descrito abaixo.

6.1 Authentication Management (AuthMgmt)

Nesse módulo a primeira fase é decriptar as informações enviadas pela aplicação do usuário. Na Segunda fase, o usuário é autenticado pela verificação dos seus dados na SMIB, após ser autenticado, uma etiqueta de transação é gerada pelo módulo TicketDistrib e enviado ao usuário final. Para incrementar o nível de segurança, essa etiqueta é encriptada com uma chave (common key) conhecida pelas três partes (usuário, SBS e o Security Interceptor).

Em todas as transações futuras entre o usuário e o provedor o usuário usa esta etiqueta como sua identificação. Essa etiqueta e algumas informações relevantes do usuário são gravadas na SMIB para verificações futuras. A figura 4 mostra as interfaces do SBS e a implementação do serviço de autenticação. Nós usamos IDL (Interface Language Definition) para definir estas interfaces.

```
// ----- BEGIN Module
module SBS {
typedef sequence< octet > Ticket;
exception NotAuthenticated
string Decrypt ( in string CryptText, in string CommonKey)
string Encrypt ( in string Text, in string CommonKey)
exception NotAuthorized{};
interface Authmgmt {    // Authentication Service
     Ticket Authenticate( in string userId,  in string passWord )
     raises( NotAuthenticated );
 };
interface AccessContMgmt { // Authorisation Service
void Authorize(   in Ticket clientTicket, in Object targetObject,
                  in string operationName  )
     raises( NotAuthorized );
};
Interface TicketDistrib {// Ticket Distributed Service
     Ticket GenerateTicket( in string UserId,  in string Password );
};
Interface SecurityBasicServer: Authmgmt, AccessContMgmt ,
     TicketDistrib {}
}
```

```
public byte[] Authenticate(  String userIdEncryted,
                             String  passWordEncrypted )
     throws NotAuthenticated {
     userID = Decrypt (userIdEncryted, CommonKey)
     passWord = Decrypt (passWordEncrypted, CommonKey)
     if( userTable.containsKey( userId ) &&
          userTable.get( userId ).equals( passWord ) ) {
          ...
          return  GenerateTicket( userId) ;
     }
     else { throw new NotAuthenticated();  }
}
// ----- END Module
```

Figura 4. Especificações IDL do SBS e implementação da Autenticação.

6.2 Access Control Management (AccessContMgmt)

O módulo Security Interceptor (SI) intercepta todas as transações entre o usuário e o provedor. Ele redireciona as requisições do usuário ao módulo Access Control Management e as respostas do provedor ao usuário. Para realizar essas interações em um contexto de segurança, todos os dados transmitidos são encriptados para garantir a confidencialidade.

Na primeira fase, o módulo AccessContMgmt decritpta a etiqueta, verifica a sua validade na SMIB e verifica se o usuário tem direito de invocar o serviço do objeto requisitado. Em caso afirmativo, uma confirmação (acknowledgement) é retornada ao módulo SecInterceptor (SI) e o serviço é executado pelo provedor. A figura 5 mostra as especificações IDL do objeto Access Control Management.

```
// ----- BEGIN Module
module SBS {
typedef sequence< octet > Ticket;
exception NotAuthenticated
string Decrypt ( in string CryptText, in string CommonKey)
string Encrypt ( in string Text, in string CommonKey)
exception NotAuthorized{};
interface Authmgmt{          // Authentication Service
     Ticket Authenticate( in string userId, in string passWord )
     raises( NotAuthenticated );
};
 interface AccessContMgmt {    // Authorisation Service
     void Authorize( in Ticket clientTicket,
                     in Object targetObject,
                     in string operationName )
     raises( NotAuthorized );
};
Interface TicketDistrib {// Ticket Distributed Service
     Ticket GenerateTicket(  in string UserId,
                             in string Password );
};
Interface SecurityBasicServer : Authmgmt, AccessContMgmt ,
     TicketDistrib {}
}
```

```
void Authorize(   byte[] ticket, org.omg.CORBA.Object
                  object, String operationName )
     throws NotAuthorized {
          String userId = Decrypt( ticket );
          return;
     }
     if (accessControlList.containsKey (userId)) {
          String allowedObjects = (String) accessControlList.get(
userId );
          if( allowedObjects.regionMatches( 0, "*", 0, 1 ) ) {
            return;
          }
          if(          allowedObjects.regionMatches(0,objectId,0,
            OjectId.length() ) ) {
            return;
          }
     }
     throw new NotAuthorized();
}
// ----- END Module
```

Figura 5. Especificações IDL do SBS e implementação da Autorização.

6.3 Ticket Distribution (TicketDistrib)

Esse módulo basicamente gera uma etiqueta de transação, dinamicamente, para identificar o usuário. Essa etiqueta é encriptada com a chave comum (common key) usando métodos avançados de criptografia [01][10].

```
Interface TicketDistrib {
     Ticket GenerateTicket( in string UserId, in string Password );
};
```

Figura 6. Especificações IDL do objeto Ticket Distributed

6.4 Security Management Information Base (SMIB)

Esse módulo é a base de dados para o SBS. Ele contém políticas de acesso e atributos do usuário como identificação e permissões (permissões são objetos e serviços que o usuário pode acessar).

6.5 Security Interceptor (SecInterceptor)

O objeto Security Interceptor (SI) reside no domínio do retailer (terceirizado do provedor) e intercepta todas as transações entre o usuário e o provedor. Ele faz referência ao SBS, redireciona as requisições do usuário ao módulo Access Control Management (AccContMgmt) para validação e espera por uma confirmação (acknowledgement). Em seguida o Security Interceptor (SI) envia as requisições para serem executadas pelo provedor, intercepta a resposta, encripta e envia ao usuário final.

Como o SBS e o usuário final (principal), o módulo Security Interceptor conhece as regras para encriptar e decriptar as requisições. O mecanismo de interceptação (Interceptor) é

importante na nossa plataforma sendo transparente e natural para o sistema.

7. CONCLUSÃO

Nós discutimos sobre características de segurança e descrevemos requisitos, serviços e mecanismos de segurança. Propomos e implementamos um modelo de arquitetura de gerenciamento de segurança baseado em segurança de objetos, autenticação, autorização, criptografia e distribuição de etiquetas. Esse modelo pode ser adaptado a sistemas baseados em TINA e foi validado em um protótipo de gerenciamento de contabilidade. A integridade e características de não-repúdio não foram tratados nesse modelo podendo ser objeto para estudos futuros.

8. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Nguessan, D., A Model of Security Management for Distributed Objects. Mester Thesis. UFSC-CPGCC. Florianópolis, Abril 2000.

[2] Hahn J.H, Lee K. H, Ryou J.C, A Model of Security Platform within TINA_based Management System, APNOMS, 16-18 Setembro 1998, Sendai, Japão.

[3] OMG, Documento Número, 98-12-09, Security Service Specification in CORBA Services: Common Object Services Specification, 1998.

[4] Brennan R., Jennings B, McArdle C, e Curran T, Teletec Irland, Evolutionary Trends in Intelligent Networks, IEEE Communications Magazine, Junho 2000.

[5] Mampaey M., Couturier A. , Alcatel, Using TINA Concepts for Evolution, IEEE Communications Magazine, Junho 2000.

[6] Sekkaki, A.; Alvarez, L. M. C.; Watanabe, W. T.; Westphall, C. B. Development of a Prototype Based on TINA Accounting Management Architecture. Submetido ao IM2001. Seattle (Washington) - USA, 14-18 Maio 2001.

[7] TINAC, Service Component Specification, Version 1.0b, Janeiro 19, 1998 http://www.tinac.com

[8] OMG, Documento Número 98-12-09, Corba/IIOP 2.2 Specification.

[9] Peter A. Loscocco, Stephen D. S., Patrick A. M., Ruth C. T., S. Jeff T., e John F. Farrell, The Inevitable of Failure: The Flawed Assumption of Security in Modern Computing Environments, Na 21ª conferência National Information Systems Security, na cidade de Cristal, Virginia, Outubro 1998. National Security Agency, NISSC, htttp://www.jya.com/paperF1.htm.

[10] William Stallings, Cryptography and Networks Security. Principals and pratices, segunda edição.

[11] Westphall, C. M. et. Al,: Authorization Schemes for Large-Scale, Systems based on Java, Corba and Web Security Models, ICON99 – The IEEE International Conference on Networks, Setembro 28 a Outubro 1, pp 327 – 334, Bisbane – Queensland , Australia.

[12] [Steegmans et al., TINA Network Resource Architecture 1997 http://www.tinac.com.

[13] Pavlou G. et al., Issues in Realizing the TINA Network Resource Architecture, Interoperable Communication Networks Journal, vol. 2, No.1, Março 1999.