UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

Jesseildo Figueredo Gonçalves

Desenvolvimento de uma Infraestrutura de Segurança para o Middleware MobileHealthNet

São Luís
2013

Jesseildo Figueredo Gonçalves

Desenvolvimento de uma Infraestrutura de Segurança para o Middleware MobileHealthNet

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como requisito parcial para a obtenção do grau de MESTRE em Engenharia de Eletricidade.

Orientador: Francisco José da Silva e Silva

Prof. Doutor em Ciência da Computação

Universidade Federal do Maranhão

São Luís

2013

Gonçalves, Jesseildo Figueredo.

Desenvolvimento de uma infraestrutura de segurança para o *Middleware* MobileHealthNet / Jesseildo Figueredo Gonçalves – São Luís, 2013.

98 f.

Impresso por computador (fotocópia).

Orientador: Francisco José da Silva e Silva.

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia de Eletricidade, 2013.

1. Redes Sociais. 2. Segurança. 3. Comunicação. I. Título.

CDU 004:392. 73

DESENVOLVIMENTO DE UMA INFRAESTRUTURA DE SEGURANÇA PARA O MIDDLEWARE MOBILEHEALTHNET

Jesseildo Figueredo Gonçalves

Dissertação aprovada em 01 de agosto de 2013.

Prof. Francisco José da Silva e Silva ,Dr.
(Orientador)

Prof. Joaquim Celestino Júnior, Dr.
(Membro da Banca Examinadora)

Prof. Zair Abdelouahab, Ph.D.
(Membro da Banca Examinadora)

Dedico este trabalho aos meus pais, meus irmãos, meus amigos e professores.

Resumo

Uma rede social pode ser definida como sendo uma estrutura social cujos membros se relacionam em grupos e cuja interação é realizada através de tecnologias da informação e comunicação, permitindo a quebra de barreiras geográficas e temporais no acesso ao conteúdo e interações entre pessoas. Nas Redes Sociais Móveis, os usuários utilizam dispositivos de computação portáteis com acesso a tecnologias de comunicação sem fio para realizar suas interações através da rede social, agregando mobilidade ao usuário e ricas informações de contexto. Na área da saúde, uma rede social pode ser definida como um grupo de pessoas (e a estrutura social que elas coletivamente constroem) que utiliza tecnologias da informação e comunicação com o propósito de conduzir coletivamente ações relacionadas à assistência médica e sua educação.

Este trabalho está inserido no contexto do projeto MobileHealthNet, desenvolvido em parceria pelo Laboratório de Sistemas Distribuídos da Universidade Federal do Maranhão e o *Laboratory for Advanced Collaboration* da Pontifícia Universidade Católica do Rio de Janeiro. O MobileHealthNet tem por objetivo desenvolver um *middleware* que permita a construção de redes sociais móveis e facilite o desenvolvimento de serviços colaborativos para o setor da saúde, a troca de experiências e a comunicação entre pacientes e profissionais da saúde, além de uma melhor gestão dos recursos da saúde por órgãos governamentais. Neste trabalho de mestrado, é proposto um modelo de segurança e privacidade adotado pelo projeto MobileHealthNet. Uma outra contribuição deste trabalho consiste em uma solução para criação de canais seguros de comunicação para sistemas baseadas no OMG DDS (Data Distribution Service), uma especificação para comunicação *publish* e *subscribe* centrada nos dados desenvolvida pela OMG (*Object Management Group*), adotada no projeto MobileHealthNet.

Palavras-chave: Redes Sociais, Redes Sociais Móveis, Segurança, Comunicação.

Abstract

A social network can be defined as a social structure whose members are related in groups and whose interaction is accomplished through information and communication technologies, overcoming geographic and temporal obstacles to access content and perform interactions between people. In Mobile Social Networks, users use portable computing devices with access to wireless communication technologies to perform their interactions through the social network, adding the possibility of user mobility and the access to rich context information. In health care, a social network can be defined as a group of people (and the social structure they collectively construct) that uses information and communication technologies in order to collectively conduct actions related to health care and health education.

This work is inserted in the context of the MobileHealthNet project, developed in partnership by the Distributed Systems Laboratory at the Universidade Federal do Maranhão and the Laboratory for Advanced Collaboration of the Pontifícia Universidade Católica do Rio de Janeiro. The MobileHealthNet aims to develop a middleware that allows the establishment of mobile social networks and facilitate the development of collaborative services focusing the health care domain, the exchange of experiences between patients and health professionals, and a better management of health resources by government agencies. In this master thesis, we propose a security and privacy model to the MobileHealthNet project. Another contribution of this work consists of a solution for creating secure communication channels based on OMG DDS (Data Distribution Service), a data centered specification for publish/subscribe communication developed by the OMG (Object Management Group), adopted in the MobileHealthNet project.

Keywords: Social Network, Mobile Social Network, Security, Communication.

Agradecimentos

Deixo aqui os meus agradecimentos às pessoas que tanto me ajudaram nesta caminhada.

A Deus, por sua infinita misericórdia e graça.

Aos meus pais, José Reginaldo e Valdenira Gonçalves, e irmãos por terem me apoiado durante esses anos.

Ao Prof. Dr. Francisco José da Silva e Silva, por sua paciência e orientações durante toda minha pesquisa e produção deste trabalho.

Aos professores Joaquin Celestino Júnior e Zair Abdelouahab, por aceitarem fazer parte da banca avaliadora.

Aos colegas de laboratório pela ajuda e apoio que todos prestaram na produção deste trabalho.

À UFMA e ao Programa de Pós-Graduação em Engenharia de Eletricidade (PPGEE), pela oportunidade e estrutura oferecida para a conclusão deste trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte durante o desenvolvimento deste trabalho.

"O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis."

José de Alencar

Lista de Figuras

2.1 Definição de Redes Sociais Móveis 22

2.2 Arquiteturas de Redes Sociais Móveis [49] 22

2.3 Arquitetura do MobiSoc [11] 25

2.4 Arquitetura do Mobilis [33] 28

2.5 Arquitetura do MyNet [26] 30

2.6 Arquitetura do MobiClique [37] 31

3.1 Arquitetura Geral do MobileHealthNet 37

3.2 Modelo de Segurança Aplicado ao MobileHealthNet 44

4.1 Arquitetura da Infraestrutura de Comunicação do MobileHealthNet . . 59

4.2 Componentes de Segurança Dispostos no Domínio DDS 65

4.3 Protocolo de Autenticação e Distribuição de Chaves Simétricas 66

4.4 Classes que Implementam os Mecanismos de Segurança 73

4.5 Diagrama de Classes Usadas na Geração de Filtros 74

4.6 Classes Usadas na Criptografia de Dados 77

5.1 Resultados de Throughputs 85

5.2 Resultados do RTT 86

5.3 Taxa de Perdas de Mensagens 87

Lista de Tabelas

3.1 Resumo dos *Middleware* Analisados 54

5.1 Tabela de Experimentos 84

Lista de Siglas

API *Application Programing Interface*.

CFM *Conselho Federal de Medicina*.

CLASP *Comprehensive, Lightweight Application Security Process*.

DDL *Data Definition Language*.

DDS *Data Distribution Service*.

EJB *Enterprise Java Bean*.

JCA *Java Cryptography Architecture*.

JEE *Java Enterprise Edition*.

KDC *Key Distribution Center*.

MC-SRES *Manual de Certificação para Sistemas de Registro Eletrônico em Saúde*.

MobileHealthNet *Mobile Social Networks for Health Care in Offside Regions*.

OMG *Object Management Group*.

RBAC *Role-Based Access Control*.

RSMs *Redes Sociais Móveis*.

SBIS *Sociedade Brasileira de Informática em Saúde*.

SSL *Secure Sockets Layer*.

SSO *Single Sign-On*.

Sumário

Lista de Figuras **vi**

Lista de Tabelas **vii**

Lista de Siglas **viii**

1 Introdução **16**

1.1 Objetivos 18

1.2 Estrutura da Dissertação 19

2 Redes Sociais Móveis **21**

2.1 Introdução a Redes Sociais Móveis 21

2.2 Arquitetura de Redes Sociais Móveis 21

2.3 *Middleware* para Redes Sociais Móveis 24

2.3.1 MobiSoc 25

2.3.2 Mobilis 27

2.3.3 MyNet 29

2.3.4 MobiClique 31

2.4 Conclusão 32

3 Modelo de Segurança e Privacidade para o MobileHealthNet **34**

3.1 O Projeto MobileHealthNet 34

3.2 O *middleware* MobileHealthNet 36

3.3 Metodologia Usada na Construção do Modelo de Segurança 38

3.4 Requisitos para Construção do Modelo de Segurança 41

3.5 O Modelo de Segurança 43
3.5.1 Autenticação de Usuários e Single Sign-On 44
3.5.2 Gerenciamento de Privacidade dos Usuários 45
3.5.3 Autorização de Usuários 46
3.5.4 Gerenciamento de Identidades 47
3.5.5 Gerenciamento de Logs 48
3.5.6 Segurança da Comunicação 49
3.5.7 Exigências Legais e Éticas 50
3.6 Aspectos de Implementação do Modelo 50
3.7 Análise Comparativa da Segurança Provida por *Middleware* para Redes Sociais Móveis 52
3.8 Conclusão 54

4 Segurança da Comunicação no MobileHealthNet 56
4.1 Infraestrutura de Comunicação do MobileHealthNet 58
4.2 Uma Proposta para Segurança da Comunicação em Sistemas Baseados no DDS 62
4.2.1 Requisitos de Segurança para Comunicação no MobileHealthNet 62
4.2.2 Arquitetura da Solução Proposta 64
4.2.3 Protocolo de Autenticação e Distribuição de Chaves Simétricas 66
4.2.4 Gerenciamento das Chaves Simétricas 68
4.2.5 Escalabilidade da Solução 70
4.3 Aspectos de Implementação no MobileHealthNet 71
4.3.1 Criptografia dos Dados 76
4.3.2 Certificação Digital 77
4.4 Conclusão 79

5 Avaliação da Infraestrutura de Comunicação Segura do MobileHealthNet 81

5.1 Objetivos dos Experimentos 81
5.2 Carga de Trabalho e Métricas 81
5.3 Descrição dos Experimentos 83
5.4 Análise dos Resultados 84
5.5 Conclusão 88

6 Conclusão e Trabalhos Futuros **89**
6.1 Contribuições Cientificas 90
6.2 Trabalhos Futuros 91

Referências Bibliográficas **93**

1 Introdução

Mídias sociais são meios de comunicação utilizados para interação social. Kaplan e Haenlein em [27] definem mídia social como um grupo de aplicações baseadas na Internet construídas sobre fundamentos ideológicos e tecnológicos da Web 2.0, permitindo a criação e troca de conteúdo gerado pelo usuário. A Web 2.0 é um termo associado a aplicações cujo objetivo é facilitar o compartilhamento participativo de informações, interoperabilidade, projeto centrado no usuário e colaboração na *World Wide Web*. Um site Web 2.0 permite aos usuários a interação e colaboração uns com os outros em um diálogo de mídia social como criadores de conteúdos gerados por usuários em uma comunidade virtual.

Um dos tipos de mídia social que tem grande popularidade atualmente é a rede social, sendo esta uma estrutura social cujos membros se relacionam em grupos e cuja interação é realizada através de tecnologias da informação e comunicação [56]. Esta interação permite a quebra de barreiras geográficas e temporais devido ao acesso e interações simultâneas de diversas pessoas em qualquer lugar. Uma nova tendência das redes sociais são as *Redes Sociais Móveis* (RSMs), nas quais os usuários utilizam dispositivos móveis para se conectar à rede. Para Zhenyu et al., em [56], uma Rede Social Móvel é uma extensão das redes sociais, agregando mobilidade ao usuário e ricas informações de contexto. Nestas, indivíduos de interesses semelhantes ou comuns podem interagir, criando e visitando perfis, enviando mensagens, vendo fotos, vídeos, escutando músicas ou trocando informações em comunidades. A utilização das RSMs tem acompanhado o rápido crescimento da computação móvel. Os aparelhos celulares estão cada vez mais baratos e com mais recursos, permitindo a execução de aplicativos cada vez mais sofisticados, dentre eles os que disponibilizam acesso às RSMs. Pesquisas recentes mostraram que já em 2010 as redes sociais foram mais acessadas através de dispositivos móveis do que de computadores fixos [52].

Uma área de aplicação dos conceitos relacionados as redes sociais é a saúde. Segundo Upkar [54], *e-health* é definido como a aplicação das tecnologias de informação e comunicação através de todo escopo das funções envolvidas na prática e entrega de cuidados médicos. Neste contexto, RSMs podem ser utilizadas para

promover o intercâmbio de informações, colaboração e integração social entre os diversos agentes envolvidos no processo de atendimento à saúde. Uma rede social na área da saúde pode ser definida como um grupo de pessoas (e a estrutura social que elas coletivamente constroem) que utiliza tecnologias da informação e comunicação com o propósito de conduzir coletivamente ações relacionadas à assistência médica e sua educação [15]. Estas ações podem incluir o próprio provimento de serviços de saúde, a edução em saúde envolvendo profissionais e pacientes, uma plataforma para o suporte e discussão sobre questões e problemas relacionadas a tratamentos, compartilhamento de documentos, consultorias com especialistas e a manutenção do contato e relacionamento entre as pessoas envolvidas no processo de atendimento à saúde que se prolongue além dos encontros presenciais. Por exemplo, grupos de pesquisadores da área da saúde que interagem e trabalham em casos cooperativamente podem estabelecer uma rede social através da qual compartilha-se opiniões e recursos, como dados relativos a análise de casos e documentos científicos escritos colaborativamente.

Contudo, o desenvolvimento de RSMs é bastante complexo. Os desenvolvedores precisam atentar para questões relacionadas ao compartilhamento de dados na rede, a mobilidade dos usuários, escalabilidade, a disponibilização de mecanismos de interação síncrona e assíncrona entre pessoas, além de questões relacionadas à segurança e privacidade das informações. Grande parte das informações que são compartilhadas nas rede sociais exigem um rigoroso nível de privacidade, como é o caso das redes sociais voltadas para saúde, onde as informações compartilhadas na rede estão diretamente relacionadas ao tratamento de pacientes. Essa característica exige que o ambiente de rede social possua mecanismos destinados a garantir a segurança das informações compartilhadas na rede. Aaron Beach et al. [9] chamam atenção para um conjunto de problemas de segurança que uma RSM pode apresentar, enquanto que Hongyu Gao et al. [20] apresentam problemas comuns em redes sociais *on-line*, que também podem se manifestar em RSMs. A especificidade do domínio pode acrescentar diversos requisitos de segurança ao software. Por exemplo, no domínio da saúde deve-se seguir um conjunto de requisitos e normas legais definidas por entidades médicas e/ou governamentais que variam de acordo com o país em que o sistema será usado. Assim, a construção de uma RSM segura

requer mecanismos de segurança destinados a lidar com problemas e peculiaridades de ambientes de RSM, além do domínio de suas aplicações.

Em síntese, para se obter uma Rede Social Móvel segura, deve-se se atentar no mínimo para dois pontos chaves:

1. Segurança das Informações Compartilhados na Rede Social: manter a segurança e privacidade das informações compartilhadas na rede social, garantindo o controle de acesso aos usuários autorizados;

2. Segurança da Comunicação: garantir uma transferência de dados segura, mantendo a autenticidade, integridade e confidencialidade das mensagens.

Este trabalho está inserido no contexto do projeto *Mobile Social Networks for Health Care in Offside Regions* (MobileHealthNet), desenvolvido em parceria pelo Laboratório de Sistemas Distribuídos da Universidade Federal do Maranhão e o *Laboratory for Advanced Collaboration* da Pontifícia Universidade Católica do Rio de Janeiro, que tem por objetivo desenvolver um *middleware* que permita a construção de redes sociais móveis e facilite o desenvolvimento de serviços colaborativos para o setor da saúde, a troca de experiências e a comunicação entre pacientes e profissionais da saúde, além de uma melhor gestão dos recursos da saúde por órgãos governamentais. Este projeto conta com apoio institucional do Hospital Universitário da UFMA (HUUFMA). Em particular, duas unidades do HUUFMA estão diretamente envolvidas com o desenvolvimento do projeto: o Programa de Assistência a Pacientes Asmáticos (PAPA) e a Casa da Dor. O PAPA possui profissionais com especialidade em pneumologia e tem como objetivo principal o tratamento de pacientes portadores desta doença crônica. A Casa da Dor, por sua vez, possui profissionais especializados no tratamento de pacientes que sofrem com dores agudas, independentemente de sua etiologia. Portanto, o modelo de segurança proposto foi inicialmente implementado no *middleware* desenvolvido no contexto do projeto MobileHealthNet.

1.1 Objetivos

O objetivo geral deste trabalho é propor um modelo de segurança e privacidade destinado a uma infraestrutura de apoio ao estabelecimento de redes

sociais móveis cujas aplicações serão voltadas ao domínio da saúde, bem como o desenvolvimento dos mecanismos responsáveis pela comunicação segura entre os nós que compõem esta infraestrutura.

Os objetivos específicos desta pesquisa são:

- Investigar o estado da arte da segurança de redes sociais e redes sociais móveis;
- Identificar requisitos de segurança das redes sociais móveis, assim como os atuais problemas de segurança e privacidade enfrentados por estas;
- Propor um modelo de segurança e privacidade para redes sociais móveis voltado para saúde;
- Implementar os mecanismos de segurança destinado à comunicação segura entre os nós da rede;
- Avaliar o modelo de segurança junto ao *middleware*, de maneira a validá-lo conforme os requisitos de segurança estabelecidos.

1.2 Estrutura da Dissertação

Esta dissertação está organizada como segue:

- No Capítulo 2, é apresentado uma fundamentação teórica sobre alguns *middlewares* destinados ao desenvolvimento de redes sociais móveis, destacando suas respectivas arquiteturas;
- No Capítulo 3, é proposto um modelo de segurança para redes sociais móveis voltadas para saúde. Este modelo é apresentado como um conjunto de mecanismos necessários para garantir a segurança das informações compartilhadas nas redes sociais;
- O Capítulo 4 descreve uma solução de comunicação segura desenvolvida para o *middleware* MobileHealthNet, tendo como base o paradigma de comunicação *publish/subscribe*;

- O Capítulo 5 apresenta os resultados dos experimentos realizados sobre a infraestrutura de comunicação segura desenvolvida para o MobileHealthNet. A partir destes experimentos foram analisadas métricas importantes para infraestruturas de comunicação, buscando identificar o impacto causado pela inclusão dos mecanismos de segurança em seu desempenho;

- Por fim, o Capítulo 6 aborda as conclusões extraídas deste trabalho de mestrado, além de apresentar alguns trabalhos futuros que podem dar continuidade a esta pesquisa.

2 Redes Sociais Móveis

Este capítulo apresenta uma fundamentação teórica sobre Redes Sociais Móveis, definindo-as e classificando-as quanto à sua arquitetura. Em seguida, são abordados alguns exemplos de *middleware* destinados ao desenvolvimento de redes sociais móveis, tendo como foco os serviços disponibilizados por cada um deles e sua arquitetura.

2.1 Introdução a Redes Sociais Móveis

RSMs podem ser definidas como a combinação de três áreas de conhecimento [49]: Redes Sociais, Computação Móvel e Sistemas Cientes de Contexto - uma extensão do conceito de Kayastha et al. [29]. As Redes Sociais promovem funcionalidades para elaboração de perfis, representando entidades que socialmente se relacionam pela troca de informações. A Computação Móvel permite ao usuário a habilidade de estar sempre *on-line*, devido ao suporte a mobilidade promovido por dispositivos móveis e tecnologias de comunicação sem fio. Os Sistemas Cientes de Contexto permitem determinar a distribuição de informações a partir de sistemas embarcados nos dispositivos como localização atual do usuário, clima e temperatura local, pessoas na vizinhança e, também, inferir informações, como a ação a ser executada pelo usuário e sua intenção em fazê-la. Esta definição é ilustrada na Figura 2.1.

2.2 Arquitetura de Redes Sociais Móveis

Na literatura, embora seja possível encontrar diversas arquiteturas para RSMs que permitem o estabelecimento de várias comunidades de usuários móveis e suas interações, *softwares* de RSMs podem ser classificados em dois principais grupos: centralizados e distribuídos, como ilustrados na Figura 2.2. A escolha da arquitetura para construção de uma RSM tem grande impacto em sua aplicabilidade e serviços que

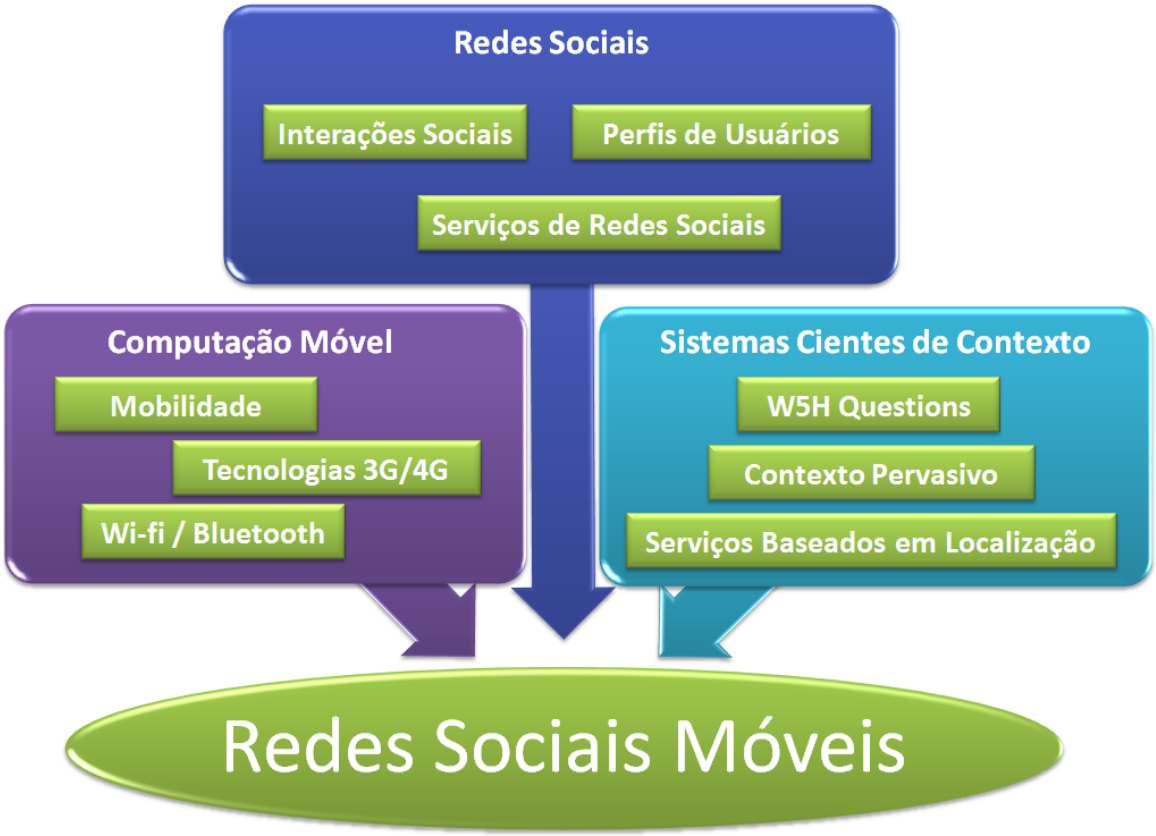

Figura 2.1: Definição de Redes Sociais Móveis

estarão disponíveis. A arquitetura também é decisiva para a escolha dos algoritmos que serão adotados na construção do software.

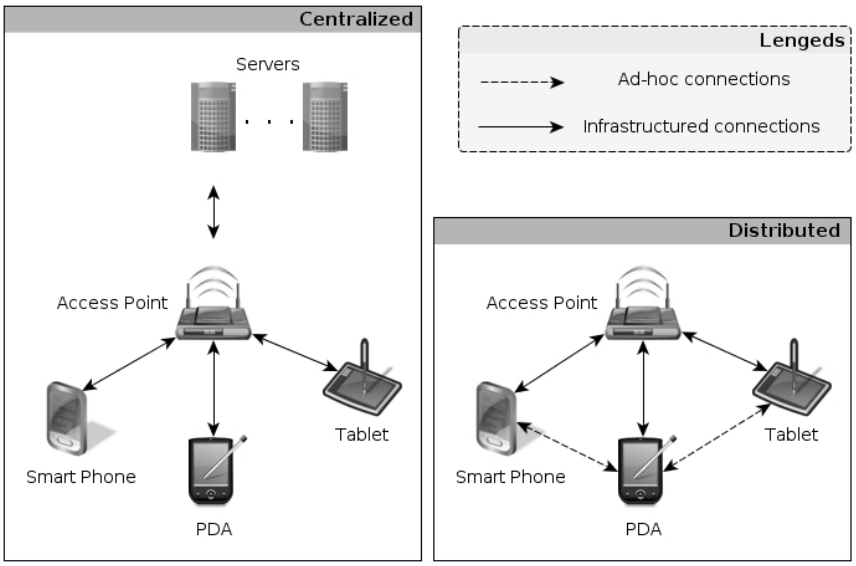

Figura 2.2: Arquiteturas de Redes Sociais Móveis [49]

Na arquitetura centralizada, os dados são centralizados em um ou mais servidores, responsáveis pelo gerenciamento e distribuição destes para os usuários

móveis. Estes dados correspondem a informações do perfil de usuários, grupos, contexto, murais, notificações, etc. Os dispositivos móveis são representados por aplicações móveis hospedadas em dispositivos como *smart phones*, *PDA's* ou *tablets*. Nesta arquitetura, toda comunicação é mediada pelos servidores. Assim, o conteúdo é necessariamente acessado por meio de servidores, e somente por eles, através de uma ligação estabelecida com aplicações móveis. As ligações são estabelecidas através de uma infraestrutura de acesso que utiliza tecnologias de comunicação sem fio. Exemplos de populares RSMs com arquiteturas centralizadas são Facebook[1], Twitter[2] e Instagram[3]. CenceMe [36] e WhozThat [8] também adotam uma abordagem centralizada, diferenciados pela maneira como os usuários acessam a rede social, exclusivamente por dispositivos móveis.

Em uma arquitetura distribuída, usuários móveis se comunicam diretamente entre si, sem utilizar servidores para mediar a interação entre eles. Como visto na Figura 2.2, conexões entre dispositivos dos usuários podem ser estabelecidos através de uma rede infraestruturada utilizando pontos de acesso ou através de conexões *ad-hoc*. Nesta arquitetura, usuários podem se comunicar e trocar conteúdo sem acesso a Internet, com a mínima infraestrutura de rede. O conhecimento inferido pelas relações sociais entre usuários pode ser utilizado para a construção de um melhor roteamento e protocolos de segurança (considerando a frequência de encontros físicos do usuário, por exemplo) [4] [38] [32].

Uma última organização dos componentes de RSMs pode ser visto em uma arquitetura híbrida. Esta combina as abordagens centralizada e distribuída permitindo que usuários móveis acessem e compartilhem dados por meio de servidores (arquitetura centralizada) e estabelecendo conexões diretas uns com os outros (arquitetura distribuída).

Todas as arquiteturas podem disponibilizar recursos que permitem um estabelecimento dinâmico de associações entre usuários, formando o que é chamado de comunidades dinâmicas (ou grupos dinâmicos) [10] [55] [34]. Comunidades dinâmicas correspondem a grupos de usuários que são construídos automaticamente baseados em: (i) interesses comuns derivados de perfis de usuários; (ii) inferência

[1] www.facebook.com

[2] www.twitter.com

[3] www.instagram.com

de interesses comuns baseada no histórico de atividades do usuário; (iii) vizinhança física de usuários ou a localização deste. Comunidades dinâmicas são formadas espontaneamente por usuários fisicamente próximos e, também, pela realização de alguma atividade conjunta com base em interesses comuns (por exemplo, estudantes de uma mesma disciplina trocando informações a respeito de um teste a ser aplicado). Neste caso, a rede é mais dinâmica, uma vez que usuários podem entrar e sair das comunidades devido a sua mobilidade ou uma mudança de interesses.

2.3 *Middleware* para Redes Sociais Móveis

O desenvolvimento de aplicações para RSMs pode ser consideravelmente simplificado utilizando um *middleware* para seu suporte. Estes sistemas proveem abstrações que reduzem o esforço de desenvolvimento, promovendo mecanismos que escondem complexidades oriundas de componentes distribuídos, oferecendo paradigmas de programação que facilitam o desenvolvimento da aplicação e promovendo interoperabilidade entre as aplicações/sistemas [3]. O *middleware* é responsável por coletar, organizar, processar e disseminar informações sociais, provendo um conjunto comum de serviços que podem ser utilizados para a construção de aplicações para RSMs.

No entanto, projetar e desenvolver *middleware* para aplicações sociais móveis não é uma tarefa trivial e sérios desafios devem ser enfrentados [39] como prover suporte para alta escalabilidade, heterogeneidade, mecanismos para adaptação em ambientes cuja mudança é constante e dispor um adequado nível de segurança e privacidade.

Nos últimos anos, várias soluções de *middleware* foram desenvolvidas [28], no entanto cada *middleware* possui características específicas e individuas. O uso de diferentes *middleware* podem gerar RSMs com características bastantes peculiares. A seguir, são descritos quatro *middleware* selecionados tendo como foco os mecanismos de segurança disponibilizados por eles. Contudo, na literatura é possível encontrar outros *middleware* para RSMs como o MobiSoft [30] e o SAMOA [12], os quais não são detalhados por não apresentarem nenhum mecanismo de segurança nos trabalhos que os descrevem.

2.3.1 MobiSoc

O MobiSoc [24] [11] provê uma plataforma que possibilita o desenvolvimento de aplicações sociais móveis, além de capturar, gerenciar e compartilhar informações sobre o estado social das comunidades em que o usuário se encontra fisicamente. Este estado é composto de informações sobre os perfis dos usuários e de lugares, além de conter informações relacionadas a afinidade entre pessoas e afinidades que pessoas possuem com lugares.

O estado dessa rede social evolui continuamente ao longo do tempo com a criação de novos perfis de usuário, laços sociais, informações relacionadas lugares, ou eventos. O MobiSoc inclui também algoritmos de aprendizagem para descobrir padrões geo-sociais previamente desconhecidos, tais como afinidades entre pessoas e entre pessoas e lugares.

A arquitetura do MobiSoc é apresentada na Figura 2.3. O MobiSoc atua como uma entidade centralizada para o gerenciamento da rede social e fornece uma API de serviço para desenvolvimento de aplicações. A arquitetura é dividida em módulos e sub-módulos. Os módulos internos podem ser fisicamente distribuídos, a fim de prover a escalabilidade para vários clientes móveis.

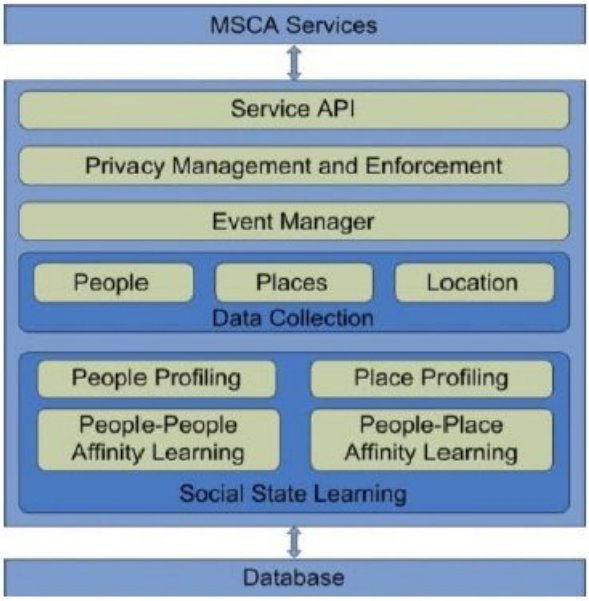

Figura 2.3: Arquitetura do MobiSoc [11]

Os módulos que compõe o MobiSoc são: Coleção de Dados (*Data Collection*), Aprendizagem de Contexto Social (*Social State Learning*), Gerenciador de Eventos

(*Event Manager*) e Gerenciador de Privacidade e Execução (*Privacy Management and Enforcement*). O Coleção de Dados é composto por três sub-módulos: Pessoas (*People*), Lugares (*Places*) e Localização (*Location*). O submódulo Pessoas permite às aplicações coletarem, armazenarem e modificarem as informações dos perfis dos usuários (como interesses, preferências, entre outras). O sub-módulo Lugares armazena informações de dados geográficos sobre construções e mapas. Além disso, provê mecanismos para adicionar informações sobre eventos relacionados com o lugar. Por fim, o submódulo Localização recebe e armazena as atualizações das informações de localização provenientes dos dispositivos móveis dos usuários.

A Aprendizagem de Contexto Social é responsável por inferir novas informações sobre o relacionamento entre pessoas e entre pessoas e lugares. Este módulo é composto pelos seguintes sub-módulos:

- Perfil do Usuário (*People Profiling*): provê informações sobre o perfil do usuário, grafo das conexões sociais deste e os grupos sociais que faz parte;
- Perfil do Lugar (*Place Profiling*): compartilha informações armazenadas sobre o lugar e aprimora a semântica do lugar com informação social;
- Aprendizagem de Afinidade Pessoa-Pessoa (*People-People Afinity Learning*): responsável por computar afinidades sociais entre pares de usuários, com base nos interesses pessoais, amigos em comum, ou ainda, lugares em comum que costumam frequentar;
- Aprendizagem de Afinidade Pessoa-Lugar(*People-Place Afinity Learning*): responsável por descobrir quais são os lugares de interesse dos usuários, através da análise histórica dos dados de localização dos mesmos.

O Gerenciador de Eventos é o módulo utilizado para comunicação assíncrona com aplicações. Estas podem registrar eventos no *middleware* para receber notificações quando algo de interesse do usuário ocorrer na rede social. Por exemplo, um usuário pode desejar uma notificação quando um outro usuário específico acessar a rede. O dispositivo móvel realiza periodicamente a busca de notificações de eventos.

O Gerenciador de Privacidade e Execução é o modulo que gerencia as regras de privacidade e acesso das entidades no sistema (usuários e aplicações). Tais regras

são armazenadas em uma base de dados. Cada aplicação registra suas preferências de acessos e privacidades. Estas regras são expressas em forma de sentenças que contém uma entidade primaria e uma secundária, esta por sua vez, pode ser um usuário ou um grupo.

O MobiSoc possui sua arquitetura orientada a serviço (SOA). Seus serviços são acessados utilizando kSOAP [35], um *framework* SOAP[4] para J2ME. O kSOAP vem de kXML, um XML utilizado em pequenos dispositivos com recursos limitados, em especial de memória e processamento.

O SOAP [1] foi escolhido devido a sua portabilidade, já que oferece uma independência de linguagem e, somado a isto, existem clientes SOAP em várias linguagens populares. Além disso, sua comunicação entre clientes e servidores se dá a partir do protocolo HTTP, pertencente a família de protocolos TCP/IP.

2.3.2 Mobilis

O Mobilis [33] [43] é um *middleware* para suporte ao desenvolvimento de aplicações de RSMs que disponibiliza serviços para o compartilhamento de informações de contexto dos usuários, gerenciamento de grupos, serviços de armazenamento de arquivos e meta-dados a eles associados, bem como a edição colaborativa de textos e imagens. Possui a arquitetura ilustrada na Figura 2.4.

O *middleware* tem sua comunicação baseada no protocolo XMPP [45] [41]. Todos os serviços mobilis têm os seus próprios identificadores XMPP que são utilizados para troca de mensagens utilizando-se este protocolo. A biblioteca cliente *XMPP Smack* é usada para a comunicação. Para a troca de mensagens XMPP, o Mobilis utiliza o servidor XMPP *Openfire*.

O componente *Mobilis Beans* compreende uma coleção de estruturas que ajudam a criar os diversos pacotes XMPP que são trocados entre o servidor Mobilis e os clientes. Estes podem ser utilizados por desenvolvedores para realizar solicitações para serviços no lado do servidor. O *MobilisServer* ou servidor Mobilis oferece diversos serviços para as aplicações cliente. O Coordenador de Serviço (*Coordinator Service*) é

[4] *Simple Object Acess Protocol* - SOAP é um protocolo para troca de informações estruturadas em XML e é utilizado no desenvolvimento de aplicações para serviços *web*.

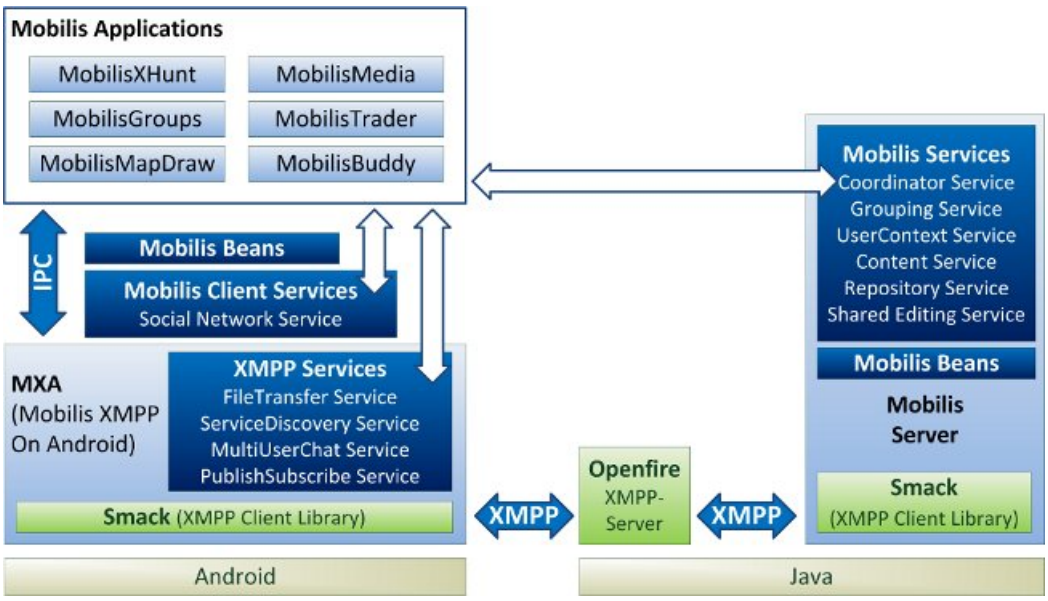

Figura 2.4: Arquitetura do Mobilis [33]

responsável por gerenciar todos os serviços oferecidos pelo servidor Mobilis, o que inclui a criação de serviços. A consulta por serviços é feita para o coordenador de serviço que possui um mecanismo de descoberta de serviços (*Service Discovery*) que retorna um identificador único do serviço solicitado, se este existir.

O Serviço de Contexto do Usuário (*User Context Service*) gerencia todos os contextos de usuários dentro da plataforma Mobilis, onde informações de contexto físico como luminosidade e temperatura, contexto técnico como largura de banda e taxa de erro, contexto pessoal como endereço e nome podem ser adquiridos a partir deste serviço. O Serviço de Grupo (*Grouping Service*) é um serviço que tem como base a criação de grupos baseados em localização, ou seja, é possível a criação e a gestão de grupos com restrições específicas (temporal e local, por exemplo) a partir de um local e até a união de grupos.

O Serviço Conteúdo (*Content Service*) e o Serviço de Repositório (*Repository Service*) tem como principal tarefa o armazenamento de mídias e meta-dados a elas associadas, respectivamente. O Serviço de Edição Colaborativa (*Collaborative Editing Service*) permite o processamento conjunto e simultâneo de dados por várias pessoas distribuídas geograficamente em um processo de edição compartilhada ou colaborativa.

O MXA (*Mobilis XMPP on Android*) encapsula as funcionalidades da biblioteca *XMPP Smack* e promove uma interface AIDL (*Android Interface Definition*

Language) que é utilizada em aplicações Mobilis para se comunicar com o servidor que oferece os diversos serviços descritos anteriormente.

2.3.3 MyNet

O MyNet [26] é uma plataforma de *middleware* P2P seguro para interação social entre usuários de dispositivos móveis que permite aos usuários compartilharem seus dispositivos, conteúdos e contatos sociais sem a necessidade de um repositório central ou infraestrutura. O MyNet permite que as informações sociais dos usuários e serviços distribuídos sejam acessadas e compartilhados entre amigos em tempo real, diretamente de seus próprios dispositivos pessoais.

O *middleware* foi construído no topo de uma tecnologia P2P chamada Unmanaged Internet Architecture (UIA) [18], a qual provê duas funcionalidades básicas: conectividade ubíqua e gerenciamento de grupos distribuídos. O UIA permite que os usuários compartilhem seus perfis e dados pessoais de forma segura pela rede. Nesta rede, cada dispositivo possui um identificados único e permanente, o Endpoint Identifier (EID), o qual é usado para identificar os dispositivos em todas as serviços disponíveis na rede. A Figura 2.5 mostra como foi desenvolvida a arquitetura do MyNet.

Uma rede social no MyNet é formada por um conjunto de dispositivos móveis devidamente identificados e autenticados, formando um *cluster*, chamado pelo autor de *Personal Divice Cluster* (PDC). A inclusão de novos usuários é realizada pelo módulo *Out-of-Band Introductions*. Esse módulo é responsável por realizar todo o protocolo de descobertas de novos dispositivos na rede, bem como os precedimentos necessários para inclusão destes ao PDC.

Os componentes do *middleware* e suas aplicações podem usar chamadas de procedimentos remotos assíncronas para comunicação, o que é realizado através da camada *MyARPC*. Opcionalmente, os componentes ou aplicações podem utilizar a *Application Programing Interface* (API) disponibilizada pelo *MyNet Messaging*, que corresponde a uma API de nível mais alto o *MyARPC*.

Todas as informações locais, como contatos, informações de contexto, entre outras, bem como o estado atual da rede são persistidos no *PDC-store*. Essas

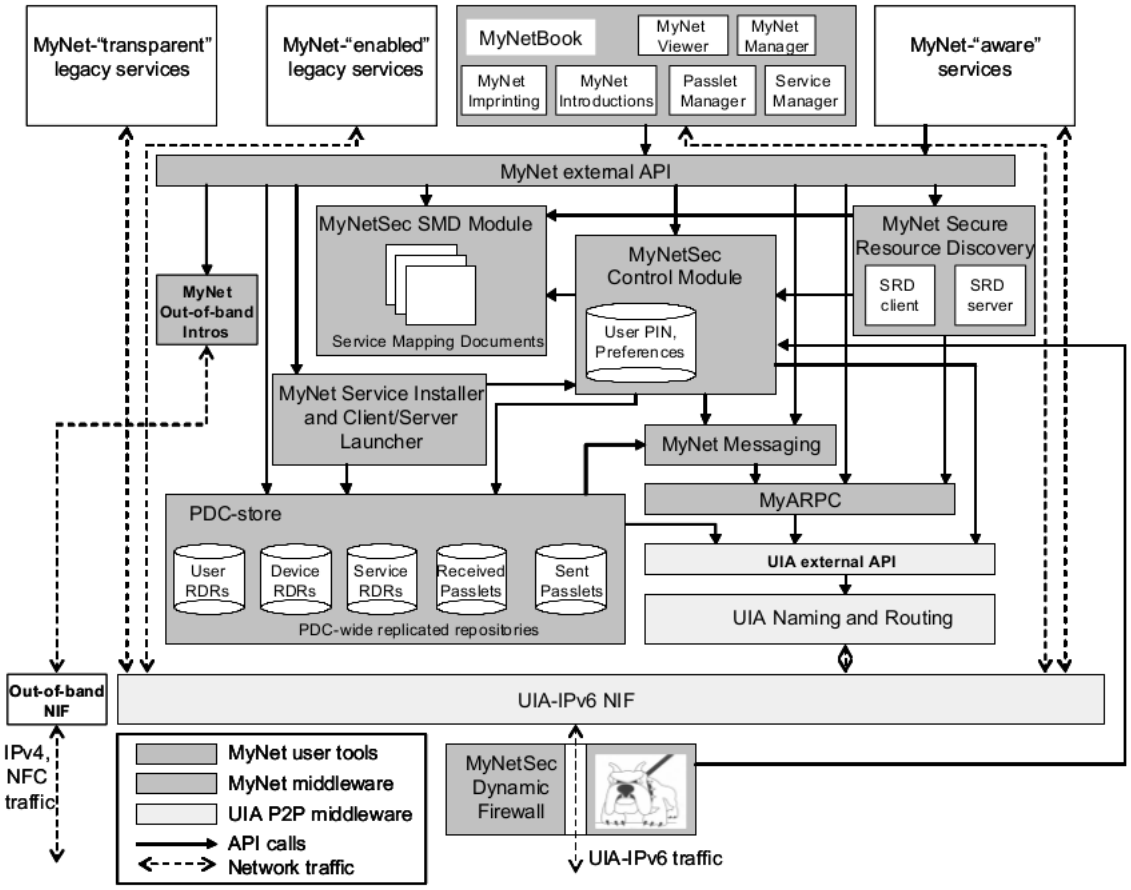

Figura 2.5: Arquitetura do MyNet [26]

informações são compartilhadas entre os diversos dispositivos, para tanto utiliza-se a API do *MyNet Messaging*.

O *MyNetSec Control Module* é responsável por gerenciar o controle de acesso às informações. Este módulo, funciona como um *firewall*, interceptando todas as chamadas realizadas ao serviços disponibilizados pelos dispositivos. Então, o *MyNetSec Control Module* decide se a operação será permitido ou não, baseada nas preferencias de compartilhamento dos usuários. Por outro lado, o MyNet também dispõe de um serviço de descoberta de recursos, o *Secure Resource Discovery*, que é responsável pela descoberta de novos recursos compartilhados na rede, respeitando as preferências de privacidade dos seus donos. Estes recursos podem ser serviços ou conteúdo disponibilizados pelos usuários na rede.

O MyNet disponibiliza também um suporte a operações desconectadas, onde caso o dispositivo esteja desconectado as mensagens são persistidas localmente em um estrutura de fila pelo *middleware* e transmitidas posteriormente.

2.3.4 MobiClique

O MobiClique [37] é um *middleware* de RSM que utiliza informações e preferências do usuário obtidas a partir de uma rede social (Facebook, por exemplo) e as utiliza para localizar seus contatos mais próximos. Dessa forma, usuários podem trocar mensagens e compartilhar conteúdo entre si utilizando uma rede *ad-hoc*. Como utiliza uma rede social já existente para obtenção de dados do usuário, como seu perfil, o MobiClique é considerado um *middleware* oportunista.

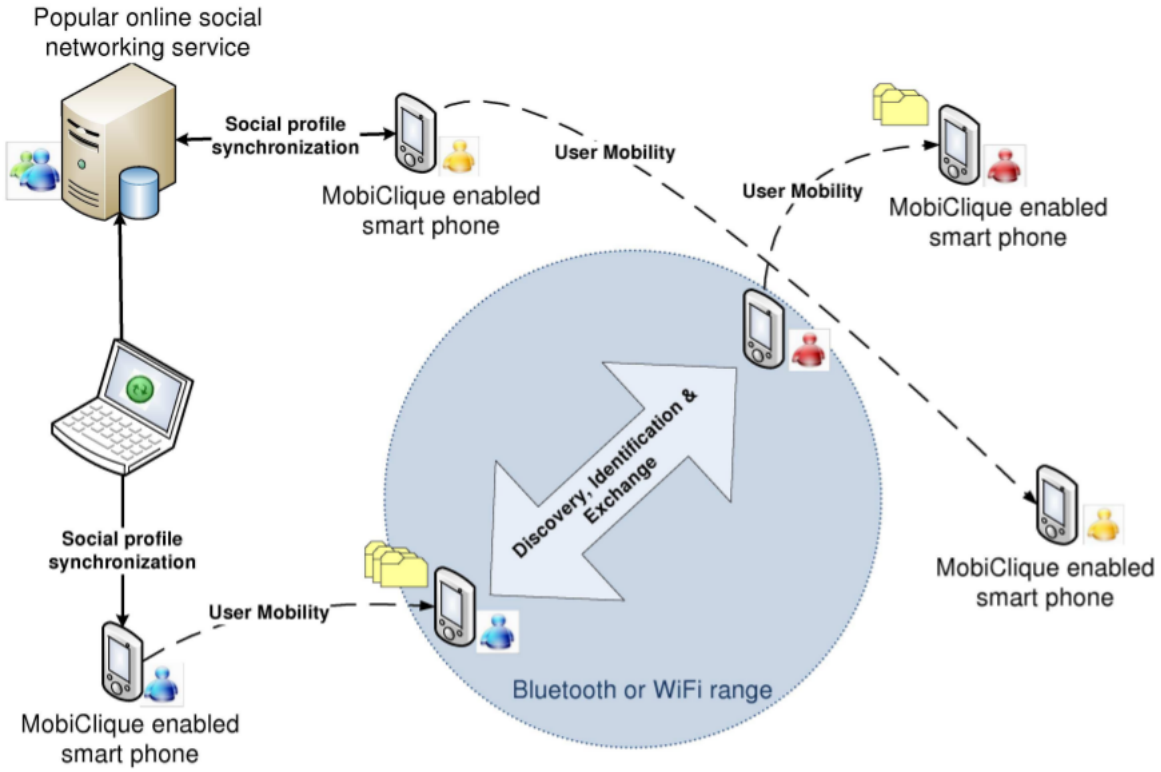

Figura 2.6: Arquitetura do MobiClique [37]

No MobiClique cada dispositivo obtém o perfil do usuário a partir de uma conexão com a Internet, podendo obtê-la através de uma *interface* sem fio do próprio dispositivo ou através de outros dispositivos que possibilitam essa conexão, conforme mostra a Figura 2.6. Essas informações são utilizadas na criação da rede social *ad-hoc*. Na rede *ad-hoc*, representada pelo circulo na figura, cada dispositivo realiza três operações básicas: (1) descoberta de vizinhança, de acordo com suas preferencias; (2) identificação do usuário (autenticação) e (3) compartilhamento de informações.

No perfil do usuário estão incluídas informações pessoais, preferências, lista de amigos, entre outros dados. Usuários são identificados pelo mesmo identificador

utilizado pela rede social *online* e este, por sua vez, é único. Cada nó executa um algoritmo de descoberta de vizinhança, dessa forma usuários saberão quais outros usuários estão ao seu redor.

Os usuários interagem entre si através de mensagens, que podem ser direcionadas a um usuário específico ou a um grupo de usuários (que pode ser obtido a partir da rede social, ou pode ser criado). As mensagens passam de nó em nó até chegar ao seu destino. Se a *interface* de rede estiver ocupada com outra aplicação ou não existir vizinho disponível, as mensagens são armazenadas localmente e são enviadas para os próximos nós quando estiverem disponíveis. Quando um dispositivo recebe uma mensagem, o usuário é automaticamente notificado. A API *Mobile Social Networking* provê métodos para gerenciar o perfil social do nó com o intuito de trocar informações pessoais e incluir ou remover amigos e interesses.

O MobiClique adota a arquitetura Haggle [47]. Essa arquitetura provê toda a comunicação do *middleware*, além de possuir mecanismos essenciais de comunicação implementados, como descoberta de vizinhos, armazenamento persistido de dados e sua disseminação. Para a comunicação entre os nós é utilizado o protocolo IEEE 802.15, o *Bluetooth*.

2.4 Conclusão

O estabelecimento de RSMs envolve aspectos relacionados a Redes Sociais, Computação Móvel e Ciência de Contexto. Esta combinação possibilita a interação de seus usuários em qualquer lugar e a qualquer hora. Contudo, existem diferentes arquiteturas de redes sociais móveis: centralizada, distribuídas e híbrida.

Na arquitetura centralizada, os dados são centralizados em um ou mais servidores, os quais são responsáveis pelo gerenciamento e distribuição destes para os usuários móveis através de uma ambiente de rede sem fio. Por outro lado, na arquitetura distribuída os dispositivos móveis se conectam diretamente uns com os outros, onde os dados ficam armazenados nos próprios dispositivos. Por fim, existe a arquitetura híbrida, que junta as duas arquiteturas (centralizada e distribuída) em um único ambiente. A escolha do tipo de arquitetura a ser utilizada causa um grande

impacto neste modo de interação e, em especial, em como a comunicação ocorrerá entre os participantes da RSM.

Diversos *middleware* para RSMs foram idealizados, cada um com suas particularidades e arquitetura. O objetivo destes sistemas é simplificar o desenvolvimento de aplicações para RSMs, promovendo uma diversidade de serviços, disponibilizados com o suporte de um *middleware*. Nesse contexto, foram descritos o MobiSoc e o Mobilis, que apresentam arquiteturas centralizadas, e o MyNet e MobiClique, como exemplos de *middleware* com arquiteturas distribuída.

3 Modelo de Segurança e Privacidade para o MobileHealthNet

Neste capítulo são apresentados maiores detalhes sobre o projeto MobileHealthNet, seus objetivos, requisitos gerais e de segurança, e sua arquitetura. Além disso, são descritos os componentes que compõem o modelo de segurança proposto neste trabalho, bem como os aspectos de implementação de cada um deles.

3.1 O Projeto MobileHealthNet

Uma área de aplicação das RSMs é a saúde. RSMs voltadas para esta área podem envolver uma combinação dos diversos agentes envolvidos no processo de atenção à saúde, incluindo profissionais da saúde (médicos, enfermeiros, fisioterapeutas, terapeutas ocupacionais, etc), pesquisadores da saúde (professores e alunos de graduação e pós-graduação), pacientes e seus familiares, bem como membros da comunidade em geral.

Quando devidamente estabelecidas, RSMs podem contribuir com o processo de atendimento à saúde, tornando-o mais eficaz e eficiente. Entre os benefícios esperados, destacam-se um melhor fluxo de informação e maior colaboração entre profissionais dos diversos níveis de atendimento à saúde; melhorias nos níveis de comprometimento e informação do paciente, o que contribui com o processo terapêutico [31]; melhor gestão de pacientes portadores de doenças crônicas, levando a diminuição da ocorrência de complicações; melhoria na qualidade da tomada de decisão relativa ao tratamento por parte dos pacientes; melhor suporte emocional aos pacientes e redução de custos do sistema de atendimento à saúde, diminuindo-se a necessidade de deslocamentos e a ocorrência de complicações no tratamento de pacientes.

O projeto MobileHealthNet [50] [51] objetiva a construção de um *middleware* para RSMs, destinado à criação de novos serviços e aplicações voltadas para a área

da saúde. Este projeto está sendo desenvolvido em parceria entre o Laboratório de Sistemas Distribuídos (LSD) da Universidade Federal do Maranhão e o *Laboratory for Advanced Collaboration* (LAC) da Pontifícia Universidade Católica do Rio de Janeiro. Este projeto conta com apoio institucional do Hospital Universitário da UFMA, através de duas unidades acadêmicas: o PAPA e a Casa da Dor.

As aplicações em desenvolvimento no contexto deste projeto têm como objetivos específicos:

- Encurtar a distância entre profissionais da saúde e pacientes, fornecendo meios para que o contato entre eles se estenda além dos encontros presenciais, disponibilizando também serviços de colaboração em tempo real;
- Facilitar a interação entre profissionais de diversas especialidades que frequentemente encontram-se dispersos geograficamente e que necessitam trabalhar colaborativamente para o acompanhamento de pacientes;
- Promover a educação em saúde tanto para pacientes como profissionais da saúde através da disponibilização de cursos, palestras e tutoriais, permitindo a construção colaborativa deste conteúdo e sua discussão em um ambiente de rede social;
- Promover um meio de comunicação entre profissionais dos diversos níveis da atenção a saúde de forma a facilitar o intercâmbio e colaboração entre os mesmos.

Após diversas sessões iniciais de interação entre profissionais da computação e da área da saúde foram definidos os principais requisitos a serem observados na construção do *middleware* e aplicações, entre os quais destacamos os seguintes:

- Componentes de software para dispositivos móveis devem apresentar baixo consumo de recursos, de maneira que possam ser executados em uma grande variedade de equipamentos, incluindo aqueles considerados de baixo custo, já que os resultados do projeto devem ser aplicáveis a comunidades carentes;
- Deve-se prover suporte a diversos tipos de comunicação sem fio (em especial sistemas celulares e redes locais sem fio), de forma a se atingir uma ampla área de cobertura, minimizando-se custos de comunicação sempre que possível;

- Deve-se prover meios para a construção e compartilhamento de conteúdo de forma colaborativa por seus usuários;

- Os usuários devem ter a liberdade de criar e participar de diversos grupos de acordo com seu interesse, fomentando-se assim a criação de comunidades dinâmicas que, no entanto, devem respeitar critérios de privacidade e segurança a serem estabelecidos;

- Disponibilizar meios para a interação de forma síncrona (*on-line*) e assíncrona (*off-line*) entre participantes das comunidades;

- Fornecer mecanismos para notificações (alertas) de eventos aos quais solicita-se atenção prioritária;

- Deve-se prover mecanismos que permitam a comunicação em tempo real e com suporte a QoS (*Quality of Service* - Qualidade de Serviço), de forma a habilitar a notificação de eventos prioritários e a transmissão de dados de monitoração remota de pacientes (a serem explorados em aplicações futuras);

- Todo o processo de desenvolvimento deve estar em conformidade com os requisitos especificados no *Manual de Certificação para Sistemas de Registro Eletrônico em Saúde* (MC-SRES) [14], publicado pela *Sociedade Brasileira de Informática em Saúde* (SBIS) e *Conselho Federal de Medicina* (CFM).

Após a definição dos objetivos e da elicitação dos requisitos, iniciou-se a concepção do *middleware*, o que inclui a elaboração da arquitetura (descrita na seção seguinte) e seu processo de desenvolvimento, implementação da infraestrutura de comunicação, além dos aspectos de segurança necessários.

3.2 O *middleware* MobileHealthNet

A arquitetura do MobileHealthNet é organizada nas camadas, ilustradas na Figura 3.1. A camada de aplicações (Applications) refere-se às aplicações previstas no projeto para serem desenvolvidas nesta etapa inicial do MobileHealthNet. Quatro aplicações estão previstas para serem desenvolvidas nesta etapa inicial: (i) Health Education possui ferramentas para o compartilhamento de arquivos multimídia e seu

objetivo é aprimorar a educação de pacientes e profissionais da saúde através de mídias com conteúdo educacional; (ii) Professional Collaboration visa diminuir a distância entre os profissionais da saúde, através de recursos como chat multiusuário, fórum de discussão e serviço de notificação que permita informar sobre a urgência de se obter o resultado de um exame, discutir a respeito do tratamento e acompanhamento de um determinado paciente; (iii) HUPD Care tem por objetivo explorar os conceitos das RSMs para promover a assistência prestada por especialistas responsáveis pelo atendimento de alta complexidade a profissionais da atenção básica a saúde no atendimento a casos específicos; (iv) Patient-Buddy-Build possibilita a criação de questionários simples e práticos a serem respondidos por pacientes, para informarem periodicamente o estado de sua doença. A camada de Serviços de Aplicação (*Application Services*) disponibiliza serviços típicos de redes sociais, como serviços de publicação de mensagem em murais, fórum e chat, além de um serviço de notificações (alertas) para os usuários.

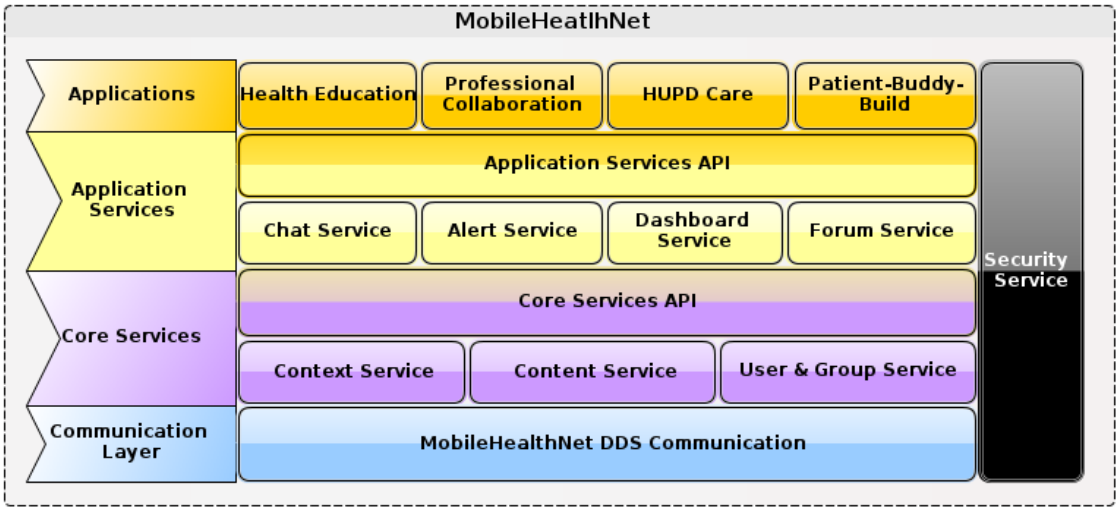

Figura 3.1: Arquitetura Geral do MobileHealthNet

A camada de serviços do núcleo (*Core Services*) disponibiliza serviços básicos que são compartilhados pelos demais serviços e aplicações do MobileHealthNet. O Serviço de Contexto é o responsável pelo armazenamento e disponibilização de informações de contexto. Por exemplo, as informações de localização de um determinado usuário podem ser compartilhadas com os outros usuários da rede social. O Serviço de Conteúdo tem como principal tarefa o compartilhamento de mídia (texto, fotos, áudio, filmes, etc) entre usuários da rede social. Este serviço permite rotular cada elemento de mídia com metainformações

definidas pela aplicação como, por exemplo, a posição geográfica na qual a mídia foi obtida ou o modelo de uma bomba de asma ao qual um tutorial se refere. O Serviço de Usuários e Grupos é responsável pelo cadastro e gerenciamento de usuários e grupos na rede, bem como os relacionamentos entre os usuários da rede, o que é característico das redes sociais. Como exemplo, podemos citar o relacionamento de "amizade" entres os usuários, os quais podem enviar solicitações de amizades uns aos outros, podendo ser aceito ou não pelo usuário solicitado.

A camada *Security Services* é transversal a todo o código gerado no MobileHealthNet e disponibiliza os componentes responsáveis pelos mecanismos de privacidade e segurança implementados no *middleware*. Este trabalho está inserido no contexto do desenvolvimento desta camada.

Finalmente, a camada de comunicação proposta em [6], que por sua vez está sendo desenvolvida tendo como protocolo base o *Data Distribution Service* (DDS). O DDS é uma especificação da OMG [22] (*Object Management Group*) e é um padrão para comunicação *publish/subscribe* com alta qualidade de serviço que visa a distribuição crítica de informações em sistemas distribuídos em tempo real. Maiores detalhes sobre o DDS e sobre a camada de comunicação do MobileHealthNet são apresentados no Capítulo 4, no qual é apresentado uma importante contribuição deste trabalho, um mecanismo de comunicação segura baseado na especificação OMG DDS.

3.3 Metodologia Usada na Construção do Modelo de Segurança

Nas RSMs, os usuários estão constantemente compartilhando informações pessoais e muitas destas exigem um rigoroso nível de privacidade. Para Joshi e Kuo [25], interações sociais tem gerado uma grande quantidade de dados nas redes sociais contendo muitos detalhes privados e sensíveis sobre seus donos. Além disso, os dispositivos móveis coletam informações de contexto dos usuários, que podem comprometer sua privacidade se acessadas por pessoas não autorizadas. Por exemplo, em muitos casos disponibilizar a real localização de um usuário representa uma quebra da sua privacidade [5]. Beach et al., em [9], afirmam que em RSMs as informações de contexto podem influenciar fortemente na quebra de privacidade dos dados, devido à

própria natureza do dado enviado para a rede social agregar informações do contexto de quem o enviou. Desta forma, é necessário que o software, neste caso o *middleware*, possua mecanismos de privacidade e segurança adequados para garantir que as informações confidenciais não sejam acessadas por indivíduos não autorizados.

No MobileHealthNet, as informações trocadas entre os usuários estão relacionadas com a saúde das pessoas, o que torna a segurança essencial para seu uso. Cada serviço do MobileHealthNet deve possuir mecanismos destinados a manutenção da privacidade dos dados, garantindo o controle de acesso às informações sensíveis. No Brasil, a *Sociedade Brasileira de Informática na Saúde* (SBIS) em conjunto com o *Conselho Federal de Medicina* (CFM) editam o *Manual de Certificação para Sistemas de Registro Eletrônico em Saúde* (MC-SRES), que corresponde a um documento de certificação que possui uma extensa lista de critérios de segurança a serem seguidos pelos sistemas computacionais voltados para saúde. Este manual é utilizado para normatizar a utilização de sistemas informatizados do atendimento em saúde e requer que as aplicações possuam um modelo de privacidade e segurança completo, que atenda as necessidades dos usuários e além disso, esteja de acordo com as exigências da legislação. O MC-SRES apresenta um conjunto de requisitos de segurança que abrange desde a comunicação entres os componentes de software do sistema até a segurança dos dados. Para que um Sistema de Registro Eletrônico em Saúde obtenha o certificado da SBIS ele precisa, obrigatoriamente, atender a todos estes requisitos.

Tais fatores tornam evidente a necessidade de se ter um modelo de segurança para RSMs, especialmente se esta for voltada para a área da saúde, como é o caso do MobileHealthNet. Este trabalho está focado no desenvolvimento de uma infraestrutura de software, neste caso o *middleware*, destinado ao desenvolvimento de RSMs voltadas para saúde. Nesse contexto, é imprescindível que essas RSMs possuam um nível de segurança necessário para manter a privacidade das informações compartilhadas na rede. Para tanto, foi realizado, inicialmente, um levantamento do estágio atual das pesquisas científicas e do conhecimento já adquirido por pesquisadores na área de RSMs, e mais especificamente, segurança em RSMs. Neste momento foram utilizados o acervo bibliográfico da UFMA, artigos científicos disponíveis nos portais da CAPES, ACM e IEEE. Foram utilizados também anais de congressos e seminários dentro do grupo de pesquisa, além de livros de

fundamentação teórica das áreas de Computação Móvel, Segurança da Informação, Privacidade Online, Segurança em Redes Móveis e outros.

Em uma segunda etapa foi realizada a definição do escopo do problema, delimitando que contribuições esta pesquisa apresentaria para a área de segurança em RSMs. Nesta etapa foi também realizado o levantamento de requisitos para o desenvolvimento de um modelo de segurança e privacidade para RSMs destinadas à saúde, fundamentada em pesquisas realizadas na etapa anterior. Então, o passo seguinte destinou-se à elaboração de um modelo contemplando os principais problemas a serem tratados. O objetivo foi de desenvolver um conjunto de mecanismos de segurança a serem aplicados no contexto das RSMs, iniciando pelo processo de modelagem e análise de sua viabilidade.

A seguir, usando o modelo elaborado, iniciou-se a fase de implementação, começando pela camada da comunicação. Para direcionar o desenvolvimento dos mecanismos de segurança do *middleware* e das primeiras aplicações, foram analisados alguns dos principais processos de desenvolvimento de software seguro, escolhendo-se o *Comprehensive, Lightweight Application Security Process* (CLASP). O CLASP [44] é um processo de desenvolvimento de software seguro proposto pela OWASP (*The Open Web Application Security Project*), que contempla um conjunto de atividades, 24 no total, que podem ser integradas em qualquer processo de desenvolvimento de software. Todas as atividades do CLASP são baseadas em papéis, onde cada papel tem uma função específica dentro do processo. O CLASP é flexível e possui quais as atividades devem ser realizadas, no entanto ele não se preocupa em informar como essas atividades são realizadas. Além disso, a equipe de desenvolvimento não é obrigada a realizar todas as 24 atividades, ficando a escolha da equipe quantas e quais atividades serão realizadas. Porém, todas as atividades possuem individualmente um fator de impacto a ser levado em consideração caso sejam removidas do processo.

No estágio atual de desenvolvimento do *middleware*, parte dos mecanismos de segurança propostos no modelo descrito neste trabalho ainda estão em fase de implementação. Uma outra contribuição deste trabalho de mestrado é o desenvolvimento e avaliação dos mecanismos de segurança referentes à camada de comunicação, descritos nos Capítulos 4 e 5. Após a conclusão da implementação de cada componente do modelo proposto devem ser realizadas avaliações quantitativas e qualitativas. Além disso, deve-se validar também a integração dos mecanismos de

segurança com as funcionalidades do *middleware*, garantindo que estes mecanismos não apresentem prejuízos ao seu desempenho e à sua usabilidade.

3.4 Requisitos para Construção do Modelo de Segurança

Nas redes sociais as informações são compartilhadas diariamente por milhares de usuários, fazendo com que uma grande quantidade de informações estejam disponíveis em um só local. Nestes ambientes as pessoas são encorajadas a publicarem seus nomes, data de nascimento, escola, cidade de nascimento, interesses pessoais, entre outras informações pessoais. O acesso a essas informações por pessoas não autorizadas pode representar um grande risco para a privacidade de seu dono, uma vez que elas podem ser utilizadas para diversos fins maliciosos. Hongyu Gao et al. em [20], apresentam alguns aspectos relevantes relacionados aos ataques contra a privacidade e segurança em redes sociais. Dada a importância das informações compartilhadas nestes ambientes quando são destinados à área da saúde, podemos afirmar que o não cumprimento de alguns requisitos de segurança podem tornar o uso destes ambientes um risco para o usuário. Assim, após um levantamento de vulnerabilidades das RSMs e buscando atender o MC-SRES, foram listados os requisitos de segurança que o *middleware* MobileHealthNet deve atender. Esses requisitos estão relacionados às diversas camadas do sistema, desde a aplicação, que correspondem à confidencialidade das informações, até a camada de comunicação, que envolve o transporte dos dados no ambiente distribuído. O desenvolvimento dos mecanismos de segurança para o MobileHealthNet deve levar em consideração os seguintes requisitos:

1. **Requisitos de Autenticação**: Todo usuário deve ser identificado e autenticado antes de qualquer acesso ao sistema. O MC-SRES lista um conjunto de métodos de autenticação, podendo ser escolhido no mínimo um deles. Todos os dados relacionados às credenciais, como senha, devem ser armazenados de forma protegida. Além disso, deve-se disponibilizar troca de senha periodicamente, além do bloqueio de usuário caso o mesmo realize um número mínimo de tentativas de acesso;

2. **Requisitos de Autorização**: Todo acesso aos dados deve ser validado pelo mecanismo de controle de acesso, para validar as permissões de acesso do usuário. As permissões de acesso devem ser configuráveis, e serem realizadas apenas por usuários com os devidos direitos;

3. **Privacidade dos Usuários**: A rede social deve disponibilizar meios precisos de garantir o acesso às informações pessoais somente por usuários explicitamente autorizados pelo dono. Diante disso, o próprio usuário deve definir quais informações estarão disponíveis para outros acessarem e quais outros usuários podem acessá-las. Por exemplo, o usuário deve informar para quais usuários ou grupos suas publicações estarão visíveis;

4. **Privacidade das Informações de Contexto**: O sistema deve garantir a privacidade das informações de contexto publicadas pelos usuários, além das informações de contexto agregadas a outras publicações, como informações de localização, tempo, etc;

5. **Não Repúdio**: O sistema deve disponibilizar meios de identificar o autor das operações realizadas no ambiente, de modo que o usuário não possa negar que realizou tal operação. Por exemplo, se um usuário publicar uma determinada informação na rede social, o sistema deve garantir com precisão qual usuário publicou tal informação;

6. **Auditoria**: O sistema deve possuir um mecanismo que permita listar todas as interações realizadas pelos usuários na rede social, informando os usuários que as realizou, o momento exato, além de outras informações necessárias. Além disso, todas as informações de tempo devem estar no mesmo formato e extraídos da mesma fonte temporal;

7. **Comunicação de Terceiros**: Os dados restritos do sistema devem ser disponibilizados somente aos parceiros previamente autorizados, mediante documento formal de autorização;

8. **Comunicação Segura**: O sistema deve garantir uma transferência de dados segura, mantendo a autenticidade, integridade e confidencialidade no transporte das mensagens;

9. **Segurança da Base de Dados**: O sistema deve possuir mecanismos de backup e restauração de backups, garantindo, sobretudo, sua consistência no processo. O MC-SRES proibe a exclusão e modificações de dados na base, qualquer alteração deve preservar os dados antigos.

10. **Notificação de Ocorrências**: Deve-se disponibilizar uma interface para que usuários possam notificar a ocorrência de incidentes de segurança, problemas, melhoramentos ou sugestões.

Esses requisitos nos levaram a propor um modelo de privacidade e segurança para RSMs aplicadas na área da saúde, descrito na próxima seção. Ele é composto por um conjunto de componentes que implementam os mecanismos de segurança necessários para garantir o grau de segurança desejado para estes ambientes.

3.5 O Modelo de Segurança

O modelo proposto [21], Figura 3.2, está descrito como um conjunto de mecanismos de segurança que serão agregados aos componentes do *middleware*, incluindo neles recursos de autenticação, autorização, comunicação segura e gerenciamento de logs. O modelo traz mecanismos de segurança para todas as camadas do MobileHealthNet e sua construção foi baseada no conjunto de requisitos levantados a partir de uma análise das principais vulnerabilidades encontradas em RSMs, além dos requisitos de segurança exigidos pelo MC-SRES.

A seguir são descritos cada um dos componentes que compõem o modelo proposto neste trabalho, incluindo sua justificativa e a identificação do requisito ao qual está associado. Por fim, são apresentados alguns aspectos de implementação do modelo e o estágio atual de seu desenvolvimento. Observamos que não é escopo deste trabalho de mestrado a implementação completa do modelo. Focamos na

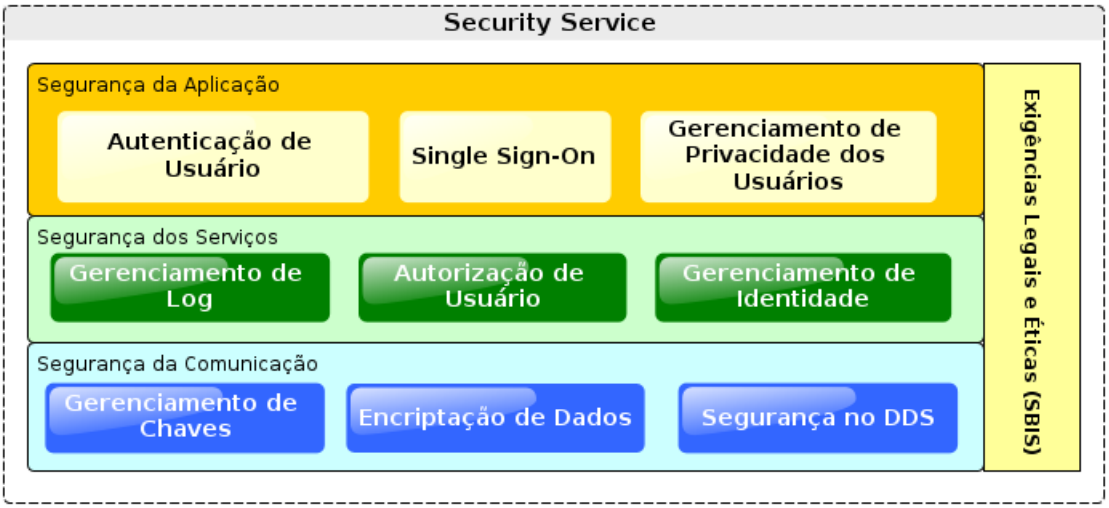

Figura 3.2: Modelo de Segurança Aplicado ao MobileHealthNet

definição do mesmo e implementação de alguns de seus serviços básicos, em especial os relacionados à infraestrutura de comunicação entre os componentes do middleware.

3.5.1 Autenticação de Usuários e Single Sign-On

A autenticação de usuários tem como objetivo validar se o usuário é realmente quem ele afirma ser, ou seja, visa validar a identidade do usuário junto ao sistema. A autenticação é um processo chave para o controle de acesso, pois para que este mecanismo funcione corretamente é necessário que o usuário seja devidamente autenticado. O processo de autenticação pode ser realizado baseado em três possibilidades [46]: (I) algo que o usuário conhece (como uma senha); (II) algo que o usuário possui (como um cartão inteligente) ou (III) alguma característica do usuário (como a impressão digital, utilizada em leitores biométricos).

O MC-SRES disponibiliza um conjunto de métodos de autenticação para que o desenvolvedor escolha no mínimo um (Requisito 1). O processo de autenticação no MobileHealthNet é realizado através da utilização de certificação digitais, o qual também é usado na criação de canais seguros de comunicação entre os dispositivos móveis distribuídos no ambiente. O *middleware* MobileHealthNet é composto por vários serviços, estes, por sua vez, podem ser distribuídos. O uso de qualquer serviço está condicionado ao usuário autenticado e autorizado a acessá-lo. Para que o usuário não seja obrigado a autenticar-se a cada serviço que ele necessite utilizar, o

MobileHealthNet dispõe de um modelo de autenticação *Single Sign-On* (SSO). O SSO é o modelo pelo qual uma única ação de autenticação permite ao usuário o acesso a todas as aplicações do sistema, para as quais ele possua permissão de acesso. O uso do SSO é muito comum em serviços Web. No entanto, seu uso pode ser estendido para qualquer sistema de arquitetura distribuída. Portanto, independente do método de autenticação, o processo só será realizado uma única vez, habilitando o usuário a ter acesso a qualquer serviço que esteja autorizado a acessar.

3.5.2 Gerenciamento de Privacidade dos Usuários

Nas redes sociais as informações são compartilhadas diariamente por milhares de usuários. Isso faz com que uma grande quantidade de informações estejam disponíveis em um único ambiente, geradas a partir das interações dos usuários. Um dos grandes desafios para a segurança desses dados está em manter a privacidade desses usuários, isto porque a maioria das informações que circulam em RSMs são consideradas pessoais. Dessa forma, o gerenciamento de privacidade da rede social deve garantir que essas informações sejam acessadas somente por usuários autorizados (Requisito 3 e 4). Este mecanismo se diferencia do mecanismo de Autorização de Usuários, descrito na próxima seção, no fato de que o Gerenciamento de Privacidade está relacionado com informações pessoais dos usuários, sendo definido pelo próprio usuário. As restrições de acesso devem ser obrigatoriamente definidas pelo dono da informação. Portanto, as aplicações de rede social devem disponibilizar mecanismos especificamente para este fim, as quais são geralmente realizadas através de interfaces gráficas específicas.

O gerenciamento de privacidade do MobileHealthNet será realizado de acordo com cada serviço e suas informações. No Serviço de Alerta e Chat, as mensagens só poderão ser acessadas pelos dois interlocutores envolvidos na interação. Por outro lado, nos serviços de Dashboard, Forum e Conteúdo, os usuários podem definir as opções de privacidade de três formas, podendo ser: (i) publico, onde informações são disponíveis para qualquer usuário da rede; (ii) privado, neste caso somente o dono da informação possui acesso, ou (iii) compartilhado, onde o dono da informação determina explicitamente com quais usuários ou grupos ele deseja compartilhar a informação. O serviço de contexto, no entanto, segue o modelo *publish* e

subscribe, onde o usuário que deseja receber informações de contexto de outro usuário, envia uma requisição de subscrição ao serviço. Esta subscrição só será efetivada se esta for autorizada pelo dono da informação. Uma vez autorizado, o subscritor receberá notificações à medida que o usuário publicar suas informações de contexto.

Uma vez definidos as configurações de compartilhamento, o *middleware* deve gerar um conjunto de *statements* (declarações), que são armazenadas e utilizadas posteriormente pelo mecanismo de gerenciamento de privacidade para verificar as opções de compartilhamento para uma determinada informação. Um *statement* é uma estrutura formal utilizada para denotar os atributos necessários ao *middleware* sobre o compartilhamento. Cada *statement* é composto de cinco atributos: (I) o usuário que está compartilhando a informação; (II) a entidade para a qual a informação está sendo compartilhada, que podem ser outro usuário ou um grupo de usuários; (III) as restrições do compartilhamento, que podem ser de tempo ou localização; (IV) as operações que a entidade destino pode realizar sobre a informação e por fim, (V) o conteúdo em si, que pode ser qualquer informação utilizadas pelos serviços, um arquivo compartilhado, uma mensagem do fórum, informações de contexto, etc.

A grande vantagem do uso de *statements* para gerenciamento de privacidade está em sua expressividade e flexibilidade. Expressividade porque em um único *statement* pode-se definir diversas possibilidades de compartilhamento para o mesmo conteúdo, variando as restrições e operações relacionadas. Além disso, são flexíveis porque permitem modificações dos atributos relacionado às permissões (restrições e operações), além de permitir também a inclusão de mais de um *statement* para o mesmo conteúdo.

3.5.3 Autorização de Usuários

O mecanismo de autorização visa verificar se um determinado usuário possui os direitos de realizar uma determinada operação sobre um dado recurso protegido do sistema. Diferente do Gerenciamento de Privacidade, as restrições de acesso serão cadastradas por um administrador do sistema. Nesse contexto, a autorização consiste, basicamente, em sujeitos que geram uma requisição ao sistema para acessar determinados objetos. Um objeto pode ser qualquer recurso do sistema: processos, arquivo, impressora, etc. Sujeitos são processos que agem em nome

de usuários, mas também podem ser objetos que precisam dos serviços de outros objetos para executar seu próprio trabalho [48]. Portanto, é necessário proteger os objetos contra invocações de sujeitos que não têm permissão de realizar determinadas operações (Requisito 2). Essa proteção é gerenciado pelo monitor de referência. O monitor de referência é a entidade que recebe todas as requisições de acesso dos sujeitos e autoriza ou nega o acesso de acordo com a política de segurança [17]. O monitor de referência registra qual sujeito pode realizar o quê e decide se um sujeito tem permissão ou não de executar uma determinada operação.

No contexto do MobileHealthNet, o mecanismo de autorização está associado com as operações que os usuários podem realizar no ambiente baseado no papel que este possui no mesmo, através do modelo *Role-Based Access Control* (RBAC). No RBAC, a identidade principal no sistema é o papel. Papéis são criados de acordo com os diferentes cargos ou funções em uma organização, e os usuários são associados a papéis de acordo com as suas responsabilidades e qualificações. Dependendo da necessidade, um usuário pode ser facilmente remanejado de uma papel para outro. Este mecanismo têm como base o fato dos direitos de acesso são atribuídos a papéis e não a usuários, já que os usuários obtêm estes direitos em virtude de terem papéis a si atribuídos. Por ser independente das políticas, o RBAC é facilmente ajustável a mudanças no ambiente e é largamente utilizado.

3.5.4 Gerenciamento de Identidades

O Gerenciamento de Identidades visa, basicamente, a criação e manutenção de identidades digitais de usuários em um sistema. Nesse contexto, a identidade digital é definida como sendo uma representação de uma entidade em um contexto específico. Por exemplo, em uma rede social, a identidade do usuário (entidade) é representada por seu perfil (identidade). Essas identidades são gerenciadas por um sistema chamado de **Sistema de Gerenciamento de Identidades**, este provê ferramentas para o gerenciamento dessas identidades em um ambiente digital.

O MobileHealthNet possui atualmente quatro aplicações distintas, apesar do *middleware* permitir a criação de quantas aplicações forem necessárias. Para evitar que os mesmos usuários sejam cadastrados em cada uma dessas aplicações, o *middleware* implementa um gerenciamento de identidades. Dessa forma, uma vez

cadastrado no sistema, o usuário pode acessar qualquer aplicação e serviço dentro do mesmo ambiente, respeitando suas permissões. Essa característica torna este modelo bastante eficiente em organizações fechadas, onde o usuário pode ser identificado por um atributo único em todos os serviços da empresa, como ocorre no MobileHealthNet.

3.5.5 Gerenciamento de Logs

Um log é um registro dos eventos que ocorrem dentro de sistemas de uma organização. Os logs são compostos de entradas de registro, onde cada entrada contém informações relacionadas a um evento específico que tenha ocorrido dentro de um sistema ou rede. Inicialmente os logs eram usados para registro de erros que ocorriam em um sistema, porém atualmente tem sido utilizados, também, para registro de ações de usuários de um sistema. Neste contexto, os logs podem ser usados para auditorias das atividades de usuários, a fim de encontrar ocorrências de ações maliciosas nos sistemas.

No MobileHealthNet o principal objetivo deste componente é viabilizar o processo de auditorias (Requisito 6). Através dele, o auditor poderá saber quais usuários estão acessado o sistema e sobre quais dados eles estão operando. O gerenciamento de log também é um recurso exigido pelo MC-SRES. Diante disso, o gerenciamento de log no MobileHealthNet deve ser realizado de duas formas: na primeira, chamada de Log de Operações, devem ser armazenadas todas as operações realizadas pelos usuários no sistema. Cada registro de operação contem o nome do usuário, a operação realizada, parâmetros de entrada e de saída de cada uma delas, a fim de garantir o Não Repúdio sobre todas as operações realizadas pelos usuários (Requisito 5).

Um segundo log, chamado de Log de Exceções, deve ter a finalidade de armazenar todos os erros que ocorrem no sistema, bem como o usuário para o qual o erro foi lançado e quais operações ele estava realizando, notificando-as ao administrador do sistema com o objetivo de se identificar tentativas de acesso indevido aos dados ou até mesmo tentativas de intrusão no sistema. Para fins de auditoria, os logs devem ser acompanhados de um *timestamp* que representa o momento exato de sua ocorrência. Este log pode ser usado também para verificar tentativas de acesso indevido ao sistema, uma vez que toda falha na autenticação e na autorização pode ser

armazenada. O *middleware* deve ainda permitir que o administrador configure quais informações serão armazenadas.

3.5.6 Segurança da Comunicação

Em um ambiente distribuído, principalmente em um ambiente móvel, a comunicação está sujeita a interceptações, onde as informações transmitidas podem ser lidas e interpretadas por usuários maliciosos, caracterizando um ataque de *eavesdropping*. Através deste, um usuário malicioso pode escutar a informação transmitida quando um dispositivo solicita informações de perfil do usuário a partir de um servidor de rede social, por exemplo. Se, eventualmente, as informações de um usuário forem interceptadas em um meio de comunicação entre os dispositivos da rede, as mesmas podem ser utilizadas para aplicar um ataque de *spoofing*. Em um ataque de *spoofing*, um usuário malicioso pode fingir ser um usuário cuja identificação foi interceptada (o usuário comprometido). Tais fatores tornam a segurança na comunicação um requisito indispensável nestes ambientes.

A segurança da comunicação deve garantir sobretudo as propriedades de autenticidade, integridade e confidencialidade das mensagens (Requisito 7 e 8). Neste contexto, a autenticidade se refere à origem da mensagem, onde o mecanismo de segurança deve garantir que a mensagem foi de fato enviada pela origem cuja mensagem afirma ter sido enviada. Já a integridade garante que a mensagem não foi alterada no transporte entre o emissor e o receptor. Finalmente, a confidencialidade garante que somente poderá acessar a mensagem o nó para o qual a mensagem foi destinada. Dessa forma, uma RSM segura deve garantir além da segurança das informações na camada de aplicação e serviços, a segurança na camada de comunicação, uma vez que não basta garantir a segurança nas camadas superiores sem que os dados sejam transmitidos de forma segura no ambiente distribuído.

A infraestrutura de comunicação do MobileHealthNet foi baseada na especificação DDS, conforme descrito na Seção 3.2. Contudo, até o momento não existe nenhuma especificação oficial que trate os aspectos de segurança da comunicação através do DDS. Este trabalho de mestrado apresenta como contribuição, além do modelo de segurança proposto neste capítulo, uma solução de segurança de comunicação para software que utilize o DDS como infraestrutura base. Esta solução

é apresentada em detalhes no Capitulo 4 e é composta por três componentes: um componente cujo objetivo é realizar o gerenciamento de chaves, responsável pela geração e distribuição das chaves criptográficas utilizadas no processo de criptografia dos dados; um componente responsável pela criptografia dos dados e, por fim, um componente responsável pela segurança sobre o DDS que implementa os protocolos de autenticação dos nós distribuídos no domínio.

3.5.7 Exigências Legais e Éticas

Todos os mecanismos de segurança devem levar em consideração um conjunto de questões legais e éticas, presentes em todas as camadas/mecanismos do modelo de segurança proposto. Tais questões legais e éticas dizem respeito as resoluções do Comitê de Ética em Pesquisa[1] da Universidade Federal do Maranhão (CEP/UFMA), do Conselho Nacional de Saúde[2] (órgão que trata de pesquisas em seres humanos) e por um conjunto de padrões e requisitos presentes no Manual de Certificação para Sistemas de Registro Eletrônico em Saúde.

Além dos requisitos de segurança, o MC-SRES apresenta um conjunto de normas que os sistemas destinados à saúde devem seguir. Portanto, o desenvolvimento e a implantação do MobileHealthNet em um ambiente de produção está condicionada a adoção desses princípios éticos e legais, que são específicos do domínio em questão.

3.6 Aspectos de Implementação do Modelo

No primeiro protótipo do MobileHealthNet, as aplicações são executadas em dispositivos móveis que adotam o Google Android. Por outro lado, os serviços do MobileHealthNet foram desenvolvidos como *Enterprise Java Bean* (EJB)s, os quais são implantados dentro de um servidor de aplicações, neste caso o Glassfish. Na primeira versão das aplicações, foi implementado o método de autenticação de usuário baseado na validação de credenciais informados no cadastro do usuário, neste caso o "nome do usuário"e sua "senha". Contudo, a versão atual do *middleware*

[1] `http://www.hu.ufma.br/site/estaticas/mostra_estat.php?id=38`

[2] `http://conselho.saude.gov.br/`

MobileHealthNet disponibiliza um método de autenticação que utiliza certificados digitais, que atualmente também está sendo utilizado na camada de comunicação na criação e manutenção de canais seguros de comunicação. O Glassfish disponibiliza arquivos de configuração para definir qual método de autenticação de usuário será adotado, os quais são configurados pelo desenvolvedor das aplicações.

O mecanismo de autorização possui uma granularidade a nível de métodos, ou seja para cada método de um EJB são definidos quais papeis podem executá-lo. A definição de quais métodos determinados papeis podem acessar é realizada através da configuração de arquivos XML, que são lidos pelo próprio servidor de aplicação. Esta forma de implementar a autorização facilita a manutenção da política de segurança, pois sempre que um novo usuário é adicionado ao sistema deve-se apenas definir o(s) papel(is) que ele deve exercer, para os quais já foi definido um conjunto de métodos associados aos mesmos. Além disso, o servidor de aplicação é responsável por realizar o controle de acesso aos métodos, baseado nas politicas definidas nos arquivos de configuração. As definições de papeis e de acesso são realizadas pelo administrador do sistema.

O gerenciamento de *logs* deve ser codificado de forma independente dos serviços do *middleware*, ou seja, o ideal é que o código de armazenamento do log não seja incluído no mesmo bloco do código que implementa as funcionalidades, pois não agrega nada à sua funcionalidade real. Para tanto, o processo de armazenamento de *logs* está sendo desenvolvido utilizando interceptações às chamadas dos serviços, através dos recursos de interceptações disponibilizados pelo *Java Enterprise Edition* (JEE). Por meio destes, o desenvolvedor consegue interceptar uma chamada a um método de qualquer EJB instanciado no contexto do servidor de aplicação, possibilitando a execução de instruções antes e/ou depois da execução desse método.

O estágio atual de desenvolvimento ainda não contempla a implementação do componente de Gerenciamento de Identidades, bem como o Gerenciamento de Privacidade de Usuários baseado nos *statements*. Contudo, alguns aspectos já foram definidos, por exemplo, o uso do **Modelo Centralizado** para realização do Gerenciamento de Identidade no *middleware* [53]. Neste modelo existe apenas um único provedor de identidade (idP) para todos os provedores de serviço (SP). Cada usuário do MobileHealthNet, possui um atributo único na rede social, chamado

de UserName, que é utilizado como identificador do usuário em todos os serviços providos pelo *middleware*.

Por fim, os aspectos de implementação da segurança da comunicação são apresentados na Seção 4.3, onde são descritos as especificidades da infraestrutura de comunicação utilizada no MobileHealthNet.

3.7 Análise Comparativa da Segurança Provida por *Middleware* para Redes Sociais Móveis

No Capitulo 2 foram descritos alguns *middleware* encontrados na literatura destinados ao desenvolvimento de RSMs. Esta seção descreve uma análise comparativa com relação ao suporte a segurança e privacidade provido por estes middleware e o modelo de segurança proposto nesta dissertação.

Poucos destes middleware descritos trazem algo sólido no que se refere à segurança. Em relação ao MobiClique, nenhum aspecto de segurança foi incluído no protótipo apresentado no trabalho de referência publicado.

O Mobilis provê um gerenciamento de privacidade e segurança das informações de contexto compartilhadas pelos usuários, onde somente tem acesso a uma determinada informação os usuários autorizados por seu dono. Isto é realizado através do modelo de comunicação *publish/subscribe*, onde qualquer usuário que deseja ter acesso a informações de contexto de outro deve se subscrever junto ao Serviço de Contexto. Para que a subscrição seja efetivada, o Serviço de Conteúdo solicita a permissão do dono da informação. Caso a permissão seja concedida, as informações de contexto são enviadas ao usuário solicitante toda vez que seu dono realizar uma publicação. Já a autenticação no Mobilis, é realizada através do Smack, utilizando os usuários cadastrados no servidor Openfire. Portanto, para acessar os serviços do Mobilis, os usuários necessitam se cadastrar no servidor Openfire. Cada usuário no XMPP possui um identificador único (JID), o qual é utilizado para identificar os usuários tanto pelo protocolo XMPP, quanto pelo Mobilis. A segurança da comunicação no Mobilis é feita através do protocolo *Secure Sockets Layer* (SSL) especificado como extensão do XMPP e implementado pelo Smack.

O MyNet disponibiliza um mecanismo de controle de acesso no qual o usuário determina quem terá acesso a suas informações, utilizado para garantir a privacidade dos usuários. Para tanto, o MyNet faz uso de uma estrutura chamada de Passlet, onde os usuários disponibilizam as permissões sobre seus dados, e que é armazenado pelo *middleware*. Os Passlets incluem informações de quem está dando as permissões, a quem, sobre quais informações e por quanto tempo. O MyNet utiliza os recursos de autenticação providos pelo UIA. Como visto na Seção 2.3, o UIA é uma tecnologia P2P que provê duas funcionalidades básicas: conectividade ubíqua e gerenciamento de grupos distribuídos. O MyNet também dispõe do SSL para prover a segurança de sua comunicação, contudo não deixa claro qual o protocolo base utilizado, se utiliza TCP ou UDP, ou até mesmo outro protocolo de um nível mais alto, como é o caso do XMPP no Mobilis.

O MobiSoc introduziu o conceito dos *statements* para gerenciar a privacidade dos usuários, que é destinado apenas ao compartilhamento de informações de perfis dos usuários. O MobiSoc não deixa explicito a forma que realiza a autenticação de seus usuários. Com relação a comunicação, o MobiSoc cria seu próprio canal seguro de comunicação. O algoritmo assimétrico RSA é utilizado para acordar chaves de sessão simétricas, utilizadas pelo algoritmo AES para criptografia dos dados.

No MobileHealthNet, o conceito de *statements* para gerenciamento da privacidade foi expandido a todos os recursos do *middleware*, incluindo os arquivos armazenados no Serviço de Conteúdo, as mensagens compartilhadas pelo Serviço de Fórum e Dashboard, além das informações de contexto. O MobileHealthNet possui seu próprio mecanismo de autenticação, baseado em certificados digitais. Na comunicação, o MobileHealhNet também implementa seu próprio canal seguro de comunicação, o qual é apresentado no Capítulo 4, que constitui uma importante contribuição desde trabalho de mestrado.

Além dos mecanismos comuns aos *middlewares* analisados, o MobileHealthNet apresenta diversos mecanismos adicionais que tornam seus componentes mais seguros, como o Gerenciamento de Logs, que ajuda no processo de auditorias, Autorização de Usuários baseado em papéis, que tornam a definição das atribuições de cada usuários pelo administrador mais flexíveis. Além disso, propomos o uso do Gerenciamento de Identidade, tornando mais simples o acesso às aplicações

e componentes do *middleware*, e o SSO, exigindo que o usuário se autentique apenas uma vez para utilizar qualquer aplicação do ambiente.

A Tabela 3.1 mostra um breve resumo comparativo dos mecanismos de segurança encontrados nos *middleware* analisados.

Middleware	**Arquitetura**	**Autenticação**	**Privacidade dos Dados**	**Segurança da Comunicação**
Mobilis	Centralizado	XMPP	Somente para Informações de Contexto	SSL via XMPP
MobiSoc	Centralizado	Sim, mas não explicito	Statements	RSA/AES
MyNet	P2P	Provido pelo UIA	Passlets	SSL
MobileHealthNet	Centralizado	Autenticação com SSO	Statements	Proposta de Segurança para DDS

Tabela 3.1: Resumo dos *Middleware* Analisados

3.8 Conclusão

O projeto MobileHealthNet visa o desenvolvimento de um *middleware* para o suporte ao desenvolvimento de aplicações para saúde. Sua arquitetura foi desenvolvida a partir de requisitos específicos, onde foram elicitados também os requisitos de segurança necessários para prover a interação segura entre os usuários da rede social a ser formada.

A arquitetura do *middleware* MobileHealthNet é centralizada e se utiliza dos protocolos RUDP e DDS para prover a comunicação, a primeira provendo interações entre clientes e *gateway* e a segunda entre *gateway* e provedores de serviços. As interações entre as aplicações são realizadas através de interações entre publicadores

e consumidores de dados (*publish/subscribe*). Por ser um *middleware* para construção de rede social destinada á área da saúde, o MobileHealthNet tem como exigência um conjunto de requisitos de segurança.

Este trabalho propõe um modelo de segurança desenvolvido com o objetivo de suprir os principais requisitos de segurança do MobileHealthNet, envolvendo todas as camadas do sistema, desde a camada de aplicação até a camada de comunicação. Seu desenvolvimento teve como elemento chave o uso de uma metodologia para desenvolvimento de software seguro chamada CLASP. Além disso, foi levado em consideração as recomendações e requisitos presentes no MC-SRES.

4 Segurança da Comunicação no MobileHealthNet

A comunicação em ambiente móveis está sujeita a interceptações, isso porque as mensagens trafegam livremente pelo meio e podem ser facilmente capturadas por qualquer dispositivo. Entretanto, existem meios de tornar essa comunicação segura, de modo que mesmo que a mensagem seja interceptada, o intruso não poderá ter acesso ao conteúdo da mesma. Isto é realizado através do estabelecimento de canais seguros de comunicação. Um canal seguro de comunicação protege emissor e receptor contra interceptações, modificações e invenção de mensagens [48]. Porém, não protege, necessariamente, contra interrupção de mensagens. As mensagens que trafegam por esses canais não podem ser interpretadas por intrusos, garantindo assim a confidencialidade de mensagens. A proteção contra modificação e invenção é garantida pela autenticação mútua dos interlocutores envolvidos e garantia da integridade das mensagens.

A criação de um canal seguro de comunicação é feita através da aplicação de algoritmos de criptografia. Criptografia refere-se a um conjunto de conceitos e técnicas que visam codificar uma informação de forma que somente o emissor e o receptor possam acessá-la, evitando que um intruso consiga interpretá-la. A criptografia é utilizada em sistemas computacionais para garantir a segurança da informação, a fim de manter a confidencialidade, integridade dos dados e autenticação das entidades. Toda criptografia é composta de três partes [53]: (I) um algoritmo de criptografia, (II) a mensagem original (também chamada de "texto limpo") e (III) a mensagem criptografada (resultado da criptografia).

Os algoritmos de criptografia em sistemas computacionais utilizam o conceito de chaves criptográficas. Uma chave criptográfica é basicamente um conjunto de bits baseados em um determinado algoritmo utilizada para codificar e de decodificar informações. Se o receptor da mensagem usar uma chave incompatível com a chave do emissor, ele não conseguirá extrair a informação a partir da mensagem criptografada. Baseado no tipo de chave utilizada na criptografia, os algoritmos

são divididos dois grupos: algoritmos de criptografia simétrica e algoritmos de criptografia assimétrica. A criptografia simétrica ocorre quando a mesma chave criptográfica é compartilhada pelas duas entidades envolvidas na comunicação. A chave que é utilizada para cifrar a mensagem é a mesma utilizada para decifrar. Na criptografia assimétrica, cada participante possui um par de chaves: uma privada e outra publica. A chave privada é mantida em segredo e é conhecida apenas pelo seu detentor, enquanto que a chave publica é disponibilizadas a todos. Neste contexto, uma chave é utilizada para cifrar e a outra para decifrar uma mensagem.

Dependendo do ambiente, o número de chaves envolvidas no processo de criptografia pode ser muito alto, o que dificulta o gerenciamento dessas chaves. O gerenciamento de chaves consiste de um conjunto de técnicas e procedimentos que visam permitir o estabelecimento, a distribuição e a manutenção de chaves criptográficas entre entidades. Em outras palavras, visa garantir a segurança dessas chaves criptográficas. Este processo inclui também a revogação (ou invalidar) de chaves quando a mesma foi exposta a risco. O estabelecimento de chaves é o processo onde duas entidades que estão interessadas em criar um canal seguro estabelecem uma chave secreta compartilhada para ser utilizada para criptografia das mensagens. A distribuição de chaves, por sua vez, é o processo onde as chaves iniciais são distribuídas de forma segura para que as partes comunicantes tenham acesso.

O *middleware* MobileHealthNet tem como característica o uso do DDS na camada de comunicação. Uma característica do DDS é ser um protocolo que permite o uso de *multicast* quando disponível, o que torna o envio de mensagens mais eficiente. Quando um nó publica algum tópico, o mesmo será distribuído para todos os subscritos naquele tópico. Essa característica necessita ser mantida pelos mecanismos de segurança [23]. Existem diversos protocolos para criação de canais seguros de comunicação, tais como TLS (*Transport Layer Security*) [16], DTLS (*Datagram Transport Layer Security*) [40], SSL (*Secure Sockets Layer*) [2], entre outros. Contudo, nenhum destes podem ser aplicados diretamente ao DDS, pois utilizam mensagens *unicast*, e suas chaves são trocadas somente pelos dois interlocutores. Portanto, o uso desses protocolos de segurança sobre o DDS implicaria na perda do *multicast*, que corresponde a uma importante propriedade do DDS.

Atualmente, não há uma especificação de segurança definida pela *Object Management Group* (OMG) para o DDS. No entanto, algumas propostas defendem o

uso de uma abordagem baseada no protocolo SRTP (*The Secure Real-time Transport Protocol*) [7]. O SRTP é um protocolo para o estabelecimento de comunicação segura desenvolvido para o RTP (*Real-time Transport Protocol*) [42], o qual possui suporte para comunicação *multicast*. Porém, o SRTP foi desenvolvido especificamente para o RTP e sua transcrição para o DDS não é algo simples pois, diferentemente do RTP, o DDS foi idealizado sobre o paradigma *publish/subscribe*. Além disso, a implementação do SRTP deve ser realizada dentro do *middleware* DCPS que implementa a especificação DDS, uma vez que seria necessário a inclusão de metadados que compõem o protocolo, como *timestamp*, *sequence number*, entre outros. A inclusão destes metadados em uma camada acima do *middleware* implicaria em fortes modificações nos tópicos das aplicações, uma vez que o desenvolvedor deveria incluir em seus tópicos atributos relativos a esses metadados. Tais fatores impossibilitam o uso deste protocolo no MobileHealthNet, dado que não temos acesso ao código do *middleware* DCPS utilizado para incluir essas modificações sem o comprometimento dos tópicos. O *middleware* DCPS utilizado no MobileHealthNet é o CoreDx[1].

Este trabalho propõe uma solução para criação de canais seguros de comunicação destinados a infraestruturas que utilizem o DDS como protocolo base. Neste capítulo são abordados algoritmos e protocolos usados na construção de um canal seguro para o MobileHealthNet. Além disso, são descritos aspectos referentes ao gerenciamento de chaves, bem como os algoritmos de criptografia utilizados. Contudo, para entender melhor os aspectos de implementação, faz-se necessário conhecer como funciona a camada de comunicação do MobileHealthNet. Para tanto, a sessão seguinte descreve a arquitetura da infraestrutura de comunicação e sua API.

4.1 Infraestrutura de Comunicação do MobileHealthNet

A infraestrutura de comunicação do MobileHealthNet provê uma interface de comunicação *publish/subscribe* voltada a colaboração em RSMs [6]. Esta baseia-se em dois protocolos: o *Mobile Reliable UDP* (MRUDP) [13] para comunicação entre clientes móveis (*Mobile Users*) e o núcleo da rede (*Core Network*) e o DDS [22] para comunicação dentro do núcleo da rede do MobileHealthNet, conforme mostra a Figura 4.1.

[1]http://www.twinoakscomputing.com/coredx

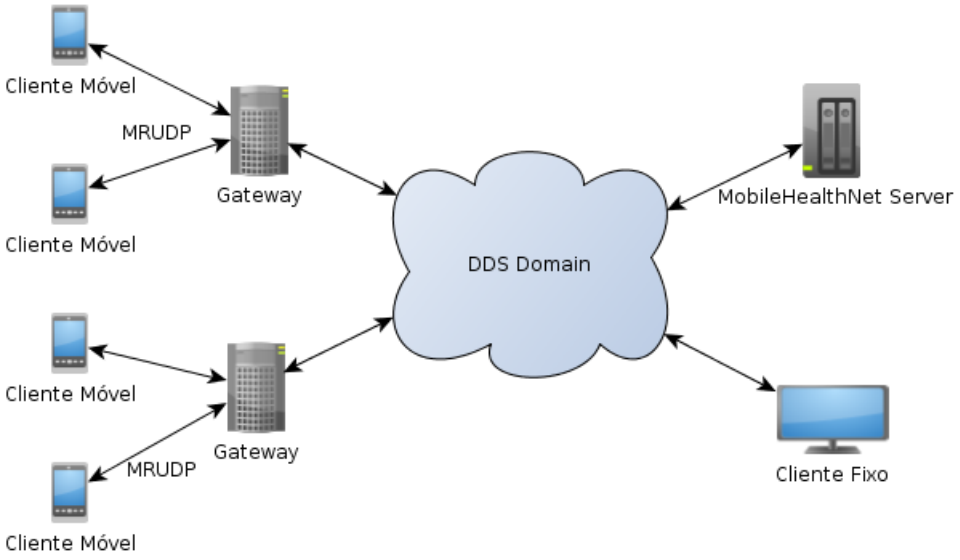

Figura 4.1: Arquitetura da Infraestrutura de Comunicação do MobileHealthNet

O MRUDP é a base para comunicação entre clientes móveis e o *gateway*. O MRUDP utiliza o protocolo UDP com a inclusão do mecanismo de confiabilidade da entrega de mensagens, travessia de Firewalls e NAT (do inglês *Network Address Translation*), suporte a mudança de endereços de IP e por fim, reduzido consumo dos recursos computacionais dos dispositivos móveis. Cada mensagem, seja de um cliente para o *gateway* ou do *gateway* para o cliente, necessita de uma mensagem para confirmação de recebimento (um *ack*), e cada transmissão é feita várias vezes antes de se considerar a perda da conexão do cliente com o *gateway*.

Para desenvolver o núcleo da infraestrutura de comunicação do MobileHealthNet foi escolhida a especificação DDS. O MobileHealthNet utiliza o CoreDx como *middleware* DDS, o qual utiliza o UDP como protocolo da camada de transporte. O DDS foi idealizado sobre um modelo *Data-Centric Publish-Subscribe* (DCPS) baseado em tópicos. Especificamente, o *middleware* DCPS gerencia automaticamente a entrega das mensagens, sem requerer qualquer intervenção das aplicações, o que inclui quem deve receber a mensagem, a localização de quem envia ou recebe mensagens e o que acontece se a mensagem não é entregue. Ao utilizar a especificação DDS, o desenvolvedor se preocupa somente com o conteúdo que será transferido entre as aplicações. O DDS possui as seguintes entidades envolvidas na transferências de informações:

- *Domain*: As aplicações DDS enviam e recebem dados dentro de um domínio. Somente participantes do mesmo domínio podem se comunicar, o que ajuda a isolar e otimizar a comunicação dentro de uma comunidade que partilha interesses em comum.

- *Data Writers*: As aplicações podem utilizar *data writers* para publicar dados no espaço global de dados em um determinado domínio.

- *Data Readers*: Recebem os dados publicados pelos *data writers*.

- *Publisher*: Um *publisher* cria e gerencia um grupo de *data writers*.

- *Subscriber*: Um *subscriber* cria e gerencia *data readers*.

- *Topics*: Um tópico conecta um *data writer* a um *data reader*. A comunicação ocorre somente se o tópico publicado por um *data writer* equivaler ao tópico subscrito pelo *data reader*. A comunicação via tópico é anônima e transparente, *publishers* e *subscribers* não precisam se preocupar como os tópicos são criados nem com quem está escrevendo ou lendo o tópico, pois o *middleware* DCPS DDS gerencia estas questões.

No DDS, toda comunicação se dá através da publicação e subscrição em tópicos. O DDS disponibiliza recursos que podem ser utilizados como filtros no processo de subscrição. Os filtros são descritos através de expressões com uma sintaxe bem definida, as quais são baseadas nos atributos de cada tópico. Os filtros são essenciais, pois diminuem a carga com tratamento de mensagens que não são do interesse do usuário. Por exemplo, um usuário pode estar interessado em receber somente os tópicos publicados pelo usuário cujo atributo ”user_name” seja igual a ”João”. Esta restrição é expressa através de um filtro.

Tendo como protocolos base essas duas tecnologias (MRUDP e DDS), a infraestrutura de comunicação do MobileHealthNet disponibiliza uma API *publish/subscribe* que contém os métodos mostradas no Código 4.1.

Código 4.1: Interfaces da Infraestrutura de Comunicação do MobileHealthNet

```
1
2 public interface NodePubSubService
3 {
```

```
4    public void publish(Object ddsTopic)
5       throws DomainParticipantNotCreatedException,
6          TopicNotRegisteredException;
7
8    public String subscribe(PubSubTopicListener listener,
9       Object ddsTopic)
10      throws DomainParticipantNotCreatedException,
11         TopicNotRegisteredException;
12
13   public String subscribe(PubSubTopicListener listener,
14      Object ddsTopic, AppendExpression expression)
15            throws DomainParticipantNotCreatedException,
16               TopicNotRegisteredException;
17
18   public void unsubscribe(String topicName,
19      String subscribeId);
20
21   public void addSystemExceptionListener(
22      SystemExceptionListener exceptionListener);
23
24 }
```

A interface `NodePubSubService` possui os métodos necessários para publicação e subscrição em tópicos. O método `publish(Object ddsTopic)` é usado para publicação de um tópico no domínio. Para subscrição, a interface disponibiliza dois métodos: o primeiro sem a inclusão de filtros, enquanto o segundo especifica um filtro através do parâmetro `AppendExpression`. A classe `AppendExpression` é usada para montar dinamicamente a expressão usada como filtro pelo DDS. Ao se subscrever, a aplicação passa um objeto que implementa a interface `PubSubTopicListener`, que será notificado sempre que um novo tópico for publicado no domínio, obedecendo os filtros eventualmente definidos. Além disso, a interface disponibiliza também o método `unsubscribe`, usado quando a aplicação deseja deixar de receber informações de uma subscrição.

4.2 Uma Proposta para Segurança da Comunicação em Sistemas Baseados no DDS

Nesta seção, são abordados os requisitos de segurança relacionados a comunicação no MobileHealthNet, definidos a partir dos requisitos gerais do projeto e da análise da infraestrutura de comunicação desenvolvida para o MobileHealthNet. Em seguida, são mostradas as principais características da proposta, bem como seus aspectos de implementação no *middleware* MobileHealthNet.

4.2.1 Requisitos de Segurança para Comunicação no MobileHealthNet

Uma importante característica do DDS está na possibilidade do uso da comunicação *multicast*, que dificulta o estabelecimento de um canal seguro, pois as chaves necessitam ser distribuídas entre os diversos membros envolvidos na comunicação. Esta distribuição de chaves criptográficas deve também ser realizada de forma segura. Portanto, qualquer protocolo de segurança para o DDS deve levar em consideração que as mensagens serão distribuídas para mais de um destinatário e todos eles devem poder decifrar-las.

O DDS é um protocolo *publish/subscribe* baseado nos dados, onde as mensagens, representadas por tópicos, são publicados no domínio e distribuídas para todos os subscritores interessados. Para informar quais subscritores estão interessados em determinados tópicos, o DDS disponibiliza meios de filtragem das mensagem baseados em seu conteúdo. Portanto, o uso de filtros também corresponde a uma importante propriedade dos DDS, que deve ser mantida pelos mecanismos de segurança.

No MobileHealthNet, faz-se necessário que os processos de configuração dos mecanismos de segurança exijam o mínimo possível de intervenção dos usuários, uma vez que os usuários MobileHealthNet não possuem muita experiencia com dispositivos computacionais. Também é necessário que os desenvolvedores de aplicações para o MobileHealthNet não se preocupem com os detalhes de baixo nível dos mecanismos de segurança, como criptografia dos tópicos. Tais funcionalidades

devem ser realizadas o mais transparentemente possível do desenvolvedor das aplicações.

O custo computacional adicional gerado pela inclusão de protocolos de autenticação e algoritmos de criptografia deve ser a menor possível, de modo que não influencie significativamente no desempenho da comunicação e nem interfira na qualidade do serviço disponibilizado. É imprescindível que os mecanismos de segurança não prejudiquem a escalabilidade do sistema, outra importante característica da infraestrutura de comunicação do MobileHealthNet. Além disso, o custo computacional gerado por esses mecanismos também deve ser levado em consideração, uma vez que o MobileHealthNet é destinado a dispositivos móveis, que possuem baixo poder de processamento e pouca memória.

Portanto, os principais requisitos de segurança para a camada de comunicação do MobileHealthNet são:

- Garantir as propriedades básicas de segurança: integridade, autenticidade e confidencialidade;
- Manutenção da comunicação *multicast*, quando disponível;
- Permitir o uso de filtros do DDS;
- Os mecanismos de segurança devem exigir o mínimo possível de conhecimento por parte do usuário final;
- Disponibilizar mecanismos de segurança da forma mais transparente possível para o desenvolvedor das aplicações, de modo que ele tenha que realizar poucas alterações em seu código para incluir o uso de segurança;
- Garantir que os mecanismos de segurança não prejudiquem a escalabilidade do sistema;
- Os mecanismos de segurança devem possuir baixo consumo de recursos computacionais (processamento e memória), pois são destinados a execução em dispositivos móveis.

A descrição desta proposta está dividida em duas partes. A primeira descreve a arquitetura básica da solução, mostrando os principais componentes

envolvidos no processo. A segunda descreve os protocolos utilizados para a publicação e subscrição de forma segura de nós a tópicos.

4.2.2 Arquitetura da Solução Proposta

A criação de canais seguros de comunicação tem como requisito básico a autenticação dos nós envolvidos. Como vimos no Capitulo 3, sistemas de computação voltados para a área da saúde exigem um severo nível de segurança. Para tanto, a solução proposta neste trabalho de mestrado faz uso de certificados digitais no processo de autenticação, objetivando manter o alto nível de segurança exigido por esses sistemas. Portanto, esta solução utiliza uma **Autoridade de Certificados (AC)**, responsável pela validação dos certificados digitais de cada nó.

No protótipo desenvolvido, todos os certificados digitais são gerados por meio da ferramenta OpenSSL[2] e são incluídos em cada nó que deseja fazer parte do sistema, seja este um dispositivo móvel ou outro componente que acesse diretamente o ambiente pelo domínio DDS. A AC possui certificado digital autoassinado, contudo os demais certificados são assinados utilizando o certificado da AC.

O processo de validação do certificado consiste de diversas etapas, como a verificação da assinatura digital, para validar se o mesmo foi assinado por uma autoridade certificadora confiável, além da verificação da lista de certificados revogados. Uma outra etapa inclui a validação da cadeia de certificados até o topo da cadeia, ou seja, a autoridade certificadora raiz, cujo certificado é autoassinado.

Cada tópico seguro do ambiente possui uma chave simétrica, que é utilizada para cifrar e decifrar os dados publicados no mesmo. O uso de várias chaves simétricas (uma para cada tópico) torna o ambiente mais seguro, pois para ter acesso a todos os dados um intruso necessitaria obter todas as chaves simétricas. A chave simétrica é distribuída a todos os nós que publicam ou se subscrevem ao tópico. Como ela é distribuída a vários nós móveis, ela está mais propensa a ser descoberta por um atacante que venha a obter acesso a um destes nós. Por este motivo, a chave é substituída periodicamente. Este período é configurado pelo próprio administrador do sistema. Quando o prazo de validade de uma chave está vencido, dizemos que a chave expirou e uma nova chave é gerada para aquele tópico. A geração e manutenção dessa

[2]http://www.openssl.org/

chave é feita pelo componente *Key Distribution Center* (KDC). O KDC é responsável pela autenticação e pela geração e distribuição das chaves simétricas. Toda vez que um nó desejar publicar um tópico de forma segura, o mesmo deverá se autenticar junto ao KDC para obter a chave simétrica referente ao tópico desejado. O mesmo ocorre para o nó que deseja decifrar um tópico seguro publicado no domínio. O KDC mantem uma base de dados atualizada contendo todas as chaves simétricas usadas na criptografia dos tópicos, uma chave para cada tópico. A Figura 4.2 ilustra esta arquitetura, que exige a inclusão de no mínimo dois componentes no domínio DDS: a AC e uma KDC

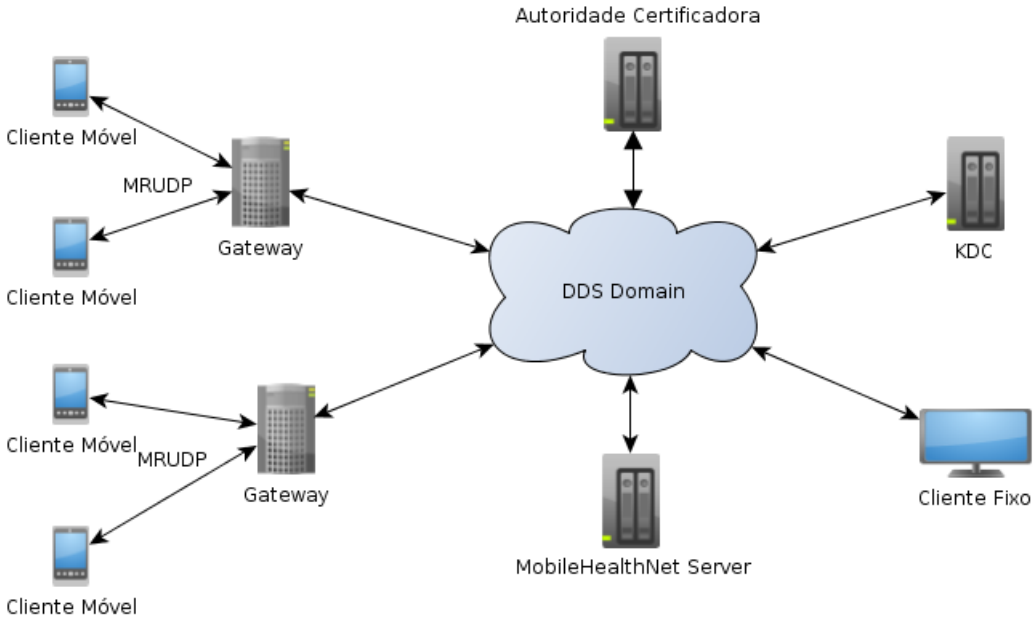

Figura 4.2: Componentes de Segurança Dispostos no Domínio DDS

No cliente estão presentes os recursos necessários para execução do protocolo de autenticação e distribuição das chaves, além da implementação dos mecanismos de criptografia dos tópicos desejados. O protocolo de autenticação e distribuição das chaves é sempre iniciado pelo cliente, que informa ao KDC de qual tópico o mesmo deseja obter a chave simétrica, além de enviar seu certificado digital. A publicação e subscrição de tópicos no cliente são disponibilizadas para as aplicações através de uma API, descrita com mais detalhes na Seção 4.3.

4.2.3 Protocolo de Autenticação e Distribuição de Chaves Simétricas

Na abordagem desenvolvida, a aplicação ao realizar uma publicação em um tópico pode faze-lo de forma segura ou não. Para indicar que a publicação deve ser criptografada utiliza-se um parâmetro booleano disponibilizado pela API de acesso ao *middleware* para as aplicações. Caso a aplicação indique que a publicação deve ser realizada de forma segura, o middleware deve providenciar a chave simétrica associada ao tópico, caso não a tenha obtido anteriormente. Para tanto, o protocolo de autenticação e distribuição de chaves, ilustrado na Figura 4.3, é executado. O protocolo de autenticação e distribuição de chaves também será automaticamente iniciado quando o middleware, ao receber uma instância de um tópico ao qual a aplicação tenha se subscrito, ainda não tiver a chave simétrica capaz de decifrá-la.

Além disso, um terceiro cenário pode causar a execução automática desse protocolo. Neste caso, o protocolo é iniciado quando o *middleware* descobre que a chave simétrica relacionada ao tópico a ser acessado já expirou, exigindo que o *middleware* obtenha a chave mais atual referente ao tópico desejado.

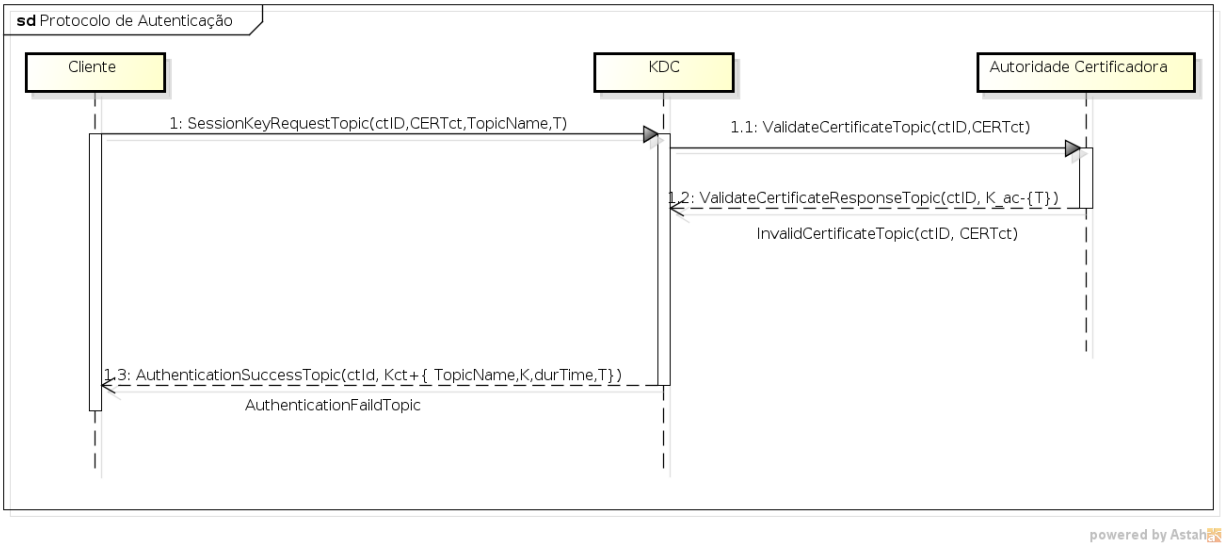

Figura 4.3: Protocolo de Autenticação e Distribuição de Chaves Simétricas

O protocolo de segurança mostrado na Figura 4.3 foi desenvolvido para o DDS. Portanto, todas as mensagens são encapsuladas em tópicos, os quais são publicados no domínio. Dessa forma, foram criados seis tópicos a fim de prover o transporte das informações necessárias dentro do domínio, descritos a seguir:

1. `SessionKeyRequestTopic`: Usado para requisitar a chave simétrica correspondente a um determinado tópico usado pela aplicação. Ele é utilizado sempre que um cliente deseja publicar ou decifrar uma instância de um tópico seguro;

2. `ValidateCertificateTopic`: Usado para requisitar uma validação de certificado;

3. `ValidateCertificateResponseTopic`: Representa a resposta a uma validação de certificado;

4. `InvalidCertificateTopic`: Representa a resposta quando um certificado foi considerado inválido;

5. `AuthenticationSuccessTopic`: Representa uma autenticação bem sucedida;

6. `AuthenticationFailedTopic`: Representa uma autenticação falha.

O protocolo se inicia com a publicação do tópico `SessionKeyRequestTopic` (Mensagem 1). Este tópico contém o identificador do cliente ($ctID$), o certificado digital do cliente ($CERTct$), o nome do tópico de aplicação que será publicado de forma segura (`TopicName`) e por fim, um **ticket** (`T`), caso a aplicação já possua.

O ticket é uma informação gerada pela AC toda vez que um certificado é validado. Seu objetivo é reduzir o tempo gasto com o processo de autenticação, pois diminui a quantidade de validações e requisições para AC, conforme segue. O processo de validação de certificado é custoso, pois inclui diversos passos, como a validação da assinatura digital do certificado, validação de sua cadeia de certificados e verificação da Lista de Certificados Revogados. Dessa forma, uma vez validado o certificado, a AC gera um ticket contendo um *hash* do certificado do cliente e o tempo de duração deste ticket. Portanto, se ocorrer um processo de autenticação subsequente, o KDC apenas validará o ticket. Caso o mesmo seja válido, nenhuma requisição à AC é realizada. No entanto, se o prazo de validade do ticket esteja vencido, ou seja, o ticket tenha expirado, o KDC requisita uma nova validação do certificado do cliente. O ticket é criptografado com a chave privada da AC ($Kac-$) para garantir que somente a Autoridade Certificadora possa gerar essa informação.

A validação do certificado ocorre através de uma requisição à AC. Neste caso, o KDC publica uma instância do tópico `ValidateCertificateTopic` (Mensagem 1.1) contendo o identificador do cliente ($ctID$) e seu certificado ($CERTct$). Em seguida, aguarda por um tópico `ValidateCertificateResponseTopic` (Mensagem 1.2), contendo o identificador do cliente ($ctID$) e o ticket referente ao certificado do mesmo. O KDC, então, realiza o processo de validação do ticket a fim de se certificar que a informação tenha sido gerada pela AC. Caso o cliente envie o ticket na requisição da chave simétrica, o KDC realiza primeiramente a validação do ticket e só envia uma requisição para validação do certificado à AC se o ticket estiver com o prazo de duração vencido. Caso o certificado seja inválido, a AC publica um tópico `InvalidCertificateTopic`, que contém o $ctID$ do cliente e o certificado apresentado.

Uma vez validado o ticket, o KDC obtém a chave publica do cliente ($Kct+$) contida em seu certificado, bem como a chave simétrica referente ao tópico requisitado e o envia ao cliente através do tópico `AuthenticationSuccessTopic` (Mensagem 1.3). A chave simétrica é enviada juntamente com o nome do tópico, o tempo de duração restante da chave e o ticket. O conteúdo enviado no tópico `AuthenticationSuccessTopic` é criptografado com a chave publica do cliente ($Kct+$), garantindo que somente o cliente destinatário possa decifrar a mensagem, para que eventuais intrusos que estejam escutando no domínio não tenham acesso à chave simétrica. Caso a autenticação seja mal sucedida, o KDC publica o tópico `AuthenticationFaildTopic` contendo apenas o $ctID$ do cliente.

4.2.4 Gerenciamento das Chaves Simétricas

Quando o KDC gera uma chave simétrica, ele atribui a mesma o *timestamp* da geração e um tempo de duração, este último de acordo com o que for definido pelo administrador do sistema. O tempo de duração indica por quanto tempo a chave permanecerá válida quando, então, deve ser trocada conforme explicado anteriormente. Uma possível abordagem para implementar esta substituição de chave seria implementar um procedimento de notificação a todos os nós móveis de sua expiração, fornecendo-se a nova chave gerada para o tópico. No entanto, esta abordagem geraria uma sobrecarga de comunicação concentrada em um único instante

de tempo, já que podem existir potencialmente milhares de dispositivos móveis que utilizem o tópico.

Para contornar este problema, o processo de substituição das chaves simétricas nos nós móveis é realizado por demanda, ou seja, somente quando há uma requisição para publicação ou leitura de uma instância do tópico. Ao requisitar uma chave simétrica ao KDC, o nó móvel obtêm, além da chave, o tempo de duração restante para a mesma, que corresponde ao tempo de geração da chave mais o tempo de duração definida para a mesma decrementando-se do valor resultante o tempo atual no KDC no ato do processamento da requisição. O *middleware* cliente armazena em uma base de dados local a chave obtida, o tópico ao qual ela se refere, o tempo restante de duração da mesma e o *timestamp* local do momento do recebimento da chave. Ao receber da camada de aplicação uma solicitação para publicação em um tópico, o *middleware* cliente verifica se a chave simétrica correspondente ao tópico encontra-se válida. Para tanto, ele calcula o tempo atual menos o tempo no qual ele recebeu a chave e verifica se este valor é menor ou igual ao tempo de duração restante para a mesma. Se for menor ou igual, a chave ainda é valida e estará vencida caso contrário.

A abordagem adotada leva em consideração que podem haver diferenças nos relógios dos clientes em relação ao relógio do KDC. Em um primeiro cenário, o relógio do cliente (publicador ou subscritor) pode estar mais rápido que o relógio do KDC. Neste caso, o cliente equivocadamente calculará que sua chave expirou antes que o KDC, gerando uma requisição de uma nova chave. Ao receber essa requisição, o KDC retornará a mesma chave, porém com o tempo de validade atual. Por exemplo, se o KDC recebe uma requisição referente a uma chave cujo tempo de validade é de oito minutos e já se passaram cinco desde a geração da mesma, o KDC envia ao cliente o tempo de validade de três minutos.

Por outro lado, pode ocorrer que o relógio do cliente publicador seja mais lento que o relógio do KDC. Neste caso, o cliente pode calcular erradamente que a chave ainda não expirou e continuar publicando com a mesma. A solução para esse problema é resolvido no subscritor. Mesmo que o subscritor já tenha obtido a nova chave correspondente ao tópico (que teria ocorrido em decorrência do recebimento a partir de um outro publicador de uma instância do tópico criptografada com a chave nova), ele mantêm em sua base de dados a chave antiga do tópico por um intervalo de tempo configurável pelo administrador do sistema. Desta forma, ele sempre tentará

descriptografar a instância do tópico com a chave atual e, caso não consiga, utilizará a chave antiga.

Contudo, pode ocorre que um subscritor entrou no ar recentemente e ainda não recebeu nenhuma chave. Ao receber um tópico, este subscritor obtém a chave junto ao KDC, porém em virtude do erro no relógio do publicador (relógio mais lento que o relógio do KDC), seus tópicos ainda estão criptografados com uma chave antiga, o que impossibilita o subscritor de decifrar a mensagem. Para solucionar esse problema, o KDC também mantém a chave antiga por um intervalo de tempo configurado pelo administrador. Durante esse intervalo o KDC envia as duas chaves ao cliente, a chave antiga e a chave mais atual. Desta forma, o subscritor poderá descriptografar a instância do tópico seguindo a mesma ordem do caso anterior, inicialmente tentará com a chave atual e, caso não consiga, utilizará a chave antiga.

4.2.5 Escalabilidade da Solução

A manutenção da escalabilidade é um dos principais requisitos da infraestrutura de segurança. Nesse contexto, a escalabilidade está relacionada ao número de nós clientes conectados ao mesmo tempo e ao número de tópicos publicados no domínio. O Gateway é um componente importante nesta solução, pois é responsável por intermediar a comunicação entres os dispositivos móveis e o núcleo da rede. A solução de segurança proposta neste trabalho não inclui nenhum *overhead* aos Gateways, uma vez que não precisam cifrar e nem decifrar os tópicos publicados, mantendo assim a escalabilidade do sistema considerando-se este componente da arquitetura. Além disso, essa característica garante a integridade e confidencialidade das mensagens também por parte dos Gateways, que não se tornam um ponto de vulnerabilidade do sistema. Entretanto, esta solução inclui dois componentes centralizados: o KDC e a AC. Estes componentes não são utilizados durante as publicações e leituras de tópicos, o que já garante uma boa escalabilidade, como podemos observar nos experimentos apresentados no Capítulo 5. O uso do ticket no protocolo diminui o número de requisições realizadas à AC, tornando o número de requisições recebidas pelo KDC maior que o número de requisições recebidas pela AC. Desse modo, o KDC representa um ponto mais crítico ao sistema, podendo se tornar um problema ao desempenho caso o número de clientes seja muito alto.

Para manter a escalabilidade do sistema propomos a inclusão de mais instâncias do KDC no domínio, quando isto se fizer necessário. Isto requer a implementação de um mecanismo de balanceamento de carga das chaves simétricas a serem mantidas pelas diferentes instâncias do KDC. Neste contexto, cada instância do KDC seria responsável por gerenciar um conjunto de tópicos. Dessa forma, todas as instâncias receberiam as requisições, porém somente a instância que gerenciaria o tópico requisitado a processaria e enviaria ao cliente a chave simétrica correspondente a ele. Opcionalmente, pode-se balancear as requisições. Neste caso, as requisições são distribuídas entre as diferentes instâncias do KDC através de um componente externo, onde apenas uma instância recebe uma determinada requisição e a responde ao cliente. No entanto, esta abordagem exige que todas as instâncias possuam acesso a uma mesma base de dados centralizada ou que sejam mantidas réplicas da base de dados que devem ser mantidas atualizadas e consistentes.

4.3 Aspectos de Implementação no MobileHealthNet

Os mecanismos de segurança foram desenvolvidos como uma camada adicional entre a infraestrutura de comunicação e as aplicações. A interface *publish/subscribe* (`NodePubSubService`) da infraestrutura de comunicação do MobileHealthNet, vista na Seção 4.1, foi estendida através do mecanismo de herança por uma nova interface chamada `SecureNodePubSubService`. Nela estão todos os métodos necessários para publicação e subscrição de tópicos de forma segura. Essa interface deve ser utilizada pelas aplicações ou serviços que desejam se comunicar de forma segura no domínio DDS. O Código 4.2 mostra os métodos dessa interface.

Código 4.2: Interface para Publicação e Subscrição Segura

```
1
2 public interface SecureNodePubSubService
3    extends NodePubSubService{
4    public void publish(Object ddsTopic, boolean secure)
5          throws DomainParticipantNotCreatedException,
6          TopicNotRegisteredException;
7
8    public String subscribe(PubSubTopicListener listener,
```

```
9           Object ddsTopic)
10          throws DomainParticipantNotCreatedException,
11          TopicNotRegisteredException;
12
13      public String subscribe(PubSubTopicListener listener,
14          Object ddsTopic, AppendExpression expression)
15          throws DomainParticipantNotCreatedException,
16          TopicNotRegisteredException;
17
18      public void requestSessionKey(Object ddsTopic);
19  }
```

Perceba que os métodos da interface `SecureNodePubSubService` são bem semelhantes aos métodos de sua interface base, acrescentando apenas um parâmetro do tipo **boolean** que é utilizado para indicar se a publicação será realizada de forma segura ou não. Contudo, um novo método denominado `requestSessionKey()` foi acrescentado para dar mais flexibilidade para as aplicações. Através deste método, as aplicações podem solicitar explicitamente a chave simétrica correspondente a um determinado tópico.

Assim como na infraestrutura de comunicação do MobileHealthNet, a camada de segurança possui duas classes concretas que implementam os métodos da interface `SecureNodePubSubService`. Uma é destinada para uso nos dispositivos móveis, chamada de `SecureClientPubSubManager`. Neste caso, o acesso ao domínio é realizado através dos *gateways*. A outra classe, chamada de `SecureDDSPubSubManager`, a qual é destinada para uso de nós que desejam acessar diretamente o domínio DDS, conforme mostra o diagrama de classe apresentado na Figura 4.4.

A implementação dos métodos da interface, bem como os atributos comuns às duas classes concretas, estão implementadas na classe `SecurePubSubManager`. Nessa classe está implementado todo o protocolo de autenticação e distribuição de chaves simétricas, que será executado tanto pelos clientes móveis, quanto pelos nós que acessam diretamente o domínio.

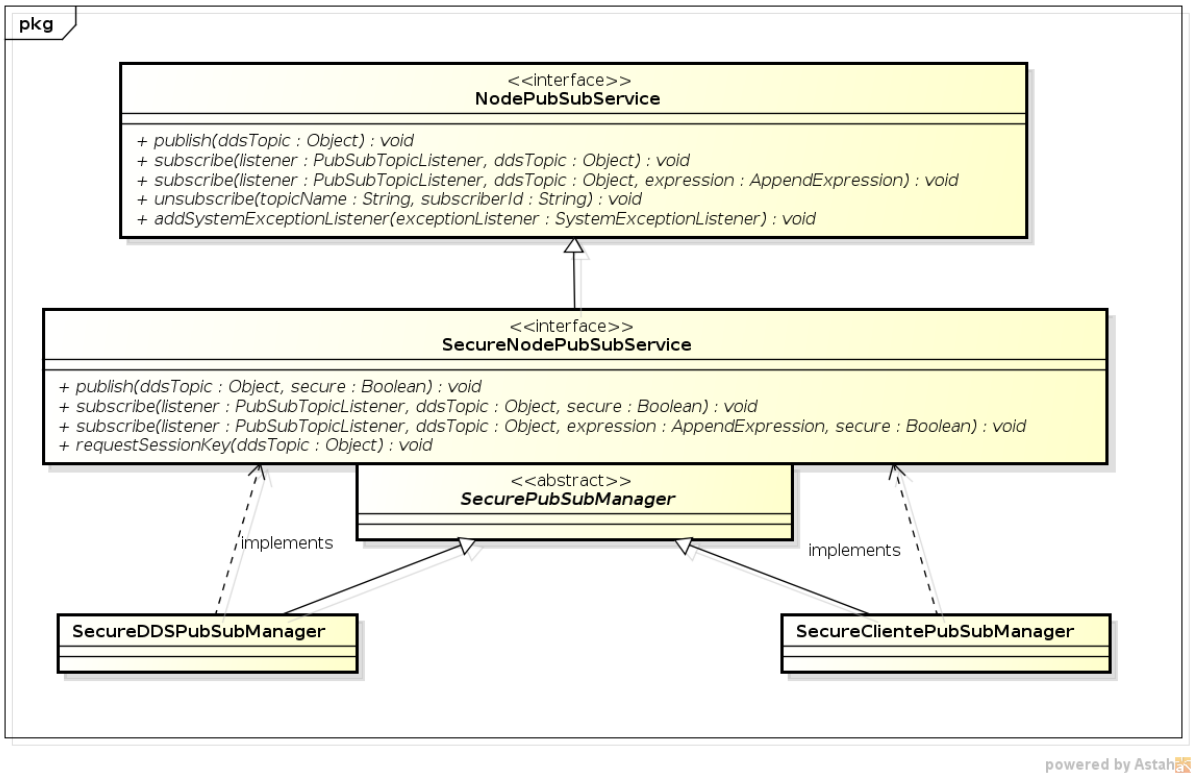

Figura 4.4: Classes que Implementam os Mecanismos de Segurança

A especificação DDS define uma sintaxe própria para criação de expressões destinada ao uso dos filtros. Visando obter uma maior flexibilidade no momento da construção da expressão, um conjunto de classes foi implementado, utilizando-se o padrão *Decorator* [19], que permite adicionar comportamentos a objetos já existentes em tempo de execução. O objetivo desta implementação é permitir, de modo prático, a definição em tempo de execução de novos filtros a serem aplicados nos leitores de dados. Isto permite que aplicações criem filtros, especificando quais dados de um determinado tópico desejam receber a partir dos atributos do tópico. A Figura 4.5 ilustra as classes criadas com este fim.

A classe `AppendExpression` representa a expressão DDS. Por meio desta classe são geradas as expressões que representam as operações de comparação, seja de igualdade (=), diferença (≠), maior que (>), menor que (<), entre outros. A interface `Appendable` fornece um único método chamado `append()`. As classes `AndAppend` e `OrAppend` implementam `Appendable` e representam os operadores lógicos *and* e *or*. Diferente das demais cláusulas, estas expressões não necessitam de atributo e valor, mas sim de dois objetos `Appendable`, pois ambas unirão um conjunto de expressões completas (p.e., `"userName=joao and age=26"` ou

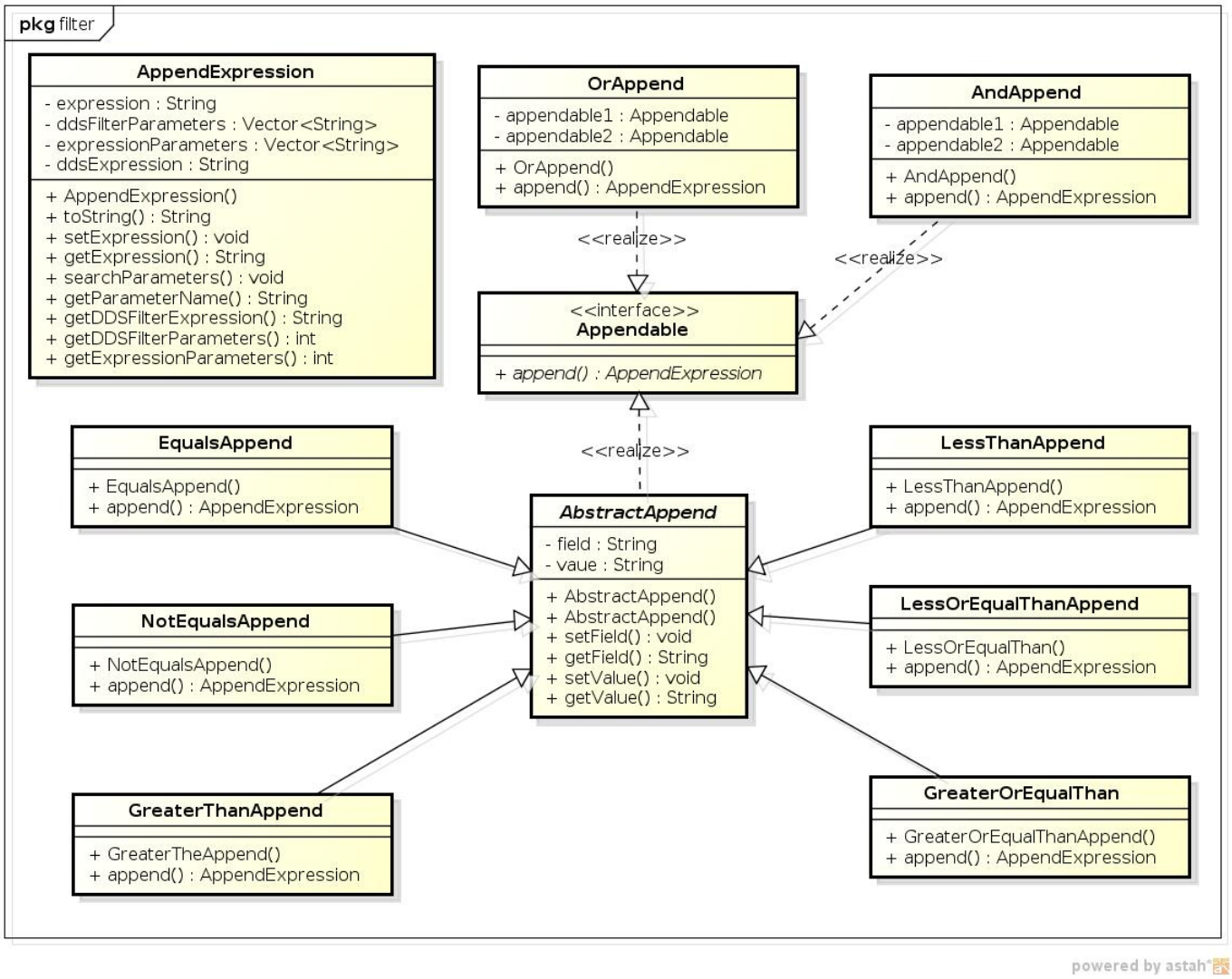

Figura 4.5: Diagrama de Classes Usadas na Geração de Filtros

`"userName=joao or userName=maria"`). A classe `AbstractAppend` também implementa `Appendable` e todas as classes que dela herdam necessitam do par atributo e valor. Isto acontece com as classes `EqualsAppend`, `NotEqualsAppend`, `GreaterThanAppend`, `GreaterOrEqualThanAppend`, `LessThanAppend` e `LessOrEqualThanAppend`.

Como a infraestrutura de comunicação do MobileHealthNet segue uma abordagem *publish/subscribe*, todo protocolo de autenticação é realizado de forma assíncrona. Dessa forma, para possibilitar o recebimento de mensagens é necessário realizar subscrições aos tópicos desejados, incluindo os tópicos específicos de segurança, que são aqueles criados para uso somente pela camada de segurança e destinados à execução do protocolo de autenticação e distribuição de chaves simétricas. As subscrições aos tópicos específicos de

segurança são feitas no ato da criação da instância dos objetos das classes `SecureClientPubSubManager` e `SecureDDSPubSubManager`, através do método privado `subscribeSecureTopics` da classe `SecurePubSubManager`. O Código 4.3 mostra como estas subscrições são implementadas.

Código 4.3: Método para Subscrição nos Tópicos da Segurança

```
1  private void subscribeSecureTopics() {
2        try {
3           SecureTopicPubSubListener secureListener =
4              new SecureTopicPubSubListener();
5           AbstractAppend appendable =
6              new EqualsAppend("uuid", getUuid());
7           AppendExpression expression = appendable.append();
8
9           getPubSubService().subscribe(secureListener,
10                new AuthenticationSuccessTopic(), expression);
11          getPubSubService().subscribe(secureListener,
12                new InvalidCertificateTopic(), expression);
13          getPubSubService().subscribe(secureListener,
14                new AuthenticationFailedTopic(), expression);
15       } catch (DomainParticipantNotCreatedException e) {
16          ...
17       } catch (TopicNotRegisteredException e) {
18          ...
19       }
20    }
```

Como ilustrado no Código 4.3, todos os tópicos dos quais se deseja receber publicações são subscritos através do método `subscribeSecureTopics`, como é o caso dos tópicos `AuthenticationSuccessTopic`, `AuthenticationFailedTopic` e `InvalidCertificateTopic`. Nas linhas 5 e 6 temos que a expressão gerada pela classe `EqualsAppend` representa uma igualdade cujo atributo do tópico é "*uuid*" e o seu valor de comparação será retornado pelo método `getUuid()`. Nesse contexto, o "*uuid*" é um identificador único para todos as instâncias da classe `SecurePubSubManager`. Desta forma, cada nó, seja ele

um dispositivo móvel ou um nó que acessa diretamente o domínio, possui um "$uuid$" único. Portanto, de acordo com o código, seu *listener* será notificado apenas se o tópico publicado pelo KDC contiver o seu próprio "$uuid$". Esse mecanismo é usado para identificar o destinatário da publicação, quando o KDC publica o tópico, ele atribuir ao "$uuid$" do tópico, o identificador referente ao cliente desejado.

Quando o *middleware* recebe um tópico, ele notifica o subscritor através dos *listeners*. O `SecureTopicPubSubListener` é o *listener* utilizado para tratamento dos tópicos específicos da segurança. Como o mesmo *listener* é utilizado na subscrição dos três tópicos (`AuthenticationSuccessTopic`, `AuthenticationFailedTopic` e `InvalidCertificateTopic`), a cada notificação de um tópico de segurança, verifica-se a instância do objeto para saber de qual tópico se trata essa notificação, para então executar o processo subsequente do protocolo.

Segundo o protocolo de autenticação e distribuição de chaves, a aplicação cliente inicia o protocolo toda vez que necessitar cifrar ou decifrar um tópico e a mesma não contiver a chave necessária ou a chave já tenha expirado. Esse protocolo se inicia com a publicação do tópico `SessionKeyRequestTopic`. Durante a execução do protocolo, o tópico publicado pela aplicação fica armazenado em um *buffer* local, até que a execução do protocolo seja concluída. Após a execução do protocolo, todos os tópicos armazenados no *buffer* são criptografados e publicados no domínio. Caso algum erro ocorra durante a execução do protocolo, a aplicação é notificada através do *listener* (`SystemExceptionListener`) informado pela própria aplicação.

Um segundo cenário ocorre quando o subscritor não possui a chave simétrica correspondente ao tópico. Neste caso, os tópicos recebidos são armazenados localmente até que o protocolo seja concluído. De posse da chave simétrica, os tópicos são decifrados e repassados para aplicação através do *listener* informado pela aplicação no ato da subscrição.

4.3.1 Criptografia dos Dados

Como visto na Seção 4.2.3, parte das mensagens enviadas no protocolo são criptografadas utilizando algoritmos assimétricos, até que o cliente obtenha a chave simétrica necessária à publicação ou subscrição ao tópico desejado. Para tanto, foi utilizado o algoritmo de criptografia assimétrica RSA. O AES

é o algoritmo de criptografia simétrica escolhido. Estes algoritmos apresentam melhores desemprenho em ambientes móveis, segundo a avaliação de desempenho disponível em [24]. Para criação de todos os mecanismos de criptografia foi utilizada a API *Java Cryptography Architecture* (JCA) padrão. A JCA é uma especificação de API Java destinada a criptografia de dados e gerenciamento de certificados digitais. A implementação destes mecanismos foram incluídas na classe `AsymmetricEncrypter`, para criptografia assimétrica e `SymmetricEncrypter`, para criptografia simétrica, conforme a Figura 4.6.

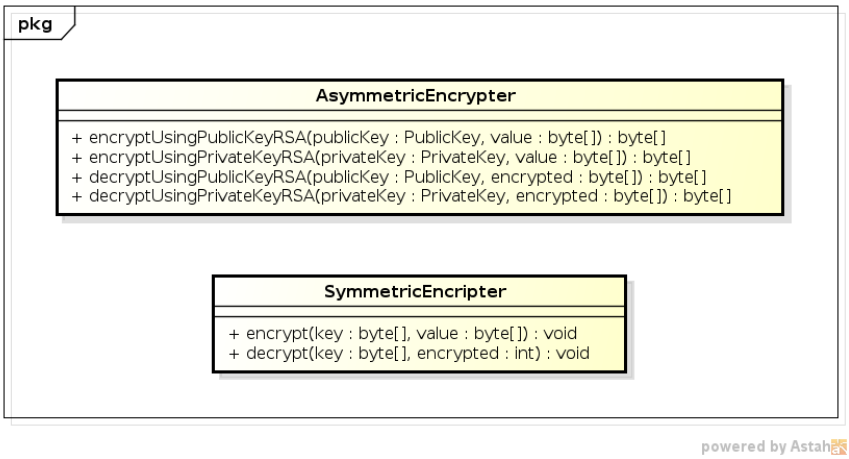

Figura 4.6: Classes Usadas na Criptografia de Dados

Na Figura 4.6, são mostrados os métodos destinados à criptografia dos dados. Para criptografia assimétrica, são disponibilizados dois métodos para cada processo de encriptar e decriptar dados, dependendo da chave utilizada, publica ou privada. No caso da criptografia simétrica, como a chave de criptografia usada é a mesma, é disponibilizado apenas um método para cada processo. O parâmetro `value` corresponde ao conteúdo a ser criptografado. Este conteúdo deve ser serializado e passado em um vetor de bytes.

4.3.2 Certificação Digital

Uma das características desta proposta é o uso de certificados digitais para autenticação dos componentes de software envolvidos. Na versão atual, todos os certificados são gerados utilizando-se a ferramenta OpenSSL. A validação do certificado é realizada pela AC. A AC não armazena estado, seu papel consiste apenas em validar os certificados e gerar os tickets correspondentes a cada certificado. Este

componente se subscreve a apenas um tópico, o `ValidateCertificateTopic`, que representa uma requisição de validação do certificado.

Todas as operações sobre os certificados são definidas através da interface `CertificateManager`, conforme o Código 4.4. A implementação da interface depende do padrão de certificado utilizado. O mais comum e também utilizado como padrão para os certificados do MobileHealthNet é o "X509". Dessa forma, a classe `X509CertificateManager`, que implementa a interface `CertificateManager`, disponibiliza todas as operações destinadas à manipulação deste padrão.

Código 4.4: Interface para Operações em Certificados Digitais

```
1  public interface CertificateManager
2  {
3     public boolean validate(Certificate certificate,
4        Certificate trustedCACertificate);
5
6     public Certificate importCertificateFromFile(
7        String filePath);
8
9     public Certificate importCertificateFromFile(
10       String filePath,
11       String provider);
12
13    public Certificate restoreCertificate(
14       byte[] cert);
15
16    public PrivateKey importPrivateKeyFromFile(
17       String privateKeyPath);
18 }
```

O método `validate(Certificate certificate, Certificate trustedCACertificate)` é responsável pela validação do certificado. Nesse método, o primeiro parâmetro corresponde ao certificado a ser validado, e o segundo corresponde ao certificado da autoridade certificadora confiável que o assinou digitalmente, o qual é usado no processo de validação.

O método `importCertificateFromFile` gera uma instância do objeto `Certificate` a partir do arquivo passado como parâmetro. Opcionalmente, pode-se informar um provedor diferente do padrão, caso a aplicação assim deseje. Neste contexto, um provedor é uma abstração de um objeto usado para manipulação dos mecanismos de baixo nível da criptografia, como algoritmos de criptografia, chaves criptográficas, entre outros. A especificação Java já possui um provedor padrão, no entanto existem outros provedores que podem ser utilizados, como o provedor disponibilizado pela implementação da API JCA desenvolvida pelo grupo "Bouncy Castle"[3].

O certificado digital pode ser apresentado tanto como um arquivo quanto como um objeto serializado. O método `restoreCertificate` é usado para gerar uma instância do objeto `Certificate` a partir do objeto serializado passado como parâmetro, através de um array de bytes. Esta forma é usada principalmente quando os certificados são transportados na rede, como ocorre quando os certificados são transportados dos clientes para o KDC e do KDC para a AC.

4.4 Conclusão

A segurança da comunicação em ambientes de rede sem fio é algo essencial, pois estes ambientes são bastantes sujeitos a interceptações. Para tanto, são criados canais seguros de comunicação, que envolvem o uso de criptografia de dados para prover as propriedades de autenticidade, integridade e confidencialidade das mensagens. O DDS não possui nenhum protocolo oficial destinado à segurança da comunicação.

Este trabalho descreve uma solução para construção de um canal seguro de comunicação para infraestruturas que utilizem a especificação DDS. A referida proposta tem como características principais ser escalável com relação à quantidade de nós envolvidos na comunicação, a utilização de certificados digitais em seu mecanismo de autenticação, e a preservação de características do DDS, como os filtros, a possibilidade de uso de *multicast*, e do paradigma *publish/subscribe*.

[3] http://www.bouncycastle.org/java.html

Podemos destacar como limitação desta proposta a necessidade do desenvolvedor, ao definir seus tópicos, incluir obrigatoriamente um atributo booleano aos mesmos (denominado "`secure`"). Esse atributo é usado para identificar se uma dada instância do tópico foi publicada de forma segura ou não. Essa limitação surge do fato de levarmos em consideração a impossibilidade de modificarmos o código do *middleware* DDS, no caso de se optar pelo uso de uma implementação que não seja de código aberto. O uso do atributo booleano poderia ser evitado alterando-se a sintaxe da linguagem *Data Definition Language* (DDL), incluindo-se anotações utilizadas para especificar quais atributos dos tópicos seriam publicados de forma segura.

5 Avaliação da Infraestrutura de Comunicação Segura do MobileHealthNet

A inclusão de mecanismos de segurança em um sistema computacional gera um custo em seu desempenho. A preocupação com este custo é ainda maior quando se trata de dispositivos móveis, que possuem baixo poder de processamento e pouca memória. Dessa forma, faz-se necessário que o custo gerado pelos mecanismos de segurança seja o menor possível, de modo que não comprometa o desempenho do sistema ao ponto de tornar inviável seu uso. Diante disso, após o desenvolvimento da infraestrutura de segurança para a comunicação no MobileHealthNet, foram realizados diversos experimentos com o intuito de validar o impacto causado pela mesma no ambiente. Esses experimentos são descritos neste capitulo.

5.1 Objetivos dos Experimentos

O objetivo dos experimentos realizados é avaliar qual o impacto que os custos relacionados à infraestrutura de segurança geraram sobre a escalabilidade do sistema, uma vez que o MobileHealthNet foi concebido para poder ser utilizado por milhares de usuários móveis. Para tanto, foram realizados diversos experimentos com e sem os mecanismos de segurança, a fim de se comparar o desempenho do sistema com e sem a utilização dos mesmos. Nos experimentos realizados, analisa-se o desempenho da infraestrutura quando milhares de usuários móveis interagem entre si. Nesse contexto, são avaliadas métricas importantes para a qualidade do serviço da infraestrutura de comunicação, que devem ser mantidas quando adicionados os mecanismos de segurança.

5.2 Carga de Trabalho e Métricas

Como carga de trabalho foram escolhidos os seguinte itens:

1. Número de publicadores: número de clientes móveis que publicam informações de contexto;

2. Número de subscritores: número de clientes móveis que receberam informações de contexto;

Para avaliar o desempenho da infraestrutura de comunicação no decorrer dos experimentos, as métricas escolhidas foram:

- *Throughput* dos publicadores (XP): a média do número total de mensagens por segundo que todos os publicadores conseguiram enviar. Esta métrica é obtida através da soma do número de publicações dividida pelo total de tempo decorrido da simulação, expresso em segundos.

- *Throughput* dos subscritores (XS): a média do número total de mensagens por segundo que todos os subscritores conseguiram receber. Métrica obtida através da soma do número de mensagens recebidas dividida pelo total de tempo decorrido, em segundos.

- *Round Trip Time* (RTT): a média do tempo (em milissegundos) que uma mensagem leva para ir (ser publicada) e voltar (ser recebida pelo subscritor). Métrica obtida calculando-se o RTT médio de todas as mensagens, ou seja, é a soma de todos os RTTs dividido pelo número de mensagens recebidas.

- Taxa de Perdas de Mensagens: a porcentagem de todas as mensagens que foram enviadas, mas não recebidas. Métrica obtida pelo cálculo da porcentagem de todas as mensagens que foram recebidas, a partir das mensagens que foram enviadas. É a soma de todas as mensagens recebidas dividido pela soma das mensagens que foram enviadas, multiplicado por cem.

As métricas *throughput* dos publicadores e dos subscritores foram escolhidas para se verificar o quanto o número de publicações e subscrições pode afetar o desempenho do sistema, dado que o número de mensagens aumenta em função do número de publicadores e subscritores, aumentando também o tempo gasto com o processo de criptografia das mensagens, podendo diminuir a quantidade de mensagens publicadas ou recebidas em um determinado tempo.

O RTT foi escolhida para obter qual a média de tempo que se demora para receber uma mensagem enviada de acordo com o número de publicadores e subscritores na rede. Semelhante ao *throughput*, o RTT também cresce em função do número de publicadores e subscritores. Além disso, dependendo do tempo adicional gerado pelos mecanismos de segurança, o RTT poderia vir a aumentar bastante, depreciando assim a infraestrutura de comunicação.

Por fim, a porcentagem de mensagens perdidas serve para medir qual a porcentagem de mensagens que são enviadas, mas não chegam a seu destino. Esta métrica permite avaliar qual a influência da sobrecarga gerada pelo aumento da quantidade de mensagens publicadas em virtude do número de clientes (publicadores e subscritores) conectados no ambiente.

5.3 Descrição dos Experimentos

Para realização dos experimentos, foi utilizado o serviço de contexto, onde foram simulados vários clientes móveis, publicadores e subscritores. A simulação consistiu de uma aplicação que instancia várias *threads*, onde cada *thread* corresponde a um cliente móvel. Cada publicador disponibilizava sua localização, ou seja, através do tópico **ContextInformation**, sua latitude e longitude eram distribuídas para todos aqueles interessados, duas vezes por minuto. Os subscritores, por sua vez, recebiam estas informações.

Com o objetivo de realizar os experimentos em um ambiente controlado, estes foram realizados em dois ambientes de rede sem fio (Wireless 802.11) e os clientes foram distribuídos em quatro maquinas, duas para cada ambiente. O motivo da distribuição dos clientes móveis em máquinas e ambientes WiFi distintos foi para evitar que fatores como a sobrecarga de processamento e memória nas máquinas responsáveis pela emulação dos clientes ou a sobrecarga do roteador sem fio pudesse vir a interferir nos resultados dos experimentos, dado que o objetivo principal era avaliar o impacto dos mecanismos de segurança considerando as métricas pré-estabelecidas.

Os experimentos foram divididos em quatro cenários, sendo que cada um deles foi executado cinco vezes, com e sem os mecanismos de segurança.

Experimento	Nº de Publicadores	Nº de Subscritores	Nº de Publicações
E1	250	750	10000
E2	600	1800	24000
E3	800	2400	32000
E4	1000	3000	40000

Tabela 5.1: Tabela de Experimentos

Nos experimentos foram adotadas uma proporção de três subscritores para cada publicador. Todos os experimentos tinham duração de 23 minutos, onde as publicações ocorriam nos primeiros 20 minutos e cada publicador realizava duas publicações por minuto. Os três minutos restantes foram adicionados para que eventuais mensagens que ainda não haviam sido entregues pudessem ser processadas. A Tabela 5.1 apresenta os parâmetros utilizados nos experimentos.

No caso dos experimentos com segurança, todo o processo de autenticação e distribuição de chaves simétricas descrito na Seção 4.2.3 é realizado no inicio do experimento. Dessa forma, quando os publicadores iniciam suas publicações, os mesmos já possuem a chave simétrica usada para criptografar o tópico, bem como os subscritores para decifrar o tópico.

5.4 Análise dos Resultados

Analisando a Figura 5.1, podemos observar que o *throughput* das mensagens não varia muito com inclusão dos mecanismos de segurança. No caso do *throughput* dos publicadores (XP), podemos ver que são iguais, isto porque a aplicação é a mesma e publica a mesma quantidade de tópicos no mesmo intervalo de tempo. A criptografia dos tópicos ocorre após a publicação do tópico pela aplicação, por esse motivo o processo de criptografia não influência no XP. Por outro lado, é natural que haja uma variação no *throughput* dos subscritores (XS), conforme observado, uma vez que o processo de criptografia acrescenta um certo esforço computacional para cifrar e decifrar o tópico, diminuindo a quantidade de mensagens recebidas naquele intervalo de tempo.

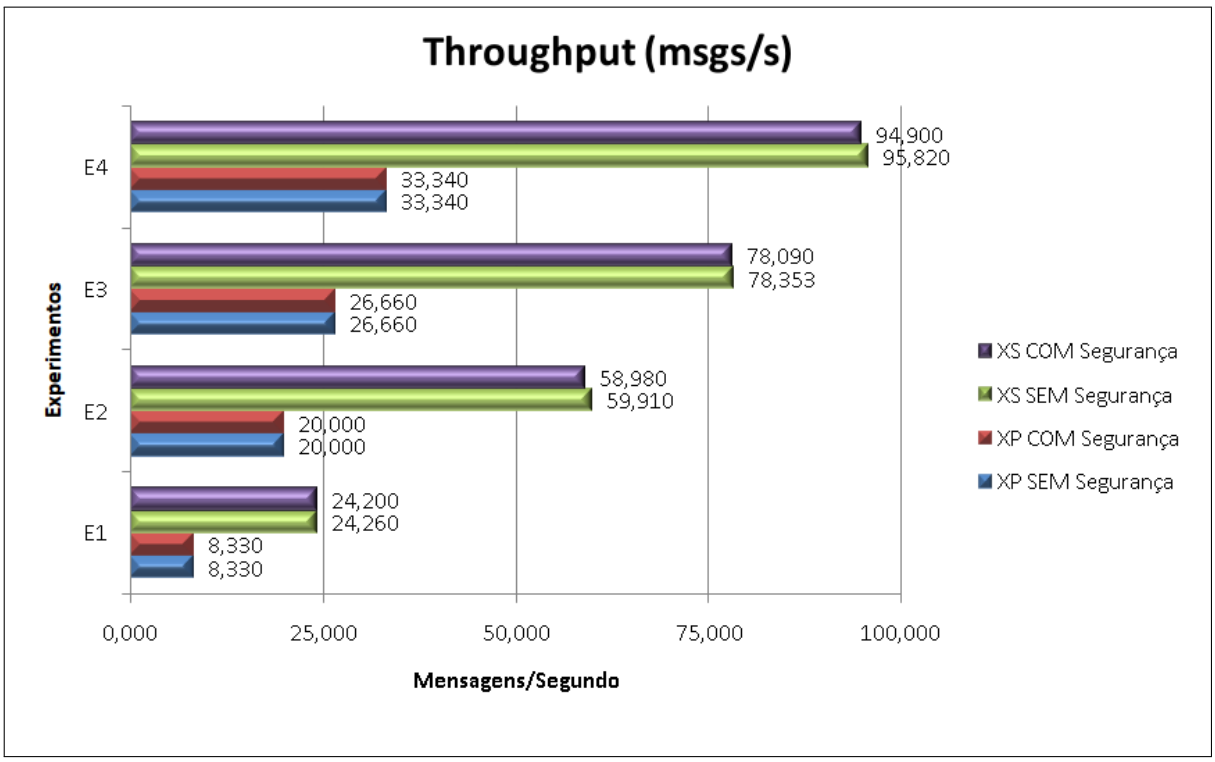

Figura 5.1: Resultados de Throughputs

Através da análise do *throughput*, podemos observar que o mesmo cresce bastante de um experimento para outro, visto que a quantidade de mensagens publicadas no domínio aumenta significativamente, aumentando também a quantidade de mensagens recebidas no mesmo intervalo de tempo. Mesmo assim, a maior diferença obtida em relação ao experimento sem segurança foi de $0,93$ mensagens por segundo no experimento $E2$, um bom resultado se analisarmos em função dos benefícios do uso de uma comunicação segura.

Analisando o RTT, mostrado na Figura 5.2, podemos observar um impacto mais significativo dos mecanismos de segurança. O RTT corresponde ao tempo médio que uma mensagem leva para chegar ao seu destino e retornar ao emissor. Com o uso dos mecanismos de segurança na comunicação, os interlocutores realizam dois procedimentos adicionais que correspondem a criptografia nos dois pontos finais, o que requer um tempo adicional se comparado com o ambiente sem segurança. Portanto, é esperado que o ambiente com segurança apresente valores de RTT maiores que ambiente sem segurança. Observando os experimentos, percebemos que a média do RTT obtido foi menor do que um segundo. Para RSMs, que almejam baixos tempos de resposta, cujas interações ocorrem quase que de maneira instantânea, o RTT obtido nesses experimentos, mesmo com os mecanismos de segurança, torna-

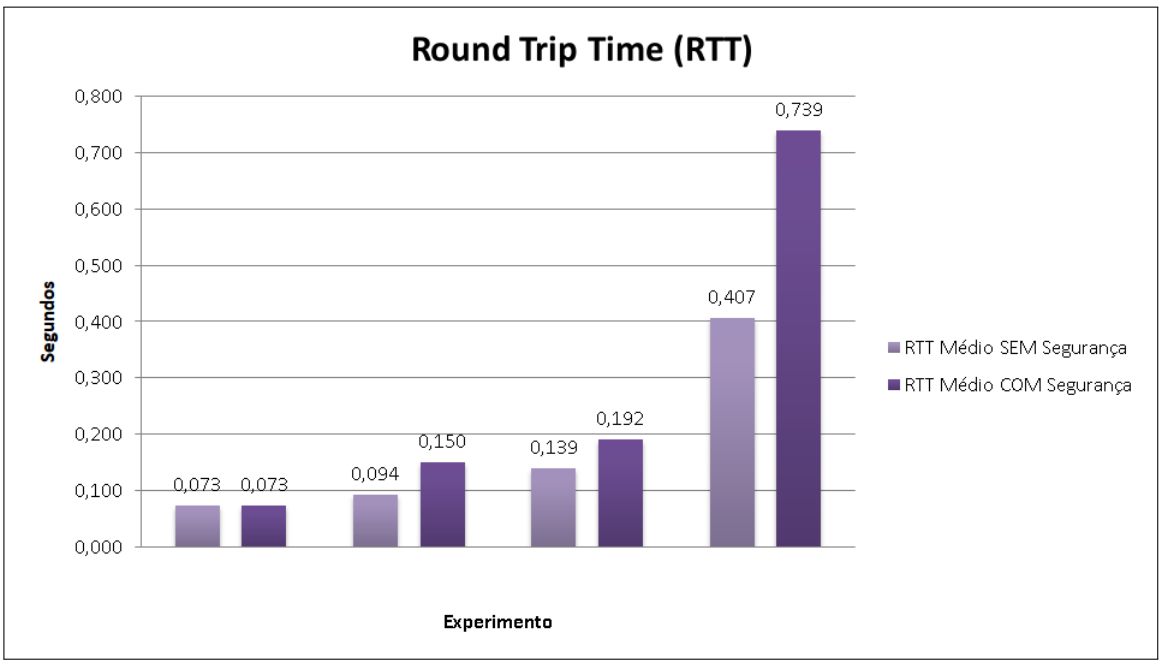

Figura 5.2: Resultados do RTT

se suficiente para comunicação entre os usuários de uma RSM criada por meio desta infraestrutura, possibilitando interações sem grandes tempos de espera (por exemplo 10, 30, 60 segundos ou mais).

Por fim, a Figura 5.3 mostra a Taxa de Perdas de Mensagens obtida nos experimentos, que é uma importante métrica, pois permite medir quantitativamente as mensagens que são enviadas e não chegam ao seu destino. Na maioria dos experimentos mais de 99% das mensagens são entregues, inclusive com a segurança. Nos três primeiros experimentos foi observada uma diferença pequena em relação a perda de mensagens entre os experimentos com e sem segurança. Contudo, uma maior diferença pode ser observada no experimento $E4$. Após diversas análises, observamos que a perda de mensagem ocorre nos dois ambientes de comunicação base, tanto no MRUDP quanto no próprio CoreDx. Foi observado que a grande maioria das mensagens perdidas eram do CoreDx, contudo os testes realizados não foram suficientes para identificar de forma precisa a causa das perdas. Os testes confirmam que a grande quantidade de clientes publicando mensagens ao mesmo tempo acaba sobrecarregando o domínio, gerando a perda dessas mensagens.

A infraestrutura de comunicação do MobileHealthNet, apesar de prover entrega confiável de mensagens, somente pode provê-las se o ambiente móvel fornecer as condições necessárias para que isso ocorra. Contudo, a confiabilidade de entrega

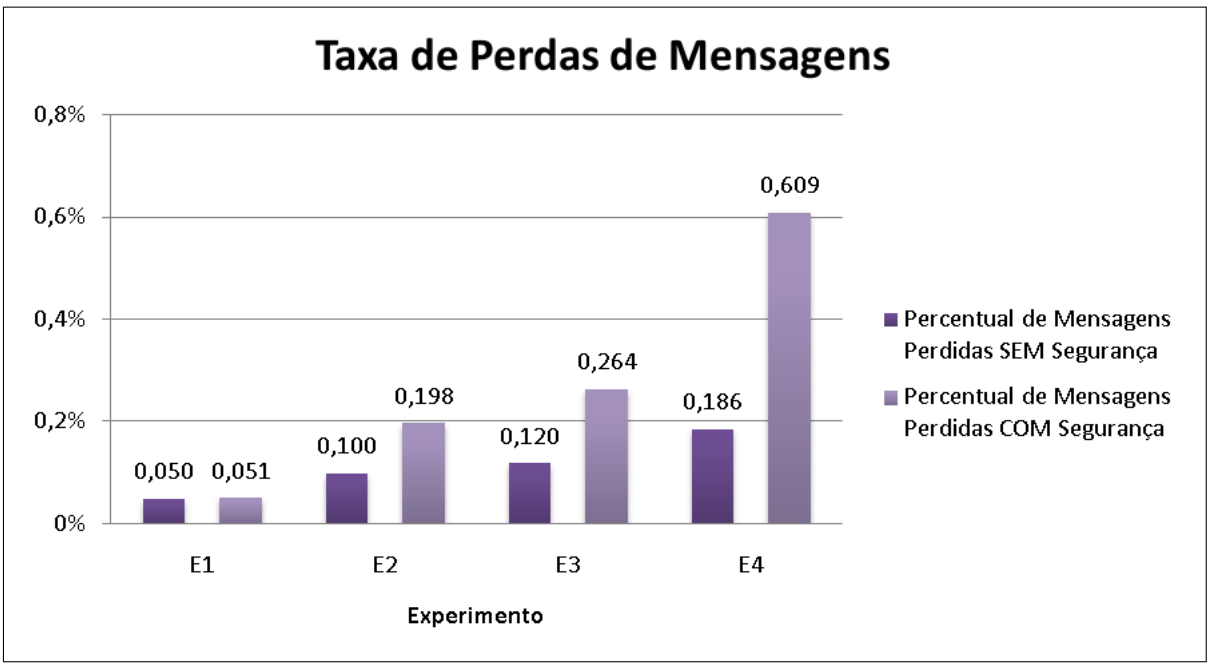

Figura 5.3: Taxa de Perdas de Mensagens

no MobileHealthNet garante que, se for possível, a mensagem será entregue, caso contrário a aplicação será notificada desta impossibilidade, dando a aplicação a possibilidade de realizar qualquer tratamento da mensagem não entregue.

Nos experimentos realizados utilizando-se a infraestrutura de comunicação segura desenvolvida, além do valor médio foram observados os valores mínimo e máximo e calculado o desvio padrão e intervalo de confiança com coeficiente de 95% relativos ao RTT e taxa de perdas de mensagens. De uma forma geral, o desvio padrão das médias obtidas em cada experimento foi baixo. O maior desvio padrão das média encontradas para o RTT ocorreu no experimento $E4$ e foi de $0,167$, para uma média de $0,739$, conforme observado na Figura 5.2. No caso da taxa de perdas de mensagens, o valor de desvio padrão foi de $0,077$. O baixo desvio padrão para as médias reforçam que os valores obtidos em cada experimento estão próximos, o que reafirma que a solução é estável.

Por fim, adotando-se um coeficiente de confiança de 95%, foram calculadas as margens de erro das médias de cada métrica analisada. Para os valores relativos ao RTT, obteve-se uma margem de erro de $0,147$, ficando o intervalo de confiança entre $0,591$ e $0,886$ segundos. Com relação às médias da taxa de perda de mensagens, a margem de erro encontrada foi de $0,067$, obtendo-se um intervalo de confiança variando entre $0,541\%$ e $0,686\%$. Pode-se perceber que os intervalos de confiança

calculados são pequenos e os resultados encontrados tanto para o RTT quanto para a taxa de perda de mensagem refletem que a solução de segurança para proposta neste trabalho possui um baixo impacto no desempenho da infraestrutura de comunicação do MobileHealthNet.

5.5 Conclusão

A análise dos dados estatísticos relativos aos experimentos realizados (média, mínimo, máximo, desvio padrão e intervalo de confiança) nos mostram que a inclusão dos mecanismos de segurança na infraestrutura de comunicação gera uma certa perda no desempenho do sistema que, no entanto, não representa prejuízo significativo ao desempenho da infraestrutura de comunicação, mesmo quando uma grande quantidade de nós móveis concorrentemente utilizam o sistema. Nesse contexto, dizemos que um prejuízo significativo é aquele cujos resultados inviabilize o uso da infraestrutura, nesse caso *throughputs* muito baixos, RTT e Taxa de Perda de Mensagens muito altas.

Além disso, a inclusão destes mecanismos trazem diversos benefícios indispensáveis para ambientes de redes sociais móveis voltados para saúde e consequentemente para o MobileHealthNet, que é a comunicação segura entre as aplicações envolvidas na rede, sendo também uma exigência do MC-SRES.

6 Conclusão e Trabalhos Futuros

Redes Sociais Móveis compreendem um tipo específico de mídia social cuja principal característica é o uso de tecnologias de comunicação sem fio para a interação entre seus membros, agregando capacidade de acesso em qualquer hora e em qualquer lugar aos recursos compartilhados, integrando também diversos tipos de informações contextuais como a localização dos dispositivos. No domínio da saúde existe uma gama de aplicações que podem utilizar RSMs para promover o intercâmbio de informações, colaboração e integração social entre os diversos agentes envolvidos no processo de atendimento à saúde.

O MobileHealthNet é um projeto desenvolvido pelo Laboratório de Sistemas Distribuídos da Universidade Federal do Maranhão e o *Laboratory for Advanced Collaboration* da Pontifícia Universidade Católica do Rio de Janeiro e tem por objetivo desenvolver um *middleware* que permita o acesso às redes sociais e facilite o desenvolvimento de serviços colaborativos para o setor da saúde, a troca de experiências e a comunicação entre pacientes e profissionais da saúde, além de uma melhor gestão dos recursos da saúde por órgãos governamentais.

Para desenvolver uma RSM segura, em especial no domínio da saúde, é necessário que sejam observados diversos requisitos éticos e legais vigentes, bem como atender a especificação descrita no MC-SRES, o que requer um alto nível de exigência para estes sistemas. Para tanto, existe a necessidade de se prover mecanismos de segurança que abrangem todas as camadas do software, desde a camada de aplicações até a camada de comunicação. Neste trabalho foi proposto um modelo de segurança com o objetivo de suprir estas exigência de segurança no contexto do *middleware* MobileHealthNet. No entanto, o modelo proposto é suficientemente genérico para que possa ser estendido e implementado em outros *middlewares* destinados ao desenvolvimento de RSMs voltadas para saúde. Além disso, o desenvolvimento destes mecanismos estão sendo realizados tendo como base um processo de desenvolvimento de software seguro, o CLASP. Todas as atividades do CLASP asseguram que os requisitos de segurança tenham a devida atenção durante o processo de desenvolvimento do *middleware*.

Uma outra contribuição deste trabalho foi o projeto e implementação de mecanismos de segurança para a infraestrutura de comunicação do projeto MobileHealthNet, que é baseada na especificação OMG DDS. A comunicação utilizando o DDS possui diversas peculiaridades, como o fato de ser baseada no paradigma *publish/subscribe* utilizando um modelo baseado em tópicos (*Data Centric Publish/Subscribe*), ser totalmente distribuída, ser frequentemente configurada para uso extensivo de *multcast* e disponibilizar um mecanismo de filtragem de dados baseado no conteúdo publicado através de tópicos. Este trabalho apresenta uma solução de comunicação segura e escalável tendo como base o DDS que atende aos requisitos de segurança definidos no âmbito do projeto MobileHealthNet.

Finalmente, uma avaliação de desempenho preliminar da infraestrutura de segurança para a comunicação proposta neste trabalho foi realizada por meio de vários experimentos. Os resultados obtidos nos permitem concluir que os mecanismos de segurança possuem um impacto relativamente baixo para infraestrutura de comunicação, pois não prejudicam significativamente o desempenho e escalabilidade do ambiente, trazendo diversos benefícios para o ambiente através da comunicação segura.

6.1 Contribuições Cientificas

Este trabalho teve como produto as seguintes contribuições cientificas:

- Levantamento dos mecanismos de privacidade e segurança implementados nos principais middleware para RSMs constantes da literatura;
- Levantamento e análise dos principais requisitos de segurança para RSMs voltadas para a área da saúde;
- Definição de um modelo de segurança para RSMs voltadas para saúde, tendo como base o conjunto de requisitos levantados;
- Projeto, implementação e avaliação de um mecanismo de segurança para o estabelecimento de canais seguro de comunicação para uma infraestrutura *publish/subscribe* que tem como base o DDS.

No âmbito deste trabalho foram obtidas as seguintes publicações:

- GONÇALVES, J. F. ; SILVA, F. J. S. E. ; VASCONCELOS, R.; BATISTA, G; ENDLER, M. A Security Infrastructure for Massive Mobile Data Distribution. Publicado em 9th ACM International Symposium on QoS and Security for Wireless and Mobile Networks - Q2SWinet 2013;
- GONÇALVES, J. F. ; TELES, A. S. ; SILVA, F. J. S. E. . Um Modelo de Segurança e Privacidade para Redes Sociais Móveis Aplicadas à Área da Saúde. Publicado no Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais - SBSeg 2012;
- TELES, A. S. ; GONÇALVES, J. F. ; ALMEIDA, V. P. ; SILVA, F. J. S. E. ; ENDLER, M. Infraestrutura e Aplicações de Redes Sociais Móveis para Colaboração em Saúde. Publicado no XIII Congresso Brasileiro de Informática em Saúde - CBIS 2012;
- TELES, A. S. ; GONÇALVES, J. F. ; SILVA, F. J. S. E. ; BATISTA, R. C. ; PINHEIRO, D. ; ALMEIDA, V. P. ; ENDLER, M. MobileHealthNet: A Middleware for Mobile Social Networks in m-Health. Publicado no 3rd International Conference on Wireless Mobile Communication and Healthcare, 2012;
- Minicurso: Redes sociais móveis: conceitos, aplicações e aspectos de segurança e privacidade. Apresentado e publicado no 31° Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2013).

6.2 Trabalhos Futuros

A partir deste trabalho inicial identificamos alguns trabalhos que podem ser futuramente desenvolvidos dando continuidade ao mesmo:

- Desenvolver os demais mecanismos de segurança propostos no modelo no *middleware* MobileHealthNet;
- Implementar mecanismos de balanceamento de cargas do gerenciamento de chaves entre diferentes instâncias do KDC. As diferentes instâncias necessitam entrar em acordo sobre quais tópicos são gerenciados por cada uma delas;

- Realizar mais avaliações dos mecanismos de segurança levando em consideração o custo computacional gerado pela segurança aos dispositivos móveis.

Referências Bibliográficas

[1] Simple Object Access Protocol (SOAP) 1.1, W3C Note. urlhttp://www.w3.org/TR/2000/NOTE-SOAP-20000508/, 2000.

[2] F. A. and P. Karlton. The Secure Sockets Layer (ssl) Protocol Version 3.0. RFC 6101 (Proposed Standard), Aug. 2011.

[3] F. Adelstein, S. K. S. Gupta, L. Schwiebert, and G. G. Richard. *Fundamentals of Mobile and Pervasive Computing*. McGraw-Hill Companies, 1st edition, 2005.

[4] J. An, Y. Ko, and D. Lee. A social relation aware routing protocol for mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, PERCOM '09, pages 1–6. IEEE Computer Society, 2009.

[5] D. Anthony, T. Henderson, and D. Kotz. Privacy in location-aware computing environments. *IEEE Pervasive Computing*, 6(4):64–72, Oct. 2007.

[6] R. Batista and F. Silva. Uma infraestrutura de comunicação para colaboração em redes sociais móveis. In *SBSC 2012: Simpósio Brasileiro de Sistemas Colaborativos*, oct 2012.

[7] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The Secure Real-time Transport Protocol (SRTP). RFC 3711 (Proposed Standard), 2004. Updated by RFC 5506.

[8] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, M. Terada, and R. Han. Whozthat? evolving an ecosystem for context-aware mobile social networks. *IEEE Network*, 22(4):50–55, 2008.

[9] A. Beach, M. Gartrell, and R. Han. Solutions to security and privacy issues in mobile social networking. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, pages 1036–1042, Washington, DC, USA, 2009. IEEE Computer Society.

[10] E. G. Boix, A. L. Carreton, C. Scholliers, T. Van Cutsem, W. De Meuter, and T. D'Hondt. Flocks: enabling dynamic group interactions in mobile social networking applications. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 425–432, New York, NY, USA, 2011. ACM.

[11] C. Borcea, A. Gupta, A. Kalra, Q. Jones, and L. Iftode. The mobisoc middleware for mobile social computing: challenges, design, and early experiences. In *Proceedings of the 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, MOBILWARE '08, pages 27:1–27:8, Brussels, Belgium, Belgium, 2007. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering.

[12] D. Bottazzi, R. Montanari, and A. Toninelli. Context-aware middleware for anyti anywhere social networks. *IEEE Intelligent Systems*, 22:23–32, September 2007.

[13] L. David, R. Vasconcelos, L. Alves, R. André, G. Baptista, and M. Endler. A communication middleware for scalable real-time mobile collaboration. *IEEE 21st International WETICE, Track on Adaptive and Reconfigurable Service-oriented and component-based Applications and Architectures (AROSA)*, 2012.

[14] S. B. de Informática em Saúde. Manual de Certificação para Sistemas de Registro Eletrônico em Saúde (S-RES), Dezember 2011.

[15] G. Demiris. The diffusion of virtual communities in health care: concepts and challenges. *Patient Education and Counseling*, 62(2):178–188, Aug. 2006.

[16] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008.

[17] M. Fiorio, C. O. Emmanoel, P. F. Pires, and F. C. Delicato. *Soluções para o desenvolvimento de sistemas seguros*. SBSEG, 2009.

[18] B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea, F. Kaashoek, and R. Morris. Persistent personal names for globally connected mobile devices. In *Proceedings of the 7th symposium on Operating systems design and implementation*, OSDI '06, pages 233–248, Berkeley, CA, USA, 2006. USENIX Association.

[19] E. Freeman and E. Freeman. *Use a Cabeça! Padrões de Projeto*. Alta Books, 2nd edition, 2011.

[20] H. Gao, J. Hu, T. Huang, J. Wang, and Y. Chen. Security issues in online social networks. *Internet Computing, IEEE*, 15(4):56 –63, july-aug. 2011.

[21] J. Goncalves, A. Teles, and F. Silva. Um modelo de segurança e privacidade para redes sociais móveis aplicadas à Área da saúde. In *SBSeg 2012: XII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, oct 2012.

[22] O. M. Group. Data distribution service for real-time systems specification. `http://www.omg.org/spec/DDS/1.2/PDF/`, July 2001.

[23] O. M. Group. Request For Proposal - DDS Secure. DDS Secure RFP, 2007.

[24] A. Gupta, A. Kalra, D. Boston, and C. Borcea. Mobisoc: a middleware for mobile social computing applications. *Mob. Netw. Appl.*, 14:35–52, February 2009.

[25] P. Joshi and C.-C. Kuo. Security and privacy in online social networks: A survey. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1 –6, july 2011.

[26] D. N. Kalofonos, Z. Antoniou, F. D. Reynolds, M. Van-Kleek, J. Strauss, and P. Wisner. Mynet: A platform for secure p2p personal and social networking services. In *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, PerCom '08, pages 135–146, 2008.

[27] A. M. Kaplan and M. Haenlein. Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, 53(1):59–68, Jan. 2010.

[28] A. Karam and N. Mohamed. Middleware for mobile social networks: A survey. *Proceedings of the 45th Hawaii International Conference on System Sciences*, pages 1482–1490, 2012.

[29] N. Kayastha, D. Niyato, P. Wang, and E. Hossain. Applications, architectures, and protocol design issues for mobile social networks: A survey. *Proceedings of the IEEE*, 99(12):2130–2158, 2011.

[30] S. Kern, P. Braun, and W. Rossak. Mobisoft: an agent-based middleware for social-mobile applications. In *Lecture Notes in Computer Science including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics – Proceedings of the International Conference On the Move to Meaningful Internet Systems*, OTM'06, pages 984–993, Berlin, Heidelberg, 2006. Springer-Verlag.

[31] J. M. Leimeister, M. Daum, and H. Krcmar. Mobile virtual healthcare communities: An approach to community engineering for cancer patients. In *in European Conference on Information Systems (ECIS*, pages 1626–1637, 2002.

[32] J. Li and Q. Li. Decentralized self-management of trust for mobile ad hoc social networks. *International Journal of Computer Networks & Communications (IJCNC)*, 3(6):1–17.

[33] R. Lübke. *Ein Framework zur Entwicklung mobiler Social Software auf Basis von Android*. PhD thesis, Dresden, Germany, March 2011.

[34] R. Lubke, D. Schuster, and A. Schill. Mobilisgroups: Location-based group formation in mobile social networks. In *Proceedings of the 9th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2011, 21-25 March 2011, Seattle, WA, USA, Workshop Proceedings*, pages 502–507. IEEE, 2011.

[35] J. McHugh. Low bandwidth soap. `http://www.xml.com/pub/a/ws/2003/08/19/ksoap.html`, Aug. 2003.

[36] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 337–350. ACM Press, 2008.

[37] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot. Mobiclique: middleware for mobile social networking. In *Proceedings of the 2nd ACM workshop on Online social networks*, WOSN '09, pages 49–54, New York, NY, USA, 2009. ACM.

[38] B. Qureshi, G. Min, and D. Kouvatsos. A framework for building trust based communities in p2p mobile social networks. In *Proceedings of the 10th IEEE International Conference on Computer and Information Technology*, CIT '10, pages 567–574. IEEE Computer Society, 2010.

[39] J. Rana, J. Kristiansson, J. Hallberg, and K. Synnes. Challenges for mobile social networking applications. In *Proceedings of the International ICST Conference*

on Communications Infrastructure, Systems and Applications in Europe, volume 16 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 275–285. Springer Berlin Heidelberg, 2009.

[40] E. Rescorla and N. Modadugu. Datagram Transport Layer Security. RFC 4347 (Proposed Standard), Apr. 2006.

[41] P. Saint-Andre, K. Smith, and R. Tronon. *XMPP: The Definitive Guide Building Real-Time Applications with Jabber Technologies*. O'Reilly Media, Inc., 2009.

[42] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFC 5506.

[43] D. Schuster, I. Koren, T. Springer, D. Hering, B. Söllner, M. Endler, and A. Schill. *Creating Applications for Real-Time Collaboration with XMPP and Android on Mobile Devices*. Handbook of Research on Mobile Software Engineering: Design, Implementation and Emergent Applications, IGI Global, 2012.

[44] I. Secure Software. The CLASP Application Security Process. Technical report, Secure Software, Inc, 2005.

[45] B. Sollner. *XMPP-based Media Sharing for Mobile Collaboration with Android Phones*. PhD thesis, Technische Universitat Dresden, Germany, Octuber 2009.

[46] M. Stamp. *Information Security: Principles and Practice*. Wiley InterScience, 2006.

[47] J. Su, J. Scott, P. Hui, J. Crowcroft, E. de Lara, C. Diot, A. Goel, M. Lim, and E. Upton. Haggle: Seamless Networking for Mobile Applications. pages 391–408. 2007.

[48] A. S. Tanenbaum and M. v. Steen. *Distributed Systems: Principles and Paradigms (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[49] A. Teles, F. J. da Silva e Silva, and R. Batista. *Security and Privacy in Mobile Social Networks*. Lecture Notes in Social Networks. Security and Privacy Preserving in Social Networks, Springer, 2013.

[50] A. Teles, J. Goncalves, F. J. Silva, V. Pinheiro, and M. Endler. Infraestrutura e aplicações de redes sociais móveis para colaboração em saúde. In *CBIS 2012: XIII Congresso Brasileiro de Informática na Saúde*, nov 2012.

[51] A. Teles, D. Pinheiro, J. Goncalves, R. Batista, F. J. Silva, V. Pinheiro, E. Haeusler, and M. Endler. Mobilehealthnet: A middleware for mobile social networks in m-health. In *MOBIHEALTH 2012: 3rd International Conference on Wireless Mobile Communication and Healthcare*, 2012.

[52] C. Tong. Analysis of some popular mobile social network systems. Technical report, Helsinki University of Technology, April 2008.

[53] J. R. Vacca. *Computer and Information Security Handbook*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009.

[54] U. Varshney. *Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[55] D. Zhang, Z. Wang, B. Guo, X. Zhou, and V. Raychoudhury. A dynamic community creation mechanism in opportunistic mobile social networks. In *Proceedings of the IEEE 3rd International Conference on Social Computing*, SocialCom/PASSAT '11, pages 509–514, 2011.

[56] W. Zhenyu, Z. Chunhong, J. Yang, and W. Hao. Towards cloud and terminal collaborative mobile social network service. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pages 623–629, Washington, DC, USA, 2010. IEEE Computer Society.