UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ESTUDO E IMPLEMENTAÇÃO DE AUTENTICAÇÃO NO ACESSO PARA O PORTAL DO HUSM

TRABALHO DE GRADUAÇÃO

Eduardo Maikel Müller

Santa Maria, RS, Brasil

2007

ESTUDO E IMPLEMENTAÇÃO DE AUTENTICAÇÃO NO ACESSO PARA O PORTAL DO HUSM

por

Eduardo Maikel Müller

Trabalho de Graduação apresentado ao Curso de Ciência da Computação
da Universidade Federal de Santa Maria (UFSM, RS), como requisito
parcial para a obtenção do grau de
Bacharel em Ciência da Computação

Orientador: Prof. Dr. Raul Ceretta Nunes

Trabalho de Graduação Nº 219
Santa Maria, RS, Brasil

2007

Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

ESTUDO E IMPLEMENTAÇÃO DE AUTENTICAÇÃO NO ACESSO PARA O PORTAL DO HUSM

elaborado por
Eduardo Maikel Müller

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA:

Prof. Dr. Raul Ceretta Nunes
(Presidente/Orientador)

Prof. Antônio Marcos de Oliveira Candia (UFSM)

Esp. Marcos Vinícius Bitencourt de Souza (UFSM)

Santa Maria, 01 de março de 2007.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por ter me dado a oportunidade de estar no mundo.

Aos meus pais, Augusto Gilseu Müller e Áurea Hübner Müller, agradeço todo o amor e apoio, sem eles nada disso seria possível, meu irmão Rafael Roger Müller e ao restante da minha família.

Gostaria de agradecer também o professor Raul Ceretta Nunes, pela atenção e auxílio às atividades e discussões sobre o andamento e dúvidas desta monografia de conclusão de curso.

Aos meus amigos, Leandro Sacchet e Marcos Vinícius Bitencourt de Souza, que colaboraram muito no decorrer deste trabalho. Estavam sempre dispostos a ajudar e esclarecer dúvidas, sendo com certeza, muito importantes no desenvolvimento do trabalho.

Por fim, aos meus colegas do curso e também grandes amigos, especialmente ao Guilherme Piêgas Koslovski, Giovani Gracioli, Tiago Nonoai e José Paulo, que torceram por mim e estiveram presentes nesta caminhada.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

ESTUDO E IMPLEMENTAÇÃO DE AUTENTICAÇÃO NO ACESSO PARA O PORTAL DO HUSM
Autor: Eduardo Maikel Müller
Orientador: Prof. Dr. Raul Ceretta Nunes
Local e data da defesa: Santa Maria, 01 de março de 2007.

Neste trabalho é tratada autenticação sob-demanda através da tecnologia *web service*, onde têm-se um serviço responsável por verificar a autenticidade dos usuários. O serviço pode comprovar a autenticação através de três mecanismos: usuário/senha, certificado digital e impressão digital.

Para que o serviço seja seguro são empregados protocolos de segurança na troca de mensagens SOAP (*Simple Object Access Protocol*). Para validar o estudo foi implementado um portal para o Hospital Universitário de Santa Maria (HUSM). O portal contém uma listagem de aplicações, onde para cada uma delas têm-se políticas de acesso específicas. Para cada nível de acesso definido nas políticas, um tipo de tecnologia diferente é necessária para o acesso.

Como resultado, o portal é acessado por diferentes grupos de usuários, cada grupo com políticas que definem o mecanismo de autenticação necessário.

Palavras-chave: Web Services, Autenticação, Portal Web.

ABSTRACT

Undergraduation Final Work
Undergraduation in Computer Science
Federal University of Santa Maria

STUDY AND IMPLEMENTATION ABOUT ACCESS AUTHENTICATION IN THE HUSM WEB PORTAL
Author: Eduardo Maikel Müller
Advisor: Prof. Dr. Raul Ceretta Nunes

In this work is treated on-demand authentication through the web services technology, where they have a responsible service for verifying the authenticity of the users. The service can prove the authentication through three mechanisms: user/password, digital certificate and fingerprint.

To provide a trust service, security protocols on SOAP (Simple Object Access Protocol) messages are used. To validate the study a web portal for the HUSM was implemented. The portal contains a list of applications, where each one has specific access politics. For each level of access specified a type of different technology is necessary for the access.

As result, the portal is accessed by different groups of users, each group with politics that defines the necessary mechanism of authentication.

Keywords: Web Services, Authentication, Web Portal.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AC	Autoridade Certificadora
API	Application Programming Interface
B2C	Business-To-Consumer
B2B	Business-To-Business
DCOM	Distributed COM
DSS	Digital Signature Standard
FBI	Departamento Federal de Investigação
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HUSM	Hospital Universitário de Santa Maria
IETF	Internet Engineering Task Force
IIOP	Internet Inter-ORB Protocol
IPSec	Internet Protocol Security
JEE	Java Enterprise Edition
JMS	Java Message Service
OASIS	Organization for the Advancement of Structured Information Standards
PDP	Policy Decision Point
PIN	Personal Identification Number
PMI	Permissions Management Infrastructure
PKI	Infraestrutura de Chave Pública
RMI	Remote Method Invocation
SAML	Security Assertion Markup Language
SIE	Sistema de Informações para o Ensino
SDK	Kit de Desenvolvimento de Software
SHA-1	Secure Hash Algorithm 1
SMS	Short Message Service

SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SSO	Single Sign-On
SSTC	Security Services Technical Commitee
UDDI	Universal Description, Discovery and Integration
URI	Uniform Resource Identifier
WSDL	Web Services Description Language
WSS4J	Web Services Security for Java
X-KISS	Key Information Service Specification
X-KRSS	Key Registration Service Specification
XACML	eXtensible Access Control Markup Language
XKMS	XML Key Management Specification
XML	eXtensible Markup Language
XrML	eXtensible rights Markup Language

SUMÁRIO

1 INTRODUÇÃO

Com o avanço da tecnologia, cada vez mais criaram-se aplicações de informática na área de saúde. Como conseqüência, sistemas de informação hospitalar contendo dados clínicos são uma vertente inevitável (COSTA, 2001). Entretanto, o HUSM ainda encontra-se num estado embrionário em relação ao assunto, principalmente no que diz respeito a disponibilização de informações e aplicações na Internet.

O acesso *web* a informações clínicas de pacientes é matéria regulada pelo governo e deve garantir a confidencialidade e integridade dos dados. Dessa forma ter feito um Portal *Web* para o HUSM, onde o sistema clínico pode ser acessado com um navegador *web* através da Internet, foi um desafio.

Um dos primeiros pontos resolvidos foi evitar o acesso por usuários não autenticados/autorizados, sendo necessário empregar alguns mecanismos de segurança. Este trabalho visa a exploração de mecanismo de segurança com enfoque na autenticação do usuário (SOUZA, 2006).

Como exemplo de mecanismos de segurança avaliados, temos certificação digital (GONZAGA, 2004), Secure Sockets Layer (SSL) (TANENBAUM, 2003), tecnologias de *firewall* e gerenciamento de *login* através de senha, impressão digital e imagem digitalizada da íris.

Naturalmente, considerando que o HUSM é um hospital público, deve-se considerar também os requisitos da arquitetura e-PING[1] (BRASILEIRO, 2006), que tem como propósito ser o paradigma para o estabelecimento de políticas e especificações técnicas que permite a prestação de serviços eletrônicos de qualidade à sociedade brasileira.

[1] "Padrões de Interoperabilidade de Governo Eletrônico"

1.1 Justificativa

Esse projeto aproveita as tecnologias na área de informática para facilitar o acesso a dados clínicos e melhorar o atendimento ao paciente. Um exemplo é a consulta, a partir de sua residência, do resultado de um exame pelo paciente, eliminando a necessidade de deslocamento até o laboratório para obter o resultado.

Pelo fato da aplicação ser desenvolvida num ambiente *web* e, levando em consideração que as informações acessadas são de interesse pessoal, deve haver restrição de acesso para tais informações.

O estudo dos mecanismos de autenticação tem como objetivo resolver esse problema, onde as informações consideradas sigilosas não podem ser acessadas por exemplo, por usuários mal intencionados, onde de alguma forma podem prejudicar um usuário cadastrado do sistema a partir de informações confidenciais desta pessoa.

Dessa forma, surgiu a idéia de desenvolver o Portal *Web* para o HUSM, uma aplicação específica para a área de saúde, numa instituição que ainda não explorou a tecnologia para ser utilizada através da Internet. Este portal é um ponto comum de entrada para o acesso humano (todos os médicos, pacientes, e equipe de funcionários) aos dados médicos.

A autenticação sob demanda é decorrente, por exemplo, de médicos e pacientes estarem acessando o mesmo sistema, onde um paciente poderá consultar o resultado de exames, mas não poderá cadastrar novos. Somente o médico, aumentando seu nível de autenticação, pode efetuar este cadastro. Por isso, a previsão de uma arquitetura de segurança que possa assegurar a privacidade e a segurança de dados sensíveis referentes a cuidados de saúde é ainda uma questão em aberto.

A escolha dos mecanismos foi feita baseada nas soluções de autenticação para cada caso, tendo dessa forma variados métodos, permitindo o seu uso no portal. Para o aspecto baseado no conhecimento, foi escolhido usuário/senha, por ser o mais utilizado nos mais diversos sistemas e também por ser prático e simples, onde o usuário precisa apenas ter conhecimento de uma senha. Certificado digital é baseado no que se tem e é considerado um mecanismo forte. A escolha deu-se pelo fato que existe uma autoridade certificadora confiável que valida a chave privada do usuário, possuindo dessa forma uma chave criptográfica própria a partir da qual poderá comprovar sua identidade através de assinaturas digitais. E impressão digital, por ser dentre os mecanismos biométricos o mais popular, além disso, é simples sendo que o usuário precisa apenas colocar seu dedo sobre a super-

fície do aparelho. Enquanto outras tecnologias, como por exemplo, reconhecimento da íris podem causar desconforto.

1.2 Objetivos

Este trabalho teve como objetivo explorar o uso de mecanismos de autenticação sob demanda, para controlar o acesso *web* a dados clínicos, bem como implementar um portal para o HUSM, o qual teve como principal objetivo homologar o estudo. Para atender ao objetivo geral, têm-se os seguintes objetivos específicos:

- implementação de um portal para acesso de informações via navegador *web*;
- padronização do sistema com e-PING;
- permitir que a aplicação fosse acessada por usuários previamente cadastrados através de *login* e senha, bem como através de mecanismos mais fortes de autenticação como, por exemplo, identificação através de reconhecimento de impressão digital;
- identificação através de certificação digital, onde somente usuários cadastrados com Certificação Digital possam ser autenticados; e
- gravação em log todas as entradas no sistema para fins de auditoria.

Este projeto estuda e implementa um mecanismo de autenticação que possibilite o portal do HUSM tratar níveis distintos de usuário (WEAVER, 2006) e de requisições de acesso através de níveis distintos de autenticação.

1.3 Trabalho relacionado

Um projeto semelhante (WEAVER et al., 2003) foi desenvolvido na Universidade de Virgínia, da mesma forma, a idéia foi implementar um portal para o acesso de dados clínicos. O serviço de autenticação é acessado a qualquer hora, no momento em que o usuário pretende estabelecer uma conexão e comprovar sua identidade. Nos sistemas antigos essa autenticação poderia ser feita simplesmente através de *passwords* de desafios/respostas. Mas com o progresso da medicina e com o auxílio dos dispositivos biométricos, podemos ter agora requisições variadas, as quais podem necessitar múltiplos tipos de autenticações.

Dessa forma, a confiança associada com o método de autenticação varia de acordo com o método utilizado. Por exemplo, usuário/senha provê um nível fraco de confiança

comparado com impressão digital, que por sua vez, é um método mais fraco comparado com reconhecimento da íris, este mais fraco comparado com reconhecimento da retina.

O serviço de autenticação é responsável por determinar o nível de confiança associado com a tecnologia utilizada e por definir as regras requeridas para estabelecer a identidade de um grupo específico (médico, paciente, enfermeiro).

Um exemplo deste projeto é o uso de um PDA HP5455 com um dispositivo de impressão digital, onde um médico decide encaminhar uma nova prescrição para um de seus pacientes. O médico primeiramente acessa o portal com uma requisição para acessar o histórico deste paciente. Quando são requisitadas informações do portal, o serviço de autenticação define um nível de confiança necessário para esta requisição. Se o usuário já possui um *ticket* válido com o nível mínimo exigido, o portal obtém os dados de resposta. Entretanto, se um *ticket* válido não existe ou se existe e é de nível inferior, o usuário é redirecionado para uma página de *login*.

Como forma de ilustração, considere que o primeiro acesso do médico faça uma requisição e que não tenha nenhum *ticket*. Por padrão, o serviço de autenticação permitirá que seja feita a identificação através dos seguintes mecanismos: usuário/senha, impressão digital, escaneamento da íris, smartcard ou outro mecanismo suportado. Neste caso, o serviço de autenticação verificará a modalidade de dispositivo suportado e a regra que deve ser usada para a requisição, referente ao nível de confiança exigido.

Quem define o nível de confiança para cada serviço disponível é a administração do hospital. Por exemplo, acessar o histórico de um paciente requer confiança igual ou superior da provida por impressão digital. Se o computador de acesso suportar o escaneamento da íris, o acesso também será validado, pois este tipo de autenticação excede a por meio de impressão digital.

A regra de autenticação reforça a segurança da tecnologia de autenticação empregada para o acesso a dados. Também pode ser especificado o número de métodos necessários a serem utilizados para acessar tais informações ou recursos.

Enfim, um ponto importante é o uso de níveis de confiança para definir a maneira como um indivíduo deve ser identificado, e somente após ter sido autenticado, o sistema questionará se o acesso a determinados dados será autorizado.

O restante deste trabalho é organizado da seguinte maneira. Para a revisão da literatura têm-se os capítulos 2 e 3. O capítulo 2 apresenta alguns conceitos introdutórios sobre

autenticação e define os mecanismos de autenticação trabalhados. O capítulo 3 descreve a tecnologia de *web services*, usada para criar um serviço de autenticação. Para o desenvolvimento do projeto propriamente dito têm-se os capítulo 4 e 5. O capítulo 4 modela e descreve o portal e o capítulo 5 mostra os detalhes da sua implementação. Finalmente o capítulo 6 apresenta as considerações finais referentes ao trabalho.

2 AUTENTICAÇÃO

O tráfego de dados através da Internet faz com que possam surgir vários problemas de segurança, que vão desde o roubo de senhas e interrupção de serviços até problemas de personificação, onde uma pessoa faz-se passar por outra para obter acesso privilegiado. Com isso, surgiu a necessidade de autenticação, que consiste na verificação da identidade tanto dos usuários quanto dos sistemas e processos. A autenticação se torna necessário para oferecer um ambiente seguro na realização de transações e armazenamento dos dados pessoais de usuários.

Este capítulo conceitualiza autenticação e apresenta os mecanismos de autenticação mais conhecidos.

2.1 Soluções de Autenticação

A identificação de uma forma geral, pode basear-se em três aspectos: o que você sabe; o que você possui ou o que você é.

2.1.1 Autenticação baseada no conhecimento (O que se sabe)

Este é o tipo de autenticação mais utilizado para fornecer uma identidade para um sistema, que pode ser senhas, frase de segurança ou *Personal Identification Number* (PIN) sendo caracterizado por alguma informação que o usuário tenha conhecimento. A seguir, temos as soluções baseadas no conhecimento.

2.1.1.1 Usuário/senha

A autenticação através de senha é a mais utilizada em sistemas *web*. Através da senha é possível identificar o usuário e a partir disto, pode-se dar permissões e privilégios de acesso, registrar o que o usuário acessou, o horário e a duração do acesso, sabendo quem alterou um arquivo.

O uso de senhas apresenta algumas desvantagens que podem ser: senhas fáceis de adivinhar, como por exemplo, nome (de algum membro da família), datas (nascimento), documentos (RG, CPF, ...), usar seqüências (123456, qwerty, abcdef), anotar a senha e compartilhar a senha. Dessa maneira, a senha do usuário deve ser pessoal e intransferível, sendo este responsável por sua utilização.

Ao usar este mecanismo de autenticação, é recomendável que se usem senhas fortes que têm pelo menos 7 caracteres e misturam caracteres de 3 dos 4 grupos seguintes: letras minúsculas, letras maiúsculas, números e caracteres especiais.

2.1.1.2 Senhas Descartáveis

São usadas somente uma vez no processo de autenticação, evitando o ataque de captura e repetição de senha, devido ao fato que, a próxima conexão exigirá uma senha diferente. Um exemplo, é quando o usuário fizer um acesso ao servidor, este envia uma mensagem para o serviço de autenticação que receberá o nome do usuário e buscará na sua base de dados o número do telefone deste usuário, Uma mensagem SMS será enviada com a senha para que o usuário possa efetuar login no servidor, também será enviado ao servidor a mesma senha, para que o servidor possa verificar se são iguais (LTD, 2006).

2.1.1.3 Perguntas Randômicas

É um mecanismo de autenticação baseado em desafio/resposta. Primeiramente, faz-se um cadastro do usuário, no qual ele responde a um questionário com perguntas diversificadas. Futuramente, ao fazer o acesso no servidor o usuário informa sua identificação. O sistema escolhe uma pergunta aleatoriamente, das que foram preenchidas pelo usuário durante a fase de cadastramento, e desafia o usuário. Se a resposta for igual à resposta fornecida no questionário, é garantido o acesso (FIORESE, 2000).

2.1.2 Autenticação Baseadas na Propriedade (O que se tem)

Caracteriza-se por um objeto físico que o usuário possui, dentre os quais pode ser um certificado digital, um cartão inteligente (*smartcard*), uma chave ou um *token*, que serve para calcular senhas descartáveis.

Apresenta as desvantagens que estes objetos podem ser perdidos, roubados ou extraviados, além do custo adicional do hardware. A vantagem diz respeito ao princípio de que a duplicação do dispositivo de autenticação sendo utilizado poderá ser mais cara que

o valor do que está sendo protegido. Os principais objetos que um usuário pode ter são descritos a seguir.

2.1.2.1 Certificado Digital

A principal função de um certificado digital é ligar o nome de um protagonista (pessoa, empresa) a uma chave pública referente a chave privada que acredita-se ser de posse unicamente da entidade especificada no certificado, os quais não precisam ser secretos, qualquer pessoa poderia exibir o seu conteúdo para outros (TANENBAUM, 2003).

O certificado é assinado digitalmente para garantir a integridade das informações contidas nele. A Autoridade Certificadora (AC) é responsável pela assinatura dos certificados, que são atestamentos feitos por essa AC que diz confiar nestes dados, certificando assim, as chaves públicas dos protagonistas.

Um Certificado Digital normalmente apresenta as seguintes informações:

- Nome da pessoa ou entidade a ser associada à chave pública;
- Período de validade do certificado;
- Chave pública;
- Nome e assinatura da entidade que assinou o certificado; e
- Número de série.

O mecanismo de assinatura digital pode ser descrito sucintamente. Como uma função *hash* unidirecional que a partir de um texto simples calcula uma *string* de bits de tamanho fixo, onde temos uma mensagem simples, a partir da qual se pode, por exemplo, gerar um *hash* de 160 bits do *Secure Hash Algorithm* 1 (SHA-1). Em seguida esse *hash* gerado pode ser assinado com a chave privada da AC. Nessa assinatura digital, o documento não sofre qualquer alteração e o *hash* cifrado com a chave privada é anexado ao documento.

Para comprovar uma assinatura digital são necessários realizar dois passos: descriptografar o bloco de assinatura com a chave pública da AC e executar o algoritmo de função *hash* no certificado. Se forem iguais, a assinatura está correta, o que significa que foi gerada pela chave privada corresponde à chave pública utilizada na verificação e que o documento está íntegro. Caso sejam diferentes, a assinatura está incorreta, o que significa que pode ter havido alterações no documento ou na assinatura pública.

Um exemplo do uso de certificados digitais é o serviço bancário provido através da Internet. Os bancos usam certificado para autenticar-se perante o cliente, garantindo que o cliente está realmente acessando o servidor do banco. O cliente também pode ao solicitar um serviço, se autenticar com o servidor do banco emitindo o seu certificado digital.

No Brasil, o Comitê Gestor da ICP-Brasil é o órgão governamental responsável por especificar os procedimentos que devem ser adotados pelas ACs e controla seis ACs: a Presidência da República, a Receita Federal, o SERPRO, a Caixa Econômica Federal, a Serasa e a CertiSign (DIGITAL, 2006).

2.1.2.2 Tokens

Tokens são dispositivos semelhantes a uma calculadora de mão e que não necessitam de dispositivos de leitura/escrita adicionais. Baseiam-se em um dos seguintes esquemas: autenticação por desafio/resposta ou autenticação sincronizada no tempo.

Nos sistemas baseados em desafio/resposta, o usuário insere sua identificação no sistema. O sistema apresenta, então, um desafio randômico como, por exemplo, na forma de um número de sete dígitos. O usuário, por sua vez, digita seu PIN no token e informa o desafio apresentado pelo sistema. O *token*, gera a resposta correspondente cifrando o desafio com a chave do usuário, a qual ele informa ao sistema. Enquanto isso, o sistema calcula a resposta apropriada baseado no seu arquivo de chaves de usuários. Quando o sistema recebe a resposta do usuário, ele a compara com a resposta que acabou de calcular. Se forem idênticas, a conexão é permitida e são atribuídos ao usuário os direitos de acesso correspondentes (FIORESE, 2000).

2.1.2.3 Cartão inteligente

Cartões inteligentes que protegem a credencial eletrônica podem ser usados para transportar seguramente estas credenciais e chaves de usuários. Tomando em conta que a credencial é armazenada permanentemente no cartão, esta nunca está disponível no programa ou na rede, para que um usuário não autorizado possa roubá-la (FIORESE, 2000).

2.1.3 Autenticação Baseadas em Características (O que se é)

Os sistemas biométricos baseiam-se em características físicas e comportamentais de pessoas. Uma característica física deve ser relativamente estável, tal como a impressão digital ou alguma característica facial. Assim são basicamente imutáveis ou variam pouco

no decorrer do tempo. Em contrapartida, uma característica comportamental reflete o estado psicológico de uma pessoa (isto é, pode ser afetada por problemas como estresse, fadiga, gripe, etc.). Entretanto, ela possui alguns elementos psicológicos que podem ser usados na identificação de uma certa característica. Por exemplo, o método de identificação baseado em comportamento mais comum é a assinatura de uma pessoa, usada pela sociedade há décadas. Outros comportamentos usados incluem o ritmo de digitação e o padrão de voz.

Os dispositivos biométricos estão sujeitos a falhas. São três os tipos de erros:

- Falsa rejeição do atributo físico de um usuário. O sistema não reconhece o padrão mesmo estando correto. É classificado na taxa de falsa rejeição.

- Falsa aceitação de um atributo físico. Neste caso, o sistema aceita a pessoa errada. O tipo de erro é classificado na taxa de falsa aceitação.

- Erro no registro de um atributo físico. São casos onde a variação de características físicas pode dificultar a operação do sistema. Alguém com problemas de voz, por exemplo, pode atrapalhar o funcionamento do dispositivo, aumentando a taxa de erro.

Como a biometria normalmente envolve o desenvolvimento de *hardware* específico, o custo pode ser alto dependendo do fator biométrico utilizado. A seguir, os métodos biométricos conhecidos.

2.1.3.1 Reconhecimento da Face

A autenticação é realizada através de uma câmera digital, que captura vários pontos delimitadores do rosto e define as proporções entre olhos, nariz, queixo, maças do rosto e orelhas, que são únicos a cada pessoa. O uso das características da face para identificação automática é uma tarefa difícil porque a aparência facial tende a mudar constantemente. Por exemplo, o uso de óculos, pode dificultar o processo de reconhecimento (SANTOS ROMAGNOLI, 2002).

2.1.3.2 Impressão Digital

Neste tipo de verificação, são usados algoritmos que analisam a posição de detalhes chamados de *minutiae*, o *American National Standards Institute* (INSTITUTE, 2004)

propôs a classificação das minúcias em quatro tipos: terminações, bifurcações, "cruzamentos"e indeterminado. A imagem de uma digital tem em média entre 30 e 40 detalhes únicos. Segundo estudos do FBI, duas pessoas não apresentam mais do que 8 pontos coincidentes (LIMA DANTAS, 2002).

Os leitores de digitais são compostos de três tipos de dispositivos: ópticos, ultra-som e baseados em chip. É importante que os dispositivos de impressão digital consigam minimizar a rotação da imagem. Problemas também podem ocorrer quando o usuário tem pequenos ferimentos no dedo, sujeira ou ressecamento da pele, para reduzir a percentagem de falsas rejeições podem ser feitas limpezas freqüentes (FIORESE, 2000).

As vantagens são: impressões digitais são únicas, são imutáveis, grande base de dados existente, grande quantidade de pesquisas e desenvolvimentos nessa área, praticidade de ser apenas necessário colocar a mão sobre o leitor e aguardar resposta (PEREIRA, 2003).

As desvantagens são: executa pesquisa 1:N, em um banco de dados grande pode se tornar um gargalo, o leitor da impressão pode estar sujo o que ocasionará falhas na autenticação e o tamanho (em bytes) do *template* das impressões digitais são relativamente grandes, variando em torno de 256 a 512 bytes por imagem, quando comparado com o tamanho de outros modelos.

2.1.3.3 Geometria da Mão

Se baseia no fato de que virtualmente não existem duas pessoas com mãos idênticas e de que o formato da mão não sofre mudanças significativas após certa idade. As vantagens no uso da forma tridimensional da mão da pessoa como um dispositivo de identificação são: método razoavelmente rápido, requer pouco espaço de armazenamento, é requerido pouco esforço ou atenção do usuário durante a verificação, e os usuários autorizados são raramente rejeitados (SANTOS ROMAGNOLI, 2002).

2.1.3.4 Reconhecimento de Retina

Os analisadores de retina medem padrão de vasos sangüíneos, usando um laser de baixa intensidade e uma câmara, que faz uma varredura para encontrar os padrões singulares da retina. É uma técnica de muita precisão e praticamente impossível de ser adulterada devido a forte relação com os sinais vitais humanos. A captura da imagem, entretanto, é difícil e incômoda, pois é necessário que o usuário olhe fixamente para um ponto luminoso de infravermelho. Muitas pessoas ficam temerosas em colocar seu olho

próximo a uma fonte de luz e aos problemas que isso possa causar (SANTOS ROMAGNOLI, 2002).

2.1.3.5 Reconhecimento de Íris

O padrão da íris do olho humano (baseada nos anéis coloridos do tecido que circunda a pupila) guarda uma imagem muito complexa e, assim como a impressão digital, é única em cada pessoa. É considerada a menos intrusiva das tecnologias que envolvem o uso dos olhos para identificação, pois não requer um contato muito próximo com o dispositivo de leitura como no caso da retina. Além diso, lentes e óculos não muito escuros podem ser usados sem comprometer o desempenho do reconhecimento (SANTOS ROMAGNOLI, 2002).

2.1.3.6 Reconhecimento de Voz

O reconhecimento de voz é um dos sistemas menos invasivos e a forma mais natural de uso é o sistema de reconhecimento de fala. Para a captura do som, o usuário posiciona-se diante de um microfone e pronuncia uma frase previamente selecionada, ou uma frase qualquer. Este processo é repetido várias vezes até que seja possível construir um modelo.

A tecnologia de reconhecimento de voz é fácil de usar e não requer grandes esforços na educação do usuário. Entretanto, deve-se cuidar para garantir que o usuário fale em um tempo apropriado e em voz clara (FIORESE, 2000).

2.1.3.7 Reconhecimento de Assinatura

Existem dois métodos de identificação: um método examina a assinatura já escrita, comparando-a, com uma imagem de um modelo armazenado. O outro método estuda a dinâmica da assinatura, que consiste no ritmo de escrita, contato com a superfície, tempo total, pontos de curva, laços, velocidade e aceleração. Os usuários desta tecnologia se identificam bastante com o processo, por já estarem acostumados a utilizar a assinatura como meio de autenticação (SANTOS ROMAGNOLI, 2002).

2.1.3.8 Ritmo de Digitação

O ritmo de digitação exibe um fator neurofísico que pode ser utilizado na identificação única de um indivíduo. Esse método analisa o modo como um usuário digita em um terminal, monitorando o teclado 1000 vezes por segundo.

Normalmente é utilizado o método das latências de digitação, o tempo entre a digitação de duas teclas. Certos dígrafos, ou digitação de duas letras adjacentes, freqüentemente, apresentam padrões de tempo únicos que podem ser usados para caracterizar um indivíduo.

Uma das vantagens é que o usuário não percebe quando está sendo autenticado, a menos que ele tenha sido informado anteriormente. Outra vantagem é que o cadastro e a verificação não são invasivos (FIORESE, 2000).

3 *WEB SERVICES*

Neste capítulo, inicialmente é tratado sobre a conceitualização de *web services*, em seguida, apresentada a sua arquitetura descrevendo como ocorre a interração entre os papéis presentes nela e têm-se ainda a definição das tecnologias usadas.

Finalmente são descritos os protocolos de segurança para a troca segura de mensagens no acesso a um serviço. Este acesso não é feito via protocolos específicos de modelos de objetos como DCOM, RMI e IIOP, mas via protocolos de rede padrão como HTTP e FTP.

3.1 Conceito

Web services, como definido pela *W3C Web Services Architecture Working Group*, é uma aplicação de *software* identificada por um *Uniform Resource Identifier* (URI), cujas interfaces e ligações são capazes de ser definidas, descritas e descobertas como artefatos eXtensible Markup Language (XML). Um serviço *web* suporta interações diretas com outros agentes de *software* usando mensagens baseadas em XML, trocadas via protocolos baseados na Internet (BOOTH et al., 2004).

Pode se dizer que um *web service* é um programa disponibilizado na *web*, que pode ser incorporado por alguma aplicação que esteja sendo implementando. Para isso, é necessário fazer uma chamada ao serviço, sendo que, devemos saber o endereço onde este programa está, quais os parâmetros que ele necessita e o que será retornado. Se já conhecemos essas informações, basta apenas escrever a chamada diretamente no código de nossa aplicação, caso contrário, devemos fazer com que nossa aplicação consulte uma base de serviços e descubra como utilizar o programa desejado. A comunicação entre os serviços é padronizada possibilitando a independência de plataforma e de linguagem de programação (RHEINHEIMER, 2004).

Web services apresentam os melhores aspectos do desenvolvimento baseado em com-

ponentes na *web*. Assim, como componentes de *software*, *web services* contêm uma funcionalidade *black box* que pode ser reutilizada sem a preocupação com a linguagem e o ambiente utilizados em seu desenvolvimento (GRAHAM et al., 2004). Por exemplo, um sistema de reserva de passagens aéreas desenvolvido na plataforma Java e rodando em um servidor Linux pode acessar, com transparência, um serviço de reserva de hotel desenvolvido na plataforma .Net rodando em um servidor Windows.

3.2 Arquitetura de *Web Services*

A arquitetura de *web services* apresentada na figura 3.1 é baseada nas interações de três papéis:

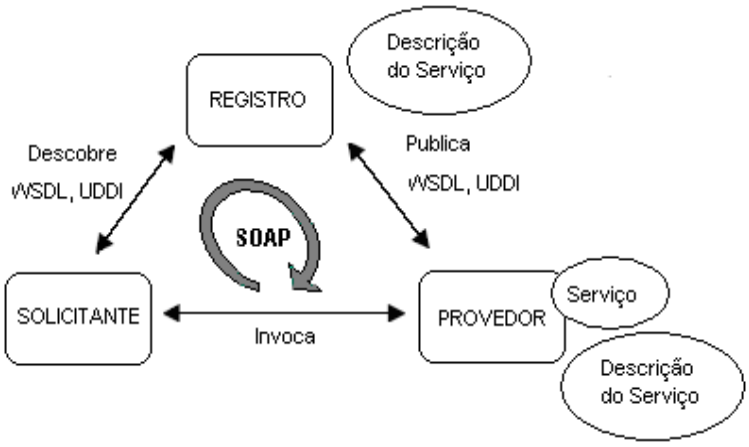

Figura 3.1: Arquitetura de *Web Services*.

- **O provedor de serviço:** é a plataforma acessada na solicitação do serviço. É quem cria o *web service*, sendo responsável por fazer sua descrição em algum formato padrão e publicar os detalhes em um registro de serviço central.

- **O solicitante de serviço:** é uma aplicação que invoca ou inicializa uma interação com um serviço. Pode ser um *browser* ou um *web service*. Usando a descrição do serviço é possível descobrir e invocar *web services*.

- **O registro de serviço:** é o local onde os provedores publicam as descrições dos serviços. Um solicitante consulta o registro, busca as informações de ligação da descrição do serviço durante a fase de desenvolvimento (ligação estática) ou em tempo de execução (ligação dinâmica).

A iteração entre os papéis acontece via rede, numa camada conceitual (Figura 3.2) de *web services*. A camada conceitual tem como camada base a rede que representa os protocolos de transporte (HTTP, FTP, SMTP). Esta camada é usada de acordo com as necessidades da aplicação: segurança, disponibilidade, desempenho e confiabilidade. Mensagem baseada em XML é a camada que utiliza o protocolo *Simple Object Access Protocol* (SOAP) para realizar a troca de mensagens num ambiente distribuído e descentralizado. A descrição do serviço é feita utilizando *Web Services Description Language* (WSDL). As camadas de publicação e descoberta de serviço usam o padrão *Universal Description, Discovery and Integration* (UDDI) para descoberta e publicação (GUDGIN et al., 2003).

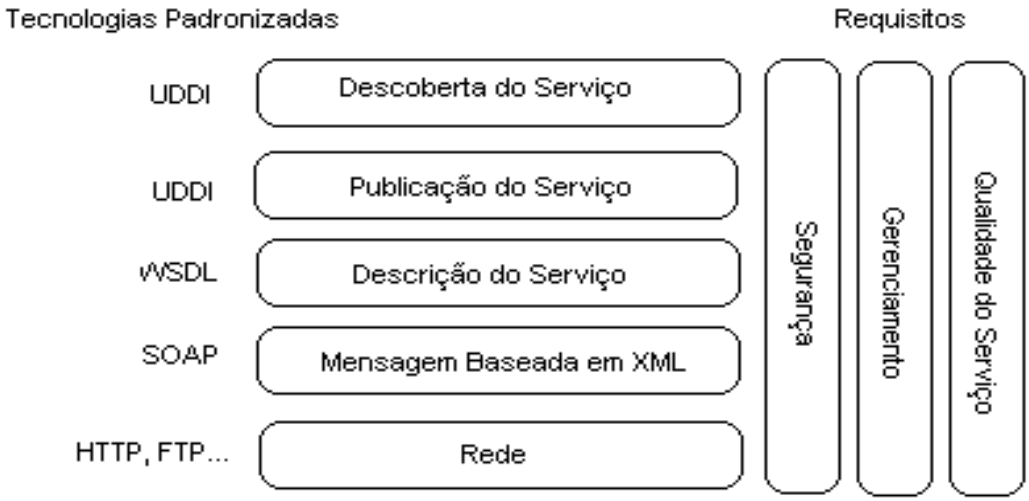

Figura 3.2: Camada Conceitual.

3.3 Tecnologias

As tecnologias que são usadas para a construção de uma estrutura básica de *web services* são WSDL, SOAP e UDDI.

3.3.1 SOAP

É um protocolo que padroniza o que está contido em um XML, que define um vocabulário para troca de mensagens eletrônicas. É o nome do envelope contendo uma mensagem codificada que irá trafegar sobre outros protocolos como HTTP ou FTP. Resumidamente, usa a estrutura XML para criar mensagens de requisição/resposta.

É um protocolo para troca de informação num ambiente distribuído e descentralizado,

permitindo comunicação entre várias aplicações (BOX et al., 2002). O protocolo SOAP pode ser usado com a combinação de outros protocolos, como HTTP, SMTP e JMS.

Nesta especificação alguns conceitos importantes são:

- **Cliente SOAP:** é um programa que cria um documento XML contendo informação necessária para invocar um método remotamente dentro de um sistema distribuído (poderia ser uma aplicação *web* ou um servidor de aplicação).
- **Servidor SOAP:** é responsável por executar uma mensagem SOAP, agir como um interpretador e realizar trocas de mensagem.
- **Mensagem SOAP:** é o formulário básico na comunicação entre nodos. A figura 3.3 ilustra o esqueleto de uma mensagem SOAP.

```
<soap:Envelope
        xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
        soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding»
        <soap:Header>
            ...
        </soap:Header>
        <soap:Body>
            ...
            <soap:Fault>
            ...
            </soap:Fault>
        </soap:Body>
</soap:Envelope>
```

Figura 3.3: Esqueleto de uma mensagem SOAP.

3.3.2 WSDL

É uma linguagem, descrita através de XML (PAOLI et al., 2006), utilizada para responder três perguntas sobre um *web service*: o que é o serviço, onde encontrá-lo e como chamá-lo. WSDL define os serviços como um conjunto de *endpoints*, isto é, pontos de acesso na rede (WEAVER et al., 2003).

Os elementos contidos em um documento WSDL estão listados a seguir:

- *types*: um container para definição de tipos;
- *message*: definição dos tipos dos dados que são passados nas operações;

- *operation*: descrição de uma ação disponibilizada pelo serviço;
- *port Type*: conjunto de operações suportadas nos *endpoints*;
- *Binding*: especificação do formato dos dados para um determinado *portType*;
- *Port*: é um *endpoint*, definido como a combinação de um binding com um endereço de rede;
- *Service*: um conjunto de *endpoints*.

A linguagem WSDL descreve um serviço como uma coleção de operações que podem ser acessadas através de mensagens (CHRISTENSEN et al., 2001).

3.3.3 UDDI

A especificação UDDI é um trabalho conjunto para criação de um registro de serviços padronizado, este registro pode ser acessado por clientes que podem localizar todos os serviços dos quais precisam (EHNEBUSKE; ROGERS; RIEGEN, 2001).

Ela possui um componente central chamado UDDI *Project*, que manipula um registro global e público chamado *business registry*. A informação oferecida pelo *business registry* consiste de três componentes:

- *white pages* - endereço, contato e identificadores conhecidos;
- *yellow pages* - categorização industrial;
- *green pages* - informação.

A implementação UDDI é um servidor de registro que fornece um mecanismo para publicar e localizar serviços. Guarda informações categorizadas sobre empresas, serviços que elas oferecem e a associação com as especificações desses serviços, feitas em WSDL através do próprio registro (HANSEN et al., 2002).

3.4 Protocolos de Segurança para *Web Services*

A tabela 3.4 cita os protocolos de segurança existentes para *web services*.

Status	Padrão	Organização de Padronização
Disponível	SSL/TLS	IETF
	XKMS 1.0	Implementações fechadas
	SAML 1.0	OASIS
Em Desenvolvimento	XKMS 2.0	W3C
	SAML 1.1	OASIS
	WS-Security	OASIS
	XACML	OASIS
	XrML	OASIS
Proposta	WS-Policy, WS-SecurityPolicy	
	WS-SecureConversation	
	WS-TrustAxiom	
	WS-KeyAgreement	
	XML Packed Encoding	

Tabela 3.1: Protocolos de Segurança para *Web Services* (VERISIGN, 2005).

3.4.1 WS-Security

A WS-Security aborda a segurança das mensagens SOAP, ao especificar um perfil para o uso da assinatura XML (XML *signature*) e da criptografia XML (XML *encryption*) que permite a garantia de integridade e confidencialidade para mensagens SOAP (VERISIGN, 2005).

- ***XML Encryption*:** definido pelo W3C, apresenta um vocabulário para garantir a confidencialidade do conteúdo de documentos XML através do uso de algoritmos criptográficos. Esse padrão é utilizado com o propósito de manter a confidencialidade de informações que estão em trânsito ou armazenadas. XML Encryption permite que diferentes elementos de um documento sejam cifrados separadamente, ou apenas partes de um documento XML sejam cifradas (IMAMURA; DILLAWAY; SIMON, 2002).

- ***XML Signature*:** foi especificado pelo W3C e é utilizado para representar assinaturas digitais. Essa especificação se aplica a requisitos de segurança para a autenticação de documentos digitais, verificação da integridade de documentos digitais e também a não-repudiação de documentos, uma vez que esses documentos foram assinados. Os algoritmos utilizados nessa especificação incluem o algoritmo de chave pública *Digital Signature Standard* (DSS) e o algoritmo de autenticação *Secure Hash* (SHA-1). Todavia, desenvolvedores podem utilizar o XML *Signature* para suportar seus próprios algoritmos e modelos de segurança. XML *Signature*

> pode ser utilizado para assinar qualquer tipo de arquivo, não somente documentos XML, e os dados assinados podem estar dentro ou fora do documento XML que contém uma assinatura. São utilizadas URIs para associar as assinaturas aos objetos de dados assinados (BARTEL et al., 2002).

A especificação WS-Security define um modelo geral de segurança de aplicações que protege cada mensagem SOAP individualmente, pela adição de campos ao cabeçalho das mensagens. Estes campos permitem que cada nó da infra-estrutura determine o nível de confiança de cada mensagem de forma independente.

O padrão WS-Security define como devem ser inseridas credenciais nas mensagens SOAP. Estas credenciais podem ser identificadores de usuários, senhas, *tickets* ou *tokens* de segurança, certificados ou assinaturas digitais e há a previsão da utilização dos formatos já existentes no mercado. O padrão WS-Security permite garantir a integridade e confidencialidade das mensagens e oferece também autorização e autenticação (CARVALHO FERREIRA, 2005).

3.4.2 Segurança para *Web Services* utilizando SSL/TLS

Se for necessário implementar seguramente um *web service* hoje em dia, algumas opções padrão disponíveis são o uso de uma Rede Privada Virtual baseada em IPSec, ou de um protocolo *Secure Socket Layer* (SSL) para proteger o transporte de mensagens SOAP. O protocolo SSL é um protocolo de segurança robusto que foi padronizado na IETF como *Transport Layer Security* (TLS). É amplamente utilizado em e-commerce e tem sido usado para proteger transações envolvendo muitos bilhões de dólares.

O SSL fornece confidencialidade através da criptografia e integridade de mensagens, por meio de um código de autenticação de mensagem. O protocolo tem duas fases; na primeira, o cliente estabelece um segredo compartilhado com o serviço, utilizando um mecanismo de acordo de chaves. Na segunda fase, os pacotes de informação trocados entre o cliente e o servidor são criptografados e autenticados utilizando uma chave derivada do segredo compartilhado (VERISIGN, 2005).

3.4.3 XKMS

O padrão XML *Key Management Specification* (XKMS) define um protocolo para registrar e distribuir chaves criptográficas em serviços *web* que utilizam infraestrutura de

chave pública (PKI). O XKMS foi inicialmente desenvolvido pela Microsoft em conjunto com a VeriSign, depois tornou-se uma iniciativa do W3C. O gerenciamento de chave pública inclui a criação de par de chaves pública e privada, a ligação dessas chaves com uma identidade e disponibilização dessas chaves em diferentes formatos.

O XKMS foi criado para auxiliar a distribuição de chaves públicas, que são essenciais para XML *Signature* e XML *Encryption*, possibilitando verificações de assinaturas e criptografia em documentos XML (VERISIGN, 2005).

O padrão XKMS é dividido em duas especificações a XML *Key Registration Service Specification* (X-KRSS), para o registro de chaves, e o XML *Key Information Service Specification* (X-KISS), para requisição e distribuição de informações referentes às chaves públicas registradas (FORD et al., 2001).

No gerenciamento de chave pública é necessário que ocorra um registro de chaves onde um par de chaves é criado e atribuído a uma identidade qualquer. Também é necessário que seja possível revogar essa associação com o par de chaves caso ocorra, por exemplo, um roubo da chave privada.

3.4.4 SAML

O SAML (*Secure Assertion Markup Language*) é um padrão para a transferência de informações de autorização e autenticação através da Internet. O SAML é desenvolvido pelo OASIS *Security Services Technical Commitee* (SSTC) como um padrão XML para comunicação em plataformas de B2B (*Business-To-Business*) e B2C (*Business-To-Consumer*) (CANTOR et al., 2001).

O SAML pode ser definido como uma Infraestrutura de Gerenciamento de Permissões (*Permissions Management Infrastructure* - PMI), e utiliza um conjunto de políticas de segurança para definir um controle de acesso e autorização sobre informações e outros recursos disponíveis em um sistema computacional. O SAML foi desenvolvido para prover interoperabilidade entre aplicações e pode ser utilizado em conjunto com protocolos de comunicação XML como SOAP. O padrão SAML permite que usuários enviem as informações para autenticação apenas uma única vez, e sejam autenticados através de vários domínios. Esse processo de autenticação única é chamado de *Single Sign-On* (SSO).

O padrão SAML apresenta um vocabulário para expressar asserções de autorizações e autenticação. Asserções são distribuídas por autoridades SAML, isto é, autoridades de

autenticação, autoridades de atributos e PDP (*Policy Decision Point*). O SAML também define um protocolo através do qual clientes podem requisitar asserções à autoridades SAML e receber a resposta para essas asserções.

Uma asserção é um pacote de informação que carrega uma ou mais declarações de uma autoridade SAML. O padrão SAML define três tipos de asserções que podem ser criados por uma autoridade SAML: na asserção de autenticação o sujeito especificado foi autenticado para um meio particular em um momento particular; na asserção de atributo o sujeito especificado está associado com atributos fornecidos; e na asserção de decisão de autorização é determinada uma decisão de acesso ao sujeito sobre um recurso específico.

Assim, asserções podem conter informações sobre ações de autenticação, atributos de sujeitos e decisões de autorização sobre permissões aos sujeitos para acesso a certos recursos. Uma asserção simples pode conter diversas declarações internas sobre autorização, autenticação e atributos. Por exemplo, uma declaração de autorização SAML pode possuir campos para identificação da declaração como identificador, nome do sujeito, domínio de segurança que o sujeito pertence, informações como a hora da emissão e quem foi o emissor e as condições para as quais a asserção é válida (VERISIGN, 2005).

4 PORTAL PARA O HUSM

Neste capítulo, têm-se a definição do portal para o HUSM com seus requisitos e sua arquitetura. É feito um comentário sobre o serviço de autenticação e finalmente, tratado sobre a Aplicação SIE-Saúde alvo.

4.1 Definição do Portal

Com o rápido crescimento e uso da Internet, tornou-se possível a criação de aplicações na área de saúde, mas a provisão de uma arquitetura de segurança de dados sigilosos ainda é uma questão em aberto. Neste projeto, acredita-se que *web services* representam uma tecnologia que promete resolver este problema. Para estes fins, foi um sistema para proteger com segurança a privacidade dos dados.

O portal do hospital é o ponto comum de entrada para os usuários acessarem os dados hospitalares, desde médicos, pacientes, equipe de funcionários entre outros. A autenticação é flexível e deve avançar conforme a tecnologia, os mecanismos empregados são *passwords*, certificado digital, reconhecimento de impressão digital. Também é sabido que diferentes tecnologias de autenticação tem diferentes níveis de confiabilidade, devido à isso, será introduzido o conceito de nível de confiança para cada tecnologia de autenticação utilizada.

Para a autenticação sob-demanda, o usuário se autentica utilizando alguma tecnologia disponível em seu computador. O acesso ao portal prossegue até o momento em que o usuário tentar acessar uma informação cuja regra requer uma autenticação com nível maior daquela provida correntemente pelo usuário. Neste momento, é dado ao usuário a oportunidade de elevar seu nível utilizando uma tecnologia de maior confiabilidade.

Toda a parte de autorização do portal é dirigida a políticas, ou seja, baseada em regras expressas em documentos XACML (*eXtensible Access Control Markup Language*).

Para esta finalidade, foi utilizado o CIBAC, modelo de controle de acesso que representa políticas de autorização através de documentos XACML (SACCHET, 2007).

As necessidades esperadas deste sistema leva a um conjunto de características e funções dos quais devemos atender os seguintes requisitos:

- Listar as aplicações.
- Segurança para autenticação de usuários.
- Segurança na parte de autorização das aplicações acessadas e as ações realizadas em cada aplicação.
- Padronização com e-PING.
- Auditabilidade.

4.2 Arquitetura do portal

A figura 4.1 ilustra a arquitetura do portal.

Nesta arquitetura do portal, é tratada a parte de segurança, onde temos o serviço de autenticação do portal que ocorre nos passos a seguir:

1. Inicialmente, o usuário acessa a página principal do portal. A partir desse momento, é necessário efetuar o *login*. Nesta operação, o usuário irá escolher dentre os tipos de mecanismos disponíveis no sistema e suportados por sua máquina (leitor de impressão digital) para efetuar sua autenticação. Dependendo do tipo de autenticação escolhido pelo usuário, se vier a ser autenticado, ele receberá um nível de confiança para acessar o sistema.
2. Neste passo, o sistema receberá os dados fornecidos pelo usuário e fará uma requisição de autenticação para o serviço de autenticação (*web service*). Para que possa fazer esta requisição, o sistema terá que processar os dados enviados pelo usuário para formatar uma mensagem SOAP enviada através do protocolo HTTP.
3. O serviço de autenticação receberá o dado desta requisição e se conectará com um servidor de banco de dados para validar os dados que recebeu.
4. A partir da validação, o serviço de autenticação retornará uma resposta informando se o usuário foi autenticado ou não.

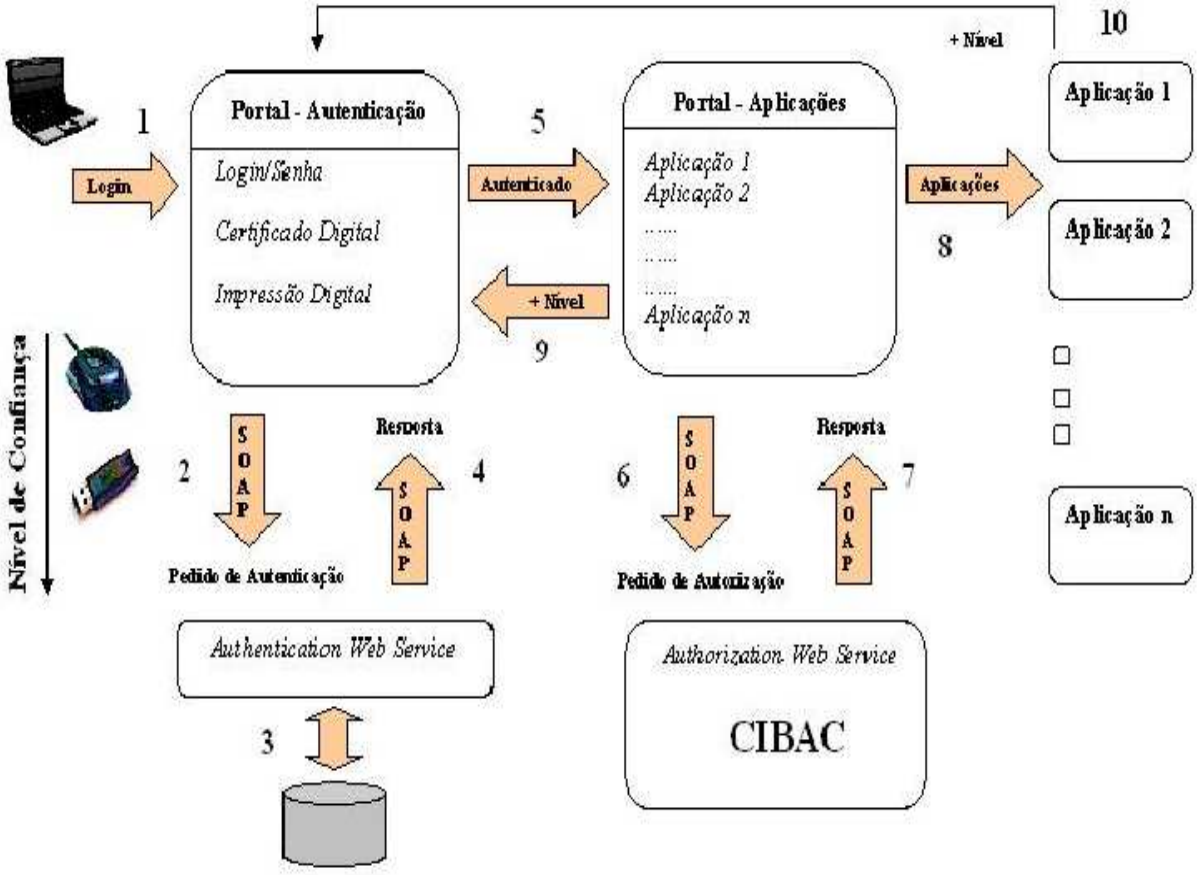

Figura 4.1: Arquitetura do portal.

5. A partir desses passos, se o usuário foi autenticado, poderá começar a navegação pelo portal e terá acesso a uma página com todas as aplicações disponíveis no sistema, podendo decidir qual das aplicações acessará. A partir desse momento, entra a segurança a nível de autorização.

6. Neste momento, ocorre a comunicação entre o portal e o CIBAC. Quando o usuário tentar acessar uma das aplicações do sistema, o Portal deverá fazer um pedido de autorização ao CIBAC, para verificar se o usuário pode ou não acessar tal aplicação (SACCHET, 2007).

7. O corre a resposta do CIBAC ao pedido de autorização feito pelo portal. De acordo com a resposta, o usuário permanecerá na página de aplicações (Portal - Aplicações) ou terá acesso a aplicação solicitada (SACCHET, 2007).

8. Representa o acesso a uma das aplicações solicitadas pelo usuário (Portal - Aplicações).

9. Se o usuário for redirecionado para a página de *login*, significa que ele não teve acesso a determinada aplicação, ou seja, o CIBAC não deu autorização de acesso. Esta restrição aconteceu devido ao usuário ter se autenticado com um nível de confiança abaixo do esperado pela aplicação ou porque este usuário não deve ter acesso a esta aplicação. Neste momento temos o conceito de autenticação sob-demanda, onde o usuário deverá elevar seu nível de confiabilidade para poder acessar determinadas aplicações.

10. Quando este passo ocorre, significa que o usuário não pode executar determinada ação (consulta, alteração, remoção, inserção, impressão, etc) na aplicação, ou seja, o CIBAC deu autorização de acesso a aplicação mas não permitiu tal ação na mesma. Esta restrição aconteceu devido ao usuário ter se autenticado com um nível de confiança abaixo do esperado pela aplicação ou porque este usuário não deve desempenhar tal ação nesta aplicação.

4.3 Serviço de Autenticação

Neste projeto, o serviço de autenticação funciona para as autenticações feitas através de usuário/senha, impressão digital e certificado digital[1].

O serviço foi desenvolvido utilizando *web service* porque o padrão e-PING exige desacoplamento, usando *web service* garantimos este desacoplamento para o sistema. Além disso, qualquer mudança no serviço não afetará as respostas para o usuário. Isto é útil para o projeto porque permite que as tecnologias usadas na autenticação possam ser atualizadas sem a necessidade de remodelar completamente o sistema.

4.4 Aplicação SIE-Saúde alvo

Nesta seção será detalhada uma das aplicações do SIE-Saúde que foi parcialmente migrada para o ambiente *web*. A aplicação implementada foi *Realização de Exames* (4.4.1). Esta representa apenas uma aplicação que poder ser acessada inicialmente sendo que posteriormente poderão ser incorporadas novas aplicações ao portal.

[1]Não foi utilizado o servidor de autenticação *Kerberos* por não fazer parte da especificação inicial do projeto.

4.4.1 Realização de Exames

Esta aplicação é usada para registrar a realização de um exame. Ela poderá ser utilizada pelos técnicos ou pelos médicos que realizam os exames.

Para registrar a realização de um exame, deve-se primeiro localizar os exames agendados (seção 4.4.2 - Localizar Exames Agendados). Após localizar um exame que deverá ser realizado, o usuário terá que informar o *Técnico Responsável*, que é o profissional que efetivamente realizou o exame, em alguns casos, poderá ser o próprio médico responsável pelo exame. O campo *Médico Responsável* (seção 4.4.2 - Localizar Profissional Prestador), é o profissional responsável pelo exame. Caso o médico tenha também realizado o exame, o seu nome irá constar duas vezes na tela, uma como técnico e outra como profissional responsável.

Após informar todos os dados necessários, podemos salvar estas informações no banco de dados. Caso o paciente tenha passado mal durante a realização do exame há a opção *Exame não realizado* para que seja informado o motivo da não realização. Algumas outras informações podem ser adicionadas através do campo *Observação*.

4.4.2 Localizadores Utilizados na Aplicação

Na aplicação *Realização de Exames* são utilizados alguns campos de localização. Esta seção apresenta estes localizadores.

- **Localizar Exames Agendados:** Permite realizar a busca de exames agendados utilizando diversos filtros de pesquisa.
- **Localizar Exames Realizados:** Semelhante ao *Localizar Exames Agendados*, diferente apenas nos filtros a serem utilizados e nos dados retornados.
- **Localizar Profissional Prestador:** Permite realizar a busca do profissional prestador, ou seja, o profissinal responsável pela realização do exame.
- **Localizar Prestador:**Permite realizar a busca do prestador, ou seja, a instituição responsável pela realização do exame.

4.4.3 Ambiente Web

A aplicação alvo num ambiente web serve, juntamente com as outras aplicações do portal (que possuem apenas a página inicial), para verificar a autenticação sob-demanda.

O portal possui uma listagem de aplicações, para que um usuário possa acessá-las com diferentes métodos de autenticação conforme políticas de acessos definidas. No caso de o usuário acessar uma aplicação, efetua seu *login*, se a próxima que acessar exigir um nível maior de confiança, deverá elevar seu nível.

Realização Exames também serve para verificar a autenticação sob-demanda nas ações que serão feitas na aplicação, onde um usuário se autentica com um mecanismo considerado de nível baixo para efetuar alguma ação qualquer, mas decide realizar uma ação que requer maior confiança, para isso é necessário efetuar no portal uma nova autenticação com um mecanismo de maior confiança.

5 IMPLEMENTAÇÃO

Neste capítulo, será tratado sobre a implementação do serviço de autenticação e também detalhado sobre cada um dos mecanismos presentes. Em seguida, descrito como foi implementado o portal para o HUSM e por último, apresentado um estudo de caso que demonstra uma situação em que pode ocorrer a autenticação sob-demanda.

5.1 Implementação do Serviço de Autenticação

Nesta seção, será descrita a parte do projeto referente ao serviço de autenticação do sistema, este serviço foi implementado utilizando a tecnologia de *web services*. Atualmente, existem diversos *frameworks* que facilitam a geração de *web services* baseados no SOAP. A escolha deu-se pelo Apache Axis2 (FOUNDATION, 2007), pelo fato de ser uma ferramenta *open source*, de uso comercial livre e muito adotado entre os desenvolvedores. Além disso, a criação e publicação de *web services* e a criação dos clientes de acesso é gerada de forma simplificada e forma automática, com o auxílio das ferramentas disponíveis no próprio Apache Axis2, onde a codificação utiliza a plataforma Java.

Para este serviço devemos levar em conta a segurança nos *web services* considerado a troca de mensagens SOAP. O Apache Axis2 suporta a implementação de segurança através do projeto WSS4J (FOUNDATION, 2006a), que é uma implementação do padrão OASIS *Web Services Security* (STRUCTURED INFORMATION STANDARDS, 2006) (WS-Security). WSS4J pode ser usado como uma biblioteca para prover uma API em Java para processar WS-Security. Neste caso, WSS4J não precisa ser usado com Axis, ou ser parte do *deploy* da aplicação *web*, esta API pode ser invocada diretamente para criar cabeçalhos com a especificação WS-Security ou processar estes cabeçalhos a partir de um envelope SOAP.

Basicamente, o WSS4J trabalha com dois níveis de segurança que são:

- **XML Security**
  - **XML Signature**
  - **XML Encryption**
- **Tokens**
  - **Username Tokens**
  - **Timestamps**
  - **SAML Tokens**
- e suporta certificados X.509.

Usando as tecnologias citadas, foi implementado uma classe contendo métodos específicos para cada mecanismo com o qual se pretende trabalhar, esta classe será usada no *deploy* juntamente com as bibliotecas utilizadas. Para a disponibilização deste serviço é necessário seguir os seguintes passos.

1. Instalar o servidor Apache Tomcat (FOUNDATION, 2006b) de aplicações Java para *web*.
2. Baixar o arquivo axis2.war.
3. Criar a pasta "meuservico"com a estrutura de diretórios da figura 5.1, onde classes representa as classes criadas e *services.xml* contém informações do serviço que são: nome, qual a sua classe e como será invocado pelo cliente.

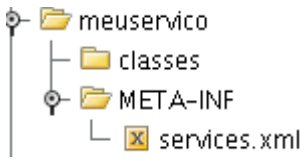

Figura 5.1: Estrutura de diretórios do serviço criado.

4. A pasta meuservico deve ser copiada para o diretório WEB-INF/services/ da pasta axis2.war e as bibliotecas utilizadas para o diretório WEB-INF/lib.
5. Mover a pasta axis2.war para dentro da pasta webapps do Tomcat e inicializar o servidor.

5.1.1 Usuário/Senha

Para a autenticação através de usuário e senha, foi utilizada a estratégia de *Username Token*, que adiciona informações de usuário e senha ao *header* (cabeçalho) do SOAP de requisição. A senha pode ser enviada no formato *plain text*, ou seja, texto literal (Figura 5.2), ou através de alguma encriptação utilizando o formato *Digest*.

O documento SOAP de requisição conterá as informações do cabeçalho, contidas na *tag <soapenv:Header>*:

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
      xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext»
      <S:Header>
          ...
          <wsse:Security>
                <wsse:UsernameToken >
                      <wsse:Username> eduardo </wsse:Username>
                      <wsse:Password> teste01 </wsse:Password>
                </wsse:UsernameToken>
          </wsse:Security>
          ...
      </S:Header>
      ...
</S:Envelope>
```

Figura 5.2: *Username token* no formato *plain text*.

É o cabeçalho do SOAP que contém as informações de autenticação através da *tag Username Token* onde são informados o usuário e a senha, respectivamente. Na senha também é informado o tipo usado. No caso, *wsse:PasswordText* informa que está no formato texto literal, que é pouco seguro.

Uma forma mais segura de informar a senha, é através do modo *PasswordDigest*, que criptografa a senha. A criptografia é baseada em 3 informações: *Nonce*, *CreationTimestamp* (data e hora da criação) + senha. É aplicado o algoritmo SHA-1 no resultado da concatenação destes dados e depois dividido na *Base64*, conforme a fórmula abaixo:

PasswordDigest = *Base64* / SHA-1(*Nonce* + *CreationTimestamp* + Senha)

A figura 5.3 mostra um exemplo do SOAP com a senha no formato *PasswordDigest*, a qual é utilizada pelo serviço de autenticação em questão.

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext»
        <S:Header>
            ...
            <wsse:Security>
                <wsse:UsernameToken
                    xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext"
                    xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/xx/utility»
                    <wsse:Username> eduardo </wsse:Username>
                    <wsse:Password Type="wsse:PasswordDigest»
                        D2A12DFE8D9F0C6BB82C89B091DF5C8A872F94DC
                    </wsse:Password>
                    <wsse:Nonce> EFD89F06CCB28C89 </wsse:Nonce>
                    <wsu:Created> 2001-10-13T09:00:00Z </wsu:Created>
                </wsse:UsernameToken>
            </wsse:Security>
            ...
        </S:Header>
        ...
</S:Envelope>
```

Figura 5.3: *Username token* no formato *PasswordDigest*.

O serviço de autenticação, ao receber esta requisição processa as informações do cabeçalho extraindo o conteúdo das *tags username*, *password*, *nonce* e *created*. Em seguida, verifica na base de dados se o usuário existe, caso não existe, não valida este usuário, caso contrário, pesquisa na base de dados a senha correspondente a este usuário, usando a senha juntamente com os valores contidos no *nonce* e em *created* calcula o valor no formato *Digest*, se for igual ao passado no cabeçalho o usuário é autenticado. Também é registrado em log o pedido de autenticação.

5.1.2 Certificado Digital

Para este tipo de autenticação, foram necessários criar certificados digitais. A ferramenta utilizada para criar os certificados digitais foi o OpenSSL. Primeiramente foi criado uma Autoridade Certificadora fictícia. Em seguida, foram criadas as requisições de certificados. A partir disso, as requisições foram assinadas através da CA criada anteriormente. Finalmente, o arquivo foi convertido para um formato suportado pela aplicação.

Também foi necessário implementar um *applet* (Figura 5.4), responsável pelo gerenciamento do certificado digital e da chave privada dos usuários, criação de mensagens SOAP e assinatura das mesmas. Este *applet* contém os seguintes campos de configuração:

- **formato** - representa o formato do certificado digital;

- **keystore password** - *password* de acesso para o *keystore*;
- **keystore alias** - é o identificador do certificado digital localizado no *keystore*;
- **alias password** - *password* de acesso para o certificado digital; e
- **localizacao** - caminho da localização do certificado digital.

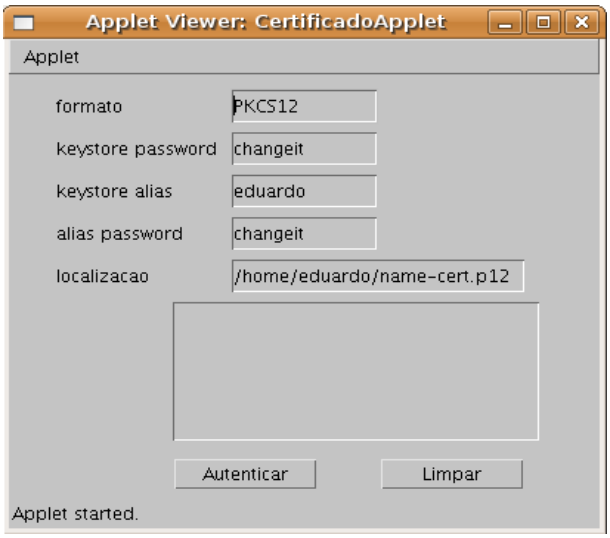

Figura 5.4: *Applet* para Certificado Digital.

As mensagens SOAP com certificado digital da mesma forma que as mensagens criadas para usuário/senha utilizam também o WSS4J, adicionando ao *header* do SOAP de requisição o certificado digital e a assinatura dessa mensagem (Figura 5.5).

```
<Signature ID?>
        <SignedInfo>
            <CanonicalizationMethod/>
            <SignatureMethod/>
            <Reference URI?>
                (<Transforms>)?
                <DigestMethod>
                <DigestValue>
            </Reference>+
        </SignedInfo>
        <SignatureValue>
        (<KeyInfo>)?
        (<Object ID?>)*
</Signature>
```

Figura 5.5: Componentes da assinatura XML.

Os elementos *SignedInfo* e *SignatureValue* são obrigatórios na estrutura dessa assinatura XML. Já os elementos *KeyInfo* e *Object*, bem como o atributo *Id*, são opcionais.

O elemento *SignedInfo* além de obrigatório, é responsável por manter os dados para configuração de todo o processo de assinatura de um documento com o padrão XML *signature*. Os filhos deste elemento contêm toda a informação sobre o que foi assinado e como foi o processamento dessa assinatura. O algoritmo da assinatura se aplica a este elemento e a todos os nós filhos para gerar a assinatura.

O elemento *CanonicalizationMethod* especifica o algoritmo que foi utilizado no *SignedInfo* para canonicalizar o documento XML, isto é, normalizar o documento antes de assiná-lo.

O elemento *SignatureMethod* especifica o algoritmo de assinatura digital utilizado para assinar o documento que contém a assinatura.

O elemento *Reference* especifica os dados que serão assinados e como isso deve acontecer antes do *hashing*. O atributo URI, identifica os dados que serão assinados. Neste elemento também é definido o algoritmo de hash e o valor do hash. Atributo *Id* é de uso opcional.

Através do elemento *SignedInfo* é possível especificar: os algoritmos que devem ser utilizados para canonicalizar o documento (normalizá-lo com referência ao padrão XML), gerar o hash e assinar os dados; e referenciar os dados que deverão ser assinados no processo.

O elemento *SignatureValue* contém o valor da assinatura. Esse valor, na maioria das vezes, corresponde ao valor do hash cifrado por um algoritmo definido no elemento *DigestMethod* filho do elemento *Reference* que foi explicado anteriormente.

O elemento *KeyInfo*, que é opcional, é utilizado para enviar, junto com a assinatura, a chave pública para que o destinatário possa validar a assinatura do documento. Este elemento contém informações a respeito da chave; dentre as quais podem estar: a chave, o nome da chave, dados de certificados digitais e informações de gerenciamento da chave pública, dados sobre o contrato da chave, entre outros. Quando esse elemento não aparece na assinatura, subentende-se que o destinatário já contém a chave pública do remetente do documento e as informações sobre o algoritmo utilizado para geração da mesma.

Esta mensagem SOAP, é enviada a partir do navegador *web* do usuário, através do objeto *ObjectOutputStream* do pacote *java.io*, para um *servlet* localizado no portal que é

responsável por encaminhar essa mensagem SOAP para o serviço de autenticação.

Ao receber esta requisição, o serviço de autenticação verifica o certificado que recebeu e processa as informações do cabeçalho, extraindo o certificado digital, em seguida pega a chave pública desse certificado em seu *keystore*. Para trabalhar com o *keystore* é utilizada a classe *Merlin* do pacote *org.apache.ws.security.components.crypto* e para isso é necessário configurar o arquivo crypto.properties (Figura 5.6).

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto.Merlin
org.apache.ws.security.crypto.merlin.keystore.type=pkcs12
org.apache.ws.security.crypto.merlin.keystore.password=security
org.apache.ws.security.crypto.merlin.keystore.alias=16c73ab6-b892-458f-abf5-2f875f74882e
org.apache.ws.security.crypto.merlin.alias.password=security
org.apache.ws.security.crypto.merlin.file=keys/x509.PFX.MSFT
```

Figura 5.6: Arquivo crypto.properties

A partir disso, se verifica a autenticidade da assinatura passada na mensagem SOAP através da *tag <Signature>*. Para fazer isso, é necessário recalcular o valor na *tag <DigestValue>* usando o algoritmo especificado na *tag <SignatureMethod>* e usar a chave pública pesquisada anteriormente no *keystore* (arquivo onde ficam armazenadas as chaves e certificados) do Serviço de Autenticação. Verificando se o valor da *tag <SignatureValue>* é igual ao valor contido na *tag <DigestValue>*, se forem valores iguais a assinatura é validada e o usuário é autenticado.

5.1.3 Impressão Digital

Primeiramente foram procurados aparelhos leitores de impressão digital do tipo ótico, mas a escolha deu-se pelo modelo da Microsoft - *Fingerprint Reader* USB por já tê-lo disponível no Centro de Processamento de Dados e por não haver recursos para a compra de um novo, sendo que dessa forma tem-se a desvantagem que o usuário fica limitado à plataforma Windows.

Como o usuário deve passar as informações da digital através de um navegador *web*, foi necessário criar um *applet*, que só funciona para o leitor Microsoft, responsável pela comunicação entre o navegador e o leitor, recebendo a imagem que o aparelho envia. Por esse motivo, foi necessário utilizar um Kit de Desenvolvimento de Software (SDK) responsável pela captura do *template* da imagem recebida pelo *applet*, este SDK, *GrFinger*

Biometric SDK *Trial* da *Griaule*, tem um método responsável por converter a imagem para um objeto do tipo *template* contendo um atributo do tipo vetor de bytes.

Em seguida, esses dados são enviados, através do objeto *ObjectOutputStream*, para um *servlet* localizado no portal que é responsável por criar uma mensagem SOAP adicionando ao corpo da mensagem os dados recebidos. Para manter seguro as informações do que será enviado é necessário criptografar o corpo dessa mensagem para finalmente invocar o serviço de autenticação. Aqui novamente se usa a API WSS4J, que tem um método para criptografar o corpo do envelope SOAP conforme mostra a figura 5.7, expressado numa forma curta, o elemento *EncryptedData* tem a seguinte estrutura (onde "?"significa zero ou uma ocorrência; "+"significa uma ou mais ocorrências; "*"significa zero ou muitas ocorrências; e um elemento tag vazio (<elemento/>) significa que o elemento será vazio):

```
<EncryptedData Id? Type? MimeType? Encoding?>
        <EncryptionMethod/>?
        <ds:KeyInfo>
            <EncryptedKey>?
            <AgreementMethod>?
            <ds:KeyName>?
            <ds:RetrievalMethod>?
            <ds:*>?
        </ds:KeyInfo>?
        <CipherData>
            <CipherValue>?
            <CipherReference URI?>?
        </CipherData>
        <EncryptionProperties>?
</EncryptedData>
```

Figura 5.7: Componentes da criptografia XML (IMAMURA; DILLAWAY; SIMON, 2002).

Os elementos mais representativos de <*EncryptedData*> são: <*EncryptionMethod*> e <*CipherData*>. <*EncryptionMethod*> aponta para algoritmo utilizado na criptografia. <*CipherData*> contém os dados criptografados ou um ponteiro para as informações criptografadas.

A mensagem SOAP recebida pelo serviço de autenticação será descriptografada com o auxílio da API WSS4J, o conteúdo do corpo da mensagem é usado na realização de uma consulta ao banco de dados para verificar quem está tentando se autenticar. Se for encontrado um *template* que casa, o serviço de autenticação verifica quem é o usuário e retorna uma resposta ao *servlet* dizendo que o usuário está autenticado e qual o seu *id*.

5.2 Portal

A implementação do portal foi feita usando a tecnologia JEE (*Java Enterprise Edition*), onde temos a página inicial (Figura 5.8), a partir da qual o usuário pode efetuar sua autenticação através de usuário/senha ou escolher outro método.

Figura 5.8: Página inicial do portal.

Para isso foi codificado um *servlet* para cada mecanismo, que recebe os dados enviados pelo usuário e chama o serviço de autenticação, se o usuário não for validado pelo serviço, o *servlet* o direciona para uma página contendo uma mensagem informando que os dados passados são inválidos.

Caso seja autenticado, é colocado na sessão que o usuário foi autenticado, seu identificador e o tipo de mecanismo usado. Em seguida, encaminhado para uma página que lista as aplicações, mas antes passa por dois filtros, um responsável por verificar se o usuário está autenticado e outro pela listagem de aplicações. Este segundo consulta o identificador desse usuário na sessão e se comunica com o CIBAC, que é responsável por buscar as aplicações que este usuário pode acessar (SACCHET, 2007).

Com a listagem das aplicações o filtro se comunica com uma base de dados buscando informações sobre as aplicações que são: id, nome e URL. Coloca na sessão um mapa que contém um objeto contendo esses dados, ao chegar na página requisitada, é consultado na sessão esse objeto, a partir do qual é construída a página com a listagem das aplicações (Figura 5.9).

Figura 5.9: Página do portal de listagem de aplicações.

A partir deste momento, o usuário ao acessar determinada aplicação passa pelo *servlet de ações*. Este *servlet* verifica se o usuário tem autorização de acesso, se tiver, recupera quais ações o mesmo pode executar e permite o acesso a aplicação solicitada, caso contrário, retorna para a página com a listagem das aplicações informando ao usuário que não está autorizado. Isto significa que o mecanismo de autenticação pode ser de nível inferior ao exigido. A partir deste momento, o usuário deve elevar seu nível de confiança.

Antes do usuário acessar a aplicação, ele passa pelos filtros de login e autorização, agora na página da aplicação, a cada ação que o usuário efetuar está sujeito aos filtros de login, autorização e ações. O filtro de ações é responsável por controlar todas as ações que o usuário executa dentro desta aplicação, verifica se aquela ação que o usuário está tentando executar é permitida. Se for permitido é garantido o acesso, caso contrário, retorna para a página da aplicação corrente informando ao usuário que não está autorizado. O usuário deve então, elevar seu nível de confiança para efetuar a ação que lhe foi negada.

5.3 Estudo de Caso

A listagem das aplicações no portal serviu como estudo de caso para testar autenticação sob-demanda para acesso a estas aplicações e ações que poderão ser feitas em cada

uma delas.

Podemos ter um cenário onde a aplicação Realização de Exames pode ser acessada somente com autenticação através de impressão digital apenas por médicos e enfermeiros, enquanto a aplicação Resultado dos Exames através de certificado digital por médicos e a Consulta Laudos Liberados utilizando usuário/senha apenas por médicos e pacientes.

Após ter acessado a aplicação Realização de Exames o usuário poderá fazer as ações criar, visualizar e alterar, mas a ação excluir, pode ser realizada apenas por um médico e, caso necessário, elevando seu nível de confiança para o exigido, neste caso certificado digital.

Para este caso, foram definidas políticas em arquivos XACML que controlam este acesso. Um exemplo das políticas é o arquivo *01.xacml* referente a aplicação Realização Exames, onde no campo *Target* definimos os sujeitos que tem função de médico ou enfermeiro e função de impressão digital ou certificado digital. Se essa política for aplicável, terão direito de executar as ações definidas no campo *Obligations*: criar, visualizar e alterar.

```
<Policy ...>
        <Target>
        <Subjects><Subject> ...
                <AttributeValue DataType=".../XMLSchema#string».*# Medico #.* | .*# Enfermeiro #.*</AttributeValue> ...
          </Subject><Subject> ...
                <AttributeValue DataType=".../XMLSchema#string».*# Impressao #.* | .*# Certificado #.*</AttributeValue> ...
        </Subject></Subjects>
        <Resources><Resource> ...
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string»aplicacao</AttributeValue> ...
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string»1</AttributeValue> ...
        </Resource></Resources> ...
        </Target> ...
        <Obligations> ...
              <AttributeAssignment AttributeId="acoes"...>criar visualizar alterar</AttributeAssignment> ...
        </Obligations>
</Policy>
```

Figura 5.10: Arquivo *01.xacml*.

Um exemplo de autenticação sob-demanda neste cenário, é quando um médico acessa o portal através de usuário/senha para acessar a aplicação Consulta Laudos Liberados, em seguida ele decide acessar a aplicação Realização Exames para cadastrar um novo exame, mas para isso deverá efetuar a autenticação através de impressão digital que é o mínimo exigido, conforme o arquivo de políticas *01.xacml*.

Se por algum motivo o médico decidir excluir algum exame, ele deve novamente elevar seu nível de confiança, se autenticando com seu certificado digital. Como descrito anteriormente, apenas o médico poderá efetuar esta ação, a qual será negada para um enfermeiro mesmo que este tenha se autenticado através de certificado digital, esta política está definida no arquivo *02.xacml*.

```
<Policy ...>
        <Target>
        <Subjects><Subject> ...
                <AttributeValue DataType=".../XMLSchema#string».*# Medico #.*</AttributeValue> ...
           </Subject><Subject> ...
                <AttributeValue DataType=".../XMLSchema#string».*# Certificado #.*</AttributeValue> ...
        </Subject></Subjects>
        <Resources><Resource> ...
             <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string»aplicacao</AttributeValue> ...
             <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string»1</AttributeValue> ...
        </Resource></Resources> ...
        </Target> ...
        <Obligations> ...
             <AttributeAssignment AttributeId="acoes"...>excluir</AttributeAssignment> ...
        </Obligations>
</Policy>
```

Figura 5.11: Arquivo *02.xacml*.

6 CONCLUSÃO

Inicialmente, o trabalho levou ao estudo das soluções de autenticação existentes e quais os mecanismos presentes em cada uma delas. Em seguida, foi estudado o conceito de *web services*, as tecnologias envolvidas para a sua criação e os protocolos de segurança envolvidos na troca de mensagens SOAP.

A partir disso foi implementado um serviço que permite a autenticação através de usuário/senha, impressão digital e certificado digital, o qual é desacoplado e atua sobre o protocolo HTTP através da troca de mensagens SOAP. Além disso, com XML sendo um padrão internacional aberto, para a troca de dados, é garantido o funcionamento em diferentes plataformas. Esta troca de mensagens é feita de forma segura utilizando o protocolo WS-Security que trabalha com assinatura XML e criptografia XML. Com isso podemos garantir a integridade e confidencialidade das mensagens.

A implementação da aplicação Realização Exames para o portal, serviu como caso de uso para o serviço de autenticação. Para isso, também foram criadas políticas de acesso, que definiram os sujeitos, objetos, regras e ações. Enfim, essas políticas determinaram o mecanismo necessário para acessar cada aplicação e as ações que poderiam ser feitas em cada uma delas, servindo para verificar o uso de autenticação sob-demanda.

Outra dificuldade encontrada foi migrar a aplicação Realização Exames para o ambiente *web*, pelo fato do modelo relacional já estar pronto, sendo que eram totalmente desconhecidos os relacionamentos. Com a criação do portal, tem-se acesso a conteúdos digitais que agora serão controlados pelo serviço de autenticação criado, e como trabalhos futuros podemos ter a integração de novas aplicações ao portal e também um sistema de *proxy* para o portal, de maneira que as aplicações possam estar hospedadas em diferentes servidores.

Também podem ser incorporados novos mecanismos de autenticação ao serviço já

existente, sendo que o serviço criado serve de base teórica e prática para a incorporação de soluções mais robustas e confiáveis. Ainda, pode ser feito um trabalho que tente encontrar vulnerabilidades e falhas neste serviço de autenticação. O serviço de autenticação também pode ser desenvolvido com suporte à outras tecnologias.

REFERÊNCIAS

BARTEL, M.; BOYER, J.; FOX, B.; LAMACCHIA, B.; SIMON, E. **XML-Signature Syntax and Processing**. Disponível em: <http://www.w3.org/TR/xmldsig-core/>. Acesso em: 13 fev. 2007.

BOOTH, D.; HAAS, H.; MCCABE, F.; NEWCOMER, E.; CHAMPION, M. **Web Services Architecture**. Disponível em: <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>. Acesso em: 13 fev. 2007.

BOX, D.; EHNEBUSKE, D.; KAKIVAYA, G.; LAYMAN, A.; MENDELSOHN, N.; NIELSEN, H. F.; THATTE, S.; WINER, D. **Simple Object Access Protocol (SOAP) Version 1.1**. Disponível em: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>. Acesso em: 13 fev. 2007.

BRASILEIRO, G. **e-PING Padrões de Interoperabilidade de Governo Eletrônico**. Disponível em: < http://www.governoeletronico.gov.br/governoeletronico/>. Acesso em: 13 fev. 2007.

CANTOR, S.; KEMP, J.; PHILPOTT, R.; MALER, E. **Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0**. Disponível em: <http://docs.oasis-open.org/security/saml/v2.0/>. Acesso em: 13 fev. 2007.

CARVALHO FERREIRA, L. de. **Segurança de Web Services**. Disponível em: <www.ciphersec.com.br/>. Acesso em: 13 fev. 2007.

CHRISTENSEN, E.; CURBERA, F.; MEREDITH, G.; WEERAWARANA, S. **Web Services Description Language (WSDL)**. Disponível em: <http://www.w3.org/TR/wsdl/>. Acesso em: 13 fev. 2007.

COSTA, C. G. A. da. **Desenvolvimento e Avaliação Tecnológica de um Sistema de Prontuário Eletrônico do Paciente, Baseado nos Paradigmas da World Wide Web e da Engenharia de Software**. 2001. Dissertação (Mestrado) — Universidade Estadual de Campinas.

DIGITAL, P. C. **O que é Certificado Digital?** Disponível em: <https://www.oficioeletronico.com.br/>. Acesso em: 13 fev. 2007.

EHNEBUSKE, D.; ROGERS, D.; RIEGEN, C. von. **UDDI Version 2.03 Data Structure Reference**. Disponível em: <http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm/>. Acesso em: 13 fev. 2007.

FIORESE, M. **Uma Proposta de Autenticação de Usuários para Ensino a Distância**. Disponível em: <http://penta.ufrgs.br/pesquisa/fiorese/>. Acesso em: 13 fev. 2007.

FORD, W.; BAKER, P. H.; FOX, B.; DILLAWAY, B.; LAMACCHIA, B.; EPSTEIN, J.; LAPP, J. **XML Key Management Specification (XKMS)**. Disponível em: <http://www.w3.org/TR/xkms/>. Acesso em: 13 fev. 2007.

FOUNDATION, T. A. S. **Apache WSS4J**. Disponível em: <http://ws.apache.org/wss4j/>. Acesso em: 13 fev. 2007.

FOUNDATION, T. A. S. **Apache Tomcat**. Disponível em: <http://tomcat.apache.org/index.html/>. Acesso em: 13 fev. 2007.

FOUNDATION, T. A. S. **Apache Axis2 Java**. Disponível em: <http://ws.apache.org/axis2/>. Acesso em: 13 fev. 2007.

GONZAGA, D. C. **Certificação Digital**. Disponível em: <http://br-linux.org/tutoriais/002209.html/>. Acesso em: 13 fev. 2007.

GRAHAM, S.; DAVIS, D.; SIMEONOV, S.; DANIELS, G.; BRITTENHAM, P.; NAKAMURA, Y.; FREMANTLE, P.; KOENIG, D.; ZENTNER, C. **Building Web Services with Java**. 2ª.ed. [S.l.]: SAMS, 2004.

GUDGIN, M.; HADLEY, M.; MENDELSOHN, N.; MOREAU, J. J.; NIELSEN, H. F. **Simple Object Access Protocol**. Disponível em: <http://www.w3.org/TR/soap/>. Acesso em: 13 fev. 2007.

HANSEN, R. P.; SANTOS, C.; PINTO, S. C.; LANIUS, G.; MANSSEN, F. **Web Services**: An Architectural Overview. Disponível em: <http://www.inf.unisinos.br/ webcomposej/links.html/>. Acesso em: 13 fev. 2007.

IMAMURA, T.; DILLAWAY, B.; SIMON, E. **XML Encryption Syntax and Processing**. Disponível em: <http://www.w3.org/TR/xmlenc-core/>. Acesso em: 13 fev. 2007.

INSTITUTE, A. N. S. **Information technology - Finger Minutiae Format for Data Interchange**. Disponível em: <http://www.techstreet.com/>. Acesso em: 13 fev. 2007.

LIMA DANTAS, G. F. de. **Identificação biométrica**: sistemas biométricos de identificação pela imagem facial. Disponível em: <http://www.peritocriminal.com.br/biometria.htm/>. Acesso em: 13 fev. 2007.

LTD, E. G. P. **Cell-ID**. Disponível em: <http://www.expertron.co.za/index.php?section=11/>. Acesso em: 13 fev. 2007.

PAOLI, J.; YERGEAU, F.; BRAY, T.; MCQUEEN, C. M. S.; MALER, E. **Extensible Markup Language (XML)**. Disponível em: <http://www.w3.org/xml/>. Acesso em: 13 fev. 2007.

PEREIRA, L. P. C. **Mapeamento de Imagens Binárias**: Um Estudo sobre Biometria da Mão. Disponível em: <www.cci.unama.br/margalho/portaltcc/tcc2003/d2817.pdf/>. Acesso em: 13 fev. 2007.

RHEINHEIMER, L. R. **WSAgent**: Um Agente Baseado em Web Services para Promover a Interoperabilidade entre Sistemas Heterogêneos no Domínio da Saúde. Disponível em: <http://www.artigocientifico.com.br/uploads/>. Acesso em: 13 fev. 2007.

SACCHET, L. **Estudo e Implementação de Autorização no Acesso para o Portal do HUSM Utilizando o CIBAC**. Monografia (Bacharel em Ciência da Computação) - Universidade Federal de Santa Maria, Santa Maria, 2007.

SANTOS ROMAGNOLI, G. dos. **Biometria**: Você é sua Senha. Disponível em: <http://www.serpro.gov.br/publicacao/tematec/2002/ttec61/>. Acesso em: 13 fev. 2007.

SOUZA, C. A. G. de. **Implementação do CIBAC no SIE usando SOA**. Monografia (Bacharel em Ciência da Computação) - Universidade Federal de Santa Maria, Santa Maria, 2006.

STRUCTURED INFORMATION STANDARDS, O. for the Advancement of. **OASIS Web Services Security (WSS) TC**. Disponível em: <http://www.oasis-open.org/>. Acesso em: 13 fev. 2007.

TANENBAUM, A. S. **Redes de Computadores**. 4ª.ed. [S.l.]: Campus / Elsevier, 2003.

VERISIGN. **Segurança Para Web Services**. Disponível em: <http://www.verisign.com/wss/>. Acesso em: 13 fev. 2007.

WEAVER, A. C. **Advancing Cyber Security with .NET**. Disponível em: <http://www.cs.virginia.edu/ acw/security/>. Acesso em: 13 fev. 2007.

WEAVER, A. C.; III, S. J. D.; SNYDER, A. M.; DYKE, J. V.; HU, J.; CHEN, X.; MULHOLLAND, T.; MARSHALL, A. **Federated, Secure Trust Networks for Distributed Healthcare IT Services**. Disponível em: <http://www.cs.virginia.edu/>. Acesso em: 13 fev. 2007.