UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS DE MELO SILVA

UMA NOVA ARQUITETURA PARA COMPARTILHAMENTO E ARMAZENAMENTO SEGURO DE REGISTROS DE SAÚDE NA NUVEM UTILIZANDO ATRIBUTOS DE IDENTIDADE FEDERADA

Belém-PA
2015

Lucas de Melo Silva

UMA NOVA ARQUITETURA PARA COMPARTILHAMENTO E ARMAZENAMENTO SEGURO DE REGISTROS DE SAÚDE NA NUVEM UTILIZANDO ATRIBUTOS DE IDENTIDADE FEDERADA

Dissertação de Mestrado apresentada para a obtenção do grau de Mestre em Ciência da Computação no Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará.
Área de Concentração: Redes de Computadores.
Orientador: Prof. Dr. Josivaldo de Souza Araújo

Belém-PA
2015

Lucas de Melo Silva

UMA NOVA ARQUITETURA PARA COMPARTILHAMENTO E ARMAZENAMENTO SEGURO DE REGISTROS DE SAÚDE NA NUVEM UTILIZANDO ATRIBUTOS DE IDENTIDADE FEDERADA

Dissertação de Mestrado apresentada para a obtenção do grau de Mestre em Ciência da Computação no Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará.

Data da Aprovação: Belém-PA, ___/___/___

Banca Examinadora:

_____

Prof. Dr. Josivaldo de Souza Araújo
UFPA/ICEN/Faculdade de Computação/PPGCC (Orientador)

_____

Prof. Dr. Claudomiro de Souza de Sales Júnior
UFPA/ICEN/Faculdade de Computação/PPGCC (Membro)

_____

Prof. Dr. Roberto Samarone dos Santos Araújo
UFPA/ICEN/Faculdade de Computação (Membro Externo)

_____

Profa. Dr. Regiane Silva Kawasaki Frances
UFPA/ICEN/Faculdade de Computação (Membro Externo)

AGRADECIMENTOS

Primeiramente devo gratidão a Deus, pois é quem tem me direcionado e auxiliado para a realização de seus desejos em minha vida, incluindo esse trabalho. É unicamente por sua graça e amor que tenho desfrutado de inúmeras bênçãos e mais essa conquista.

A minha amada esposa Priscila Cavalcante, uma mulher admirável e única, que também tem me feito desfrutar de seu amor, apoio e virtudes, fazendo com que as minhas dificuldades e desafios sejam mais superáveis.

Aos meus pais e irmãs, que representam mais uma dádiva de Deus e que me apoiaram e investiram em meus desejos e sonhos, realizando sacrifícios e comprometimentos para que isso fosse possível.

Às famílias que me acolheram em Belém ainda antes da graduação, enquanto minha família estava longe. Elas também me apoiaram e realizaram seus próprios sacrifícios para me disponibilizar conforto e segurança. Em especial às famílias de Márcio e Elizabeth Andrade, Paulo e Quézia Oliveira.

Ao meu orientador, Prof. Dr. Josivaldo de Souza Araújo pela orientação e revisão desse trabalho. Como também ao Prof. Dr. Roberto Araújo Samarone, pelos ensinamentos e competência, além das oportunidades de pesquisa e aprimoramento acadêmico que me proporcionou.

Aos meus amigos com quem há anos venho compartilhando dificuldades, provações e júbilo. Em especial ao Felipe Leite, que dês de a graduação tem se disposto a avaliar e auxiliar minha pesquisa.

"O temor do Senhor é o princípio da sabedoria;
e o conhecimento do Santo é o entendimento."

Provérbios 9.10

RESUMO

A computação em nuvem é um serviço tecnológico que se tornou tendência devido a sua facilidade e benefícios de utilização. Neste contexto, muitos são os serviços que podem compor a sua estrutura. Entre eles está o de armazenamento, que é muito útil em razão de grande quantidade de dados gerados nas mais variadas áreas do conhecimento. Neste sentido, o armazenamento de Registros Pessoais de Saúde (*Personal Health Records* - PHR) e dos Registos Eletrônicos de Saúde (*Electronic Health Records* - EHR) em nuvem torna-se essencial para uma melhor disponibilidade e provisionamento sob demanda, não apenas para os profissionais da saúde, como médicos e agentes de saúde, mas também para todos aqueles que necessitam acessar esses tipos de dados, como familiares e amigos. No entanto, esta tecnologia aumenta o risco de vazamento de dados sensíveis de saúde.

Apesar de fornecer vários benefícios, a fragilidade da segurança da utilização de registros de saúde em nuvem é um fator que pode inviabilizar tal serviço em saúde eletrônica. Para que esse problema não impeça o armazenamento terceirizado de EHR e PHR, este trabalho apresenta uma nova arquitetura que faz proveito de federação de identidade para viabilizar armazenamento e compartilhamento seguros de registros de saúde no ambiente de nuvem.

A arquitetura proposta utiliza criptografia baseada em atributos para garantir a segurança de registros. Para isso, o mecanismo faz uso de federação de identidade integrada ao protocolo criptográfico como parte do processo de controle de acesso e autenticação.

PALAVRAS-CHAVE: Computação em nuvem, segurança, privacidade, registros eletrônicos de saúde, identidade federada, criptografia baseada em atributos.

ABSTRACT

Cloud computing is a technological service that has become a trend due to its ease of use and many benefits. In this context, there are many services that can comprise its structure, including data storage, which is very useful due to the large amount of data that is generated in various areas of knowledge. In this sense, the storage of Personal Health Records (PHR) and Electronic Health Records (EHR) in a cloud becomes essential. It allows for a better availability and on demand provisioning, not only for health professionals, such as doctors and health workers, but also to all those who need access to these data types, such as family and friends. However, this technology increases the risk of leakage of sensitive health data.

Although it provides many benefits, the fragility of safety in health records storage in the cloud is a factor that can derail such integration. As a means of solving this problem so that it does not impede the outsourced storage of EHR and PHR, the work here presents a new architecture that takes advantage of identity federation to enable secure storage and sharing of health records in the cloud environment.

The proposed architecture uses attribute-based encryption to ensure the safety of records. For this, the mechanism makes use of identity federation integrated with the cryptographic protocol as part of the access and authentication control process.

KEYWORDS: cloud computing, security, privacy, health records, federated identity, attribute-based encryption

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

AA	***Attribute Autority***
ABE	***Attribute Based Encryption***
CP-ABE	***Ciphertext Attribute Based Encryption***
e-health	***Eletronic Health***
EHR	***Eletronic Health Record***
FIM	***Federal Identity Management***
IdP	***Identity Provider***
MA-ABE	***Multiple Authority Attribute Based Encryption***
PHR	***Personal Health Record***
PK	***Public Key***
RS	**Registro de Saúde**
SAML	***Security Assertion Markup Language***
SK	***Secret Key***
SP	***Service Provider***
TC	**Texto Cifrado**

SUMÁRIO

1 INTRODUÇÃO

Este capítulo apresenta inicialmente a contextualização do trabalho necessária para o entendimento da motivação que levou à sua elaboração. Em seguida são apresentados os objetivos e a metodologia definida para a execução do trabalho, e finalmente é apresentada a estrutura desta dissertação.

1.1 Contexto do Trabalho

A computação em nuvem é um paradigma que se caracteriza por oferecer recursos pela rede de forma ubíqua, transparente e sob demanda com um mínimo de esforço de gestão por parte de seus usuários (MELL e GRANCE, 2010), diminuindo a complexidade na provisão desses recursos e os seus custos.

A implantação e manutenção de grandes infraestruturas de armazenamento de dados são caras e complexas. Como alternativa, o serviço de armazenamento de dados pode ser terceirizado para um serviço de nuvem de armazenamento. Isso resulta em maior facilidade e, principalmente, economia, fazendo dessa característica um dos fatores que tornou a nuvem uma tendência e um dos serviços tecnológicos que mais cresceram nos últimos anos (BHADAURIA e SANYAL, 2012).

Esse aumento é justificado pelas grandes quantidades de dados que são gerados a cada momento. Informações como dados pessoais, registros profissionais, de saúde, impostos, financeiros, resultados de pesquisas, enfim, diversas informações que precisam, de alguma forma, ser acessadas com segurança e por dispositivos ou sistemas próprios, para que possam ser consultadas sempre que necessário.

Uma das áreas que pode se beneficiar bastante da computação em nuvem, principalmente pela redução dos custos e o aumento da disponibilidade (LOHR, SADEGHI e WINANDY, 2010), é a área de saúde, através dos ambientes de saúde eletrônica (*e-health*).

Este é o caso do armazenamento de Registros Eletrônicos de Saúde (*Eletronic Health Record* - EHR). Os EHRs são uma alternativa para os registros tradicionais em papel e guardam dados médicos de um paciente, como resultados laboratoriais, receitas médicas, diagnósticos e histórico de saúde.

Devido aos EHRs serem digitais, facilitam a gerência, atualização, compartilhamento e acesso aos dados de pacientes. Cabe aos médicos e outros profissionais de saúde a gerência de cada EHR, como também controlar o acesso e edição de seus conteúdos (LOHR, SADEGHI e WINANDY, 2010).

Diferentemente dos EHRs, os Registros Pessoais de Saúde (*Personal Health Record* - PHR) são centrados no paciente. Isto significa que os próprios pacientes podem inserir informações em seus registros, bem como importar do EHR. O seu uso tem por objetivo fornecer aos pacientes uma melhor visão do seu próprio histórico de saúde, centralizar as informações ao consultar com médicos diferentes e compartilhar informações com a família e amigos. Em PHR, o paciente decide quem possui acesso e pode editar informações no registro (AVANCHA, BAXI e KOTZ, 2012).

Os PHRs também podem ser beneficiados pelo uso de armazenamento de dados na nuvem. De fato, há sistemas comerciais surgindo para oferecer essa terceirização no contexto de *e-health*, como por exemplo, *Microsoft Health Vault* (2015). Com esse tipo de registro se almeja a melhoria geral na qualidade de saúde de indivíduos, de forma contínua e mesmo quando não estando sobre cuidados médicos no momento.

Para autenticação de identidade, a maioria dos sistemas de computação em nuvem faz uso da abordagem tradicional onde são requisitadas credenciais de usuários, como por exemplo, nome de usuário (*login*) e senha. Com isso, além de serem provedores de serviços, também implementam o gerenciamento de usuários e suas respectivas credenciais. Como por exemplo, para utilizar a nuvem Dropbox (2015), um usuário inicialmente se cadastra através de um formulário onde informa email e senha, para futuramente poder se autenticar e utilizar os serviços de armazenamento.

Essa gerência exige atividades adicionais por parte do provedor da nuvem, e que não estão relacionadas diretamente com a provisão do serviço fim. Além disso, usuários passam a ter múltiplas credenciais para acessarem múltiplos provedores de serviços, o que implica em maior esforço para memorizar diferentes senhas e *logins*.

Uma alternativa para essa abordagem de autenticação é o uso de uma federação de entidades (*Federal Identity Management*- FIM) que se unem com o objetivo de prover e gerenciar identidades, isto é, uma Federação de Identidade. Neste modelo, existem Provedores de Identidade (*Identidy Provider* - IdP) e Provedores de Serviço (*Service Provider* - SP). Em uma federação, um usuário possui sua conta cadastrada e gerenciada por um IdP, que é o responsável por autenticá-lo. Por outro lado, os SPs confiam nas informações de autenticação fornecidas pelos IdPs, fazendo com que um usuário não necessite ter uma credencial para cada serviço que queira utilizar (AUTHENTICATION WORLD, 2015).

Assim, sistemas de gerência federada de identidade possibilitam a formação de redes colaborativas entre instituições que possuem algum tipo de afinidade e que desejam o compartilhamento de serviços entre seus usuários. Exemplos dessas redes são as federações acadêmicas, como a Comunidade Acadêmica Federada (CAFe, 2013) da Rede Nacional de Pesquisa (RNP) do Brasil e a *InCommon* (2013) da rede *Internet 2* dos EUA; e as redes federadas para governança eletrônica, como as iniciativas na Nova Zelândia, Austrália, Canada e Estados Unidos (OECD, 2011).

Um possível recurso desejável para essas federações é o de armazenamento de dados. Isso pode ser disponibilizado através de um SP de armazenamento em nuvem. Outro benefício da presença de múltiplos SPs de nuvem é que membros da federação não estão limitados a somente um serviço de nuvem.

Apesar do uso da nuvem agregar diversos benefícios, ainda existem desafios relativos à segurança dessa tecnologia. Um desses desafios refere-se à provisão de armazenamento seguro juntamente com proteção à privacidade dos usuários e suas informações de ataques por agentes maliciosos internos ou externos aos sistemas de computação em nuvem. Além desse tipo de ataque, os dados ainda estão sujeitos a acesso pelo próprio provedor terceirizado. Isso pode ocorrer para fins como de publicidade, mineração de dados e pesquisa comercial. Portanto, o próprio provedor é capaz de infringir contra a privacidade de usuários visando algum benefício próprio (LI *et al*, 2010; TSENG e CHEN, 2012).

Uma das funcionalidades mais utilizadas em computação em nuvem é o compartilhamento de dados (ZHU *et al*., 2013), como por exemplo, a troca de arquivos entre usuários. Em uma federação, esta funcionalidade é de alta relevância devido ser natural a esse ambiente a presença de parcerias e colaborações. Portanto, é fundamental que esse compartilhamento seja feito de forma segura.

Diante do exposto, este trabalho apresenta uma nova arquitetura para armazenamento e compartilhamento seguro de registros de saúde em nuvem em uma federação de identidade. Para isso, será utilizado um mecanismo composto por protocolo criptográfico, mas sem a necessidade de compartilhamento de senhas ou chaves entre usuários.

1.2 Motivação

A computação em nuvem surgiu para solucionar diversos problemas, entre os quais, o alto custo de se implantar e manter grandes conglomerados de hardware e software. Esse novo paradigma trouxe muitos benefícios para a área de Tecnologia da Informação (TI), passando a ser largamente utilizado. No entanto, o surgimento da computação em nuvem também trouxe desafios que, quando não devidamente tratados, podem tornar problemática a adoção dessa tecnologia (LIU *et al*, 2013)

Alguns desses desafios são: interoperabilidade, padronizações (de acesso, operação, dentre outros), questões jurídicas e de políticas organizacionais relacionadas à terceirização de serviços de TI, privacidade, segurança, monitoramento, auditoria e confiabilidade. Dentre esses, provavelmente o mais importante seja a segurança e de que forma a nuvem a provê (SABAHI, 2011), principalmente por ser um fator determinante para a sua adoção.

Uma pesquisa realizada pela empresa norte americana de consultoria e pesquisa tecnológica Gartner, em 2009, apontou que 70% dos entrevistados tinham receio quanto à adoção de serviços de computação em nuvem, principalmente por questões de privacidade e segurança. Essa porcentagem já se alterou e em uma pesquisa mais recente 50% afirmavam que estão confiantes em delegar informações de negócio para a nuvem, apesar de desafios de segurança ainda permanecerem como o maior obstáculo para sua adoção (SECURITY FOR BUSINESS INNOVATION COUNCIL, 2013; TAN e AIB, 2011).

Uma das maiores preocupações relacionadas à segurança em nuvem é o armazenamento de dados de consumidores (SAHA e DAS, 2012). Quando esses consumidores utilizam o armazenamento em nuvem, os dados deixam de estar apenas sobre o seu controle, passam também, ao controle do provedor da nuvem. Tais dados são armazenados em um local desconhecido pelo cliente, em um ou vários servidores que podem residir, até mesmo, em diferentes países.

Redes de identidade federada buscam criar uma colaboração de serviços entre diversas instituições. Por ter uma natureza de parceria entre grupos diferentes e muitas vezes em lugares geográficos diferentes, o armazenamento e compartilhamento de dados através de computação em nuvem é um serviço desejável para seus usuários. Com o ingresso da nuvem na forma de serviço de um SP em uma federação, tanto os benefícios quanto os riscos de segurança do uso da nuvem se tornam presentes na rede federada.

Em se tratando de uma federação de identidade para *e-health*, registros de saúde armazenados na nuvem possuem uma sensibilidade ainda maior que o comum por se tratarem de dados pessoais sensíveis. O compartilhamento também é essencial por ser natural e corriqueiro o uso dos registros por profissionais de saúde, amigos e familiares. Portanto, essas funcionalidades devem ser viáveis com a exigência e garantia de segurança e confidencialidade.

1.3 Justificativa

A área de assistência médica pode ser melhorada com o uso de serviços de tecnologia da informação, e isso impacta diretamente no cuidado dos pacientes. Os registros eletrônicos de saúde são uma aplicação da TI que tem como beneficiar pacientes e provedores de assistência médica. A fim de diminuir os custos com esses registros e potencializar seu uso e compartilhamento, é possível lançar mão ao armazenamento em nuvem.

A questão sensível do armazenamento de registros de saúde na nuvem é: como garantir o controle de acesso e proteção de informações de saúde? O uso indevido desses dados e o acesso ou divulgação para membros não autorizados pode ser catastrófico para o paciente. Como a computação em nuvem apresenta questões de desconfiança na sua provisão de segurança e privacidade de dados, o seu uso em *e-health* apresenta ainda maiores riscos, uma vez que essa também possui sua adoção refreada em decorrência de questões de segurança e privacidade (GRAVILOV e TRAJKOVIK, 2012).

Diante desse cenário, se vê como necessário uma forma de os riscos de segurança no armazenamento terceirizado serem mitigados a um nível satisfatório para a privacidade dos pacientes.

1.4 Objetivos

Este trabalho tem como objetivo geral propor uma nova arquitetura e mecanismo que permita o armazenamento e o compartilhamento seguro de arquivos em nuvens que atuam como provedores de serviço em uma federação de identidades para e-health.

Para atender ao objetivo geral, devem ser contemplados os seguintes objetivos específicos:

- Identificar protocolos criptográficos aplicáveis ao contexto de controle de acesso à registros de saúde;
- Identificar pontos de integração e adaptação do protocolo criptográfico em componentes de uma rede federada de identidade;
- Elaborar arquitetura de um mecanismo que integre as diversas entidades e interações necessárias para prover o armazenamento e compartilhamento seguro de dados em nuvem;
- Descrever papéis de usuário, tarefas e procedimentos para o processo sugerido;
- Analisar mecanismo em relação a sua segurança e passos para usuário;
- Desenvolver um protótipo como prova de conceito para o mecanismo proposto.

1.5 Metodologia do Trabalho

A realização deste trabalho ocorreu através das etapas descritas a seguir:

a) Etapa de Estudo Inicial

Nessa etapa foi realizada a análise de trabalhos relacionados ao armazenamento seguro de registro de saúde na nuvem. Buscou-se identificar na literatura quais as técnicas e propostas que já existiam para tal finalidade e suas limitações. Também buscou-se protocolos criptográficos, principalmente os que poderiam ter alguma aplicabilidade em um contexto de autenticação federada e *e-health*.

b) Etapa de Elaboração da nova Solução

Nessa etapa se realizou um estudo mais aprofundado dos protocolos de criptografia baseados em atributos e dos protocolos e padrões de autenticação federada e seu possível uso

relacionado à EHR e PHR. Com os conhecimentos obtidos desse estudo, foi feita a elaboração da arquitetura da nova solução, detalhando os componentes (protocolos, padrões, entidades necessárias e papeis envolvidos), operações e mensagens trocadas.

c) Etapa de Construção dos Protótipos

Para essa etapa se definiu inicialmente a arquitetura dos protótipos de prova de conceito com base na arquitetura criada na etapa anterior. Em seguida se deu o desenvolvimento dos protótipos, testes e análises sobre o seu uso prático. Os testes e análises buscaram apenas comprovar a viabilidade da proposta, em uma forma de prova de conceito.

d) Etapa de Documentação

Por fim, nessa etapa se deu a redação final da dissertação.

1.6 Trabalhos Relacionados

A FIM é considerada um passo necessário para federação da nuvem (CHADWICK *et al*., 2013). Por meio desta tecnologia, nuvens cooperam para permitir a migração de dados armazenados (FORMISANO *et al*., 2014) ou a partilha de recursos (por exemplo, quando uma nuvem está sobrecarregada). O trabalho aqui proposto tem uma abordagem diferente e integra FIM e nuvens com foco no (*Attribute Based Encryption*) ABE como apoio para armazenamento e compartilhamento seguro de dados. Apesar de existirem várias propostas que relacionam a nuvem com ABE (LEE, CHUNG e HWANG, 2013), poucos trabalhos empregam FIM e ABE.

Tassanaviboon e Gong (2011) apresentam um esquema de autorização que se assemelha ao uso de FIM. Essa proposta utiliza CP-ABE (ver subseção 2.3) para criptografar dados armazenados em nuvens e adapta o oAuth para usar tokens de delegação de acesso com base em ABE. Através desse esquema, um usuário concede a um SP (por exemplo, uma companhia de impressão) o acesso a recursos (fotos) em um SP de nuvem. A principal limitação desse mecanismo proposto é a necessidade de o usuário estar obrigatoriamente presente em todos os processos de autorização de acesso, tendo que se autenticar em uma autoridade e enviar um *token* de autorização para quem deseja acessar o recurso.

Quando há necessidade de compartilhar dados armazenados, os usuários são, dentro desse modelo, obrigados a repetir essas mesmas operações várias vezes, o que pode ser desgastante ou inviável dependendo do número de interessados. Além disso, em uma situação em que o dono do recurso não esteja presente, não é possível realizar o compartilhamento. Em contraste, a solução proposta nesta dissertação visa armazenar dados de saúde num SP e compartilhá-los entre vários usuários, sem a participação direta dos proprietários de arquivos no processo de requisição de acesso ao recurso compartilhado.

Outra proposta que usa ABE com uma estrutura similar a FIM é a de **Niwa, Kanaoka e Okamoto (2013)**. Os autores apresentam uma arquitetura que faz uso de uma infraestrutura para fornecer informações para serviços de criação de chaves criptográficas. Essas chaves podem ser de protocolos de criptografia funcional, como por exemplo, ABE, IBE ou baseada em tempo. As informações fornecidas dependem da criptografia funcional, podendo ser identidade de usuários e atributos, por exemplo.

O principal trunfo da proposta, segundo os autores, é ser um esquema genérico. Isto permite, por exemplo, uma entidade criar chaves de usuários com base na informação consultada a partir de tais infraestruturas. Da mesma forma que essa solução, a proposta desta dissertação também faz uso de um provedor de informações do usuário, neste caso a federação de identidade para fornecer atributos ao processo de criação de chaves e políticas para cifragem. No entanto, tal trabalho relacionado carece de um mecanismo de revogação de acesso e não permite a emissão de chaves em uma forma escalável e descentralizada. Além disso, não garante a confiança na infraestrutura de fornecimento de informações. Finalmente, a proposta não se ajusta a utilização de recursos em nuvem nem em um cenário de *e-health*.

Além destes dois trabalhos, a proposta desta dissertação está relacionada aos mecanismos que empregam ABE para garantir a segurança de registros de saúde armazenados na nuvem. **Li *et al.* (2013)** apresentam tal cenário e dividem o controle de acesso e gerenciamento de chaves em dois domínios. O primeiro é o domínio pessoal, em que os próprios pacientes administram identidade e atributos ABE e gerenciam chaves para quem desejam compartilhar seus registros. O segundo é domínio público, onde atributos para o ABE e o gerenciamento de chaves é feito pelas autoridades de atributo.

Infelizmente nessa proposta o domínio pessoal adiciona complexidade para os usuários. Eles são sobrecarregados com operações de responsabilidade de uma Autoridade de Atributos (AA) (por exemplo, criar e distribuir chaves). Já no esquema proposto por essa dissertação, IdPs interagem com AAs enquanto que os usuários continuam com total controle de seus

registros. Além disso, também é utilizada a FIM para garantir a confiança nos atributos de usuários informados para AAs, bem como isenta usuários de operações complexas do ABE.

Outro mecanismo relacionado foi criado por **Alshehri, Radziszowski e Raj (2012).** Esse mecanismo é simples e propõe a utilização de ABE para criptografar EHR armazenado na nuvem. Tal solução é suscetível ao problema de centralização de chave por utilizar uma única AA (ver Subseção 2.3). Ademais, não aborda a questão de comprovação de atributos e solicitação de criação de chave perante AA. Já na proposta dessa dissertação, de modo diferente, IdPs comprovam atributos de usuários e interagem com AAs para realizar solicitações de criação de chaves.

Nzanywayingoma e Huang (2012) apresentam um esquema que usa uma variação do CP-ABE. Em seu esquema, os pacientes agem como AAs, criptografam PHRs e gerenciam a criação e a distribuição de chaves. Embora tivessem a intenção de entregar mais controle sobre o sistema para o usuário, mais uma vez isso aumenta a complexidade para os pacientes, que recebem responsabilidades normalmente de uma AA. Além disso, é importante considerar que os pacientes podem não estar disponíveis ou serem incapazes de realizar essas operações (por exemplo, certos pacientes debilitados e idosos). Além de aliviar os pacientes desta complexidade, a proposta detalhada no Capítulo 3, permite maior disponibilidade e ainda mantém controle do usuário sobre seus registros, concomitantemente não há necessidade de esperar por pacientes para emitir chaves para usuários.

Embora existam várias pesquisas com aplicação de ABE em PHR e EHR no ambiente de nuvem, a maioria não aborda a prova de atributos do usuário diante de AAs. Isso reduz a segurança dessas soluções. Usando FIM para verificar e gerenciar os atributos de usuários, o mecanismo aqui proposto apresenta uma solução que tira proveito de sistemas padronizados (SAML e oAuth) e o protocolo ABE.

1.7 Estrutura da Dissertação

Este trabalho está estruturado em cinco capítulos. Neste Capítulo 1, foi apresentada uma contextualização das necessidades desta pesquisa, bem como, as motivações, objetivos, metodologias utilizadas, como também os trabalhos relacionados.

O Capítulo 2 descreve a fundamentação teórica sobre computação em nuvem, enfatizando as nuvens de armazenamento e sua segurança, bem como, os conceitos de registros de saúde em *e-health*. Também é abordada a criptografia baseada em atributos, além de federação de identidade.

O Capítulo 3 apresenta a nova arquitetura, sendo abordados os detalhes do mecanismo proposto por esse trabalho. Aqui se explica o fluxo de comunicações e operações em cada componente da arquitetura, assim como os papéis e responsabilidades das entidades e usuários. Além disso, são apresentadas variações e possíveis adaptações na arquitetura, destacando-se vantagens e desvantagens com essas modificações.

No Capítulo 4 dois protótipos conceituais são descritos. Também se destacam as limitações do protótipo desenvolvido, os testes aplicados, uma breve avaliação e lições aprendidas com a prototipagem.

Por fim, o Capítulo 5 apresenta as considerações finais, contribuições, limitações e trabalhos futuros desta dissertação.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica utilizada para a elaboração da nova arquitetura proposta. Aqui são apresentados os conceitos sobre computação em nuvem, além de uma breve descrição sobre seus aspectos de segurança e logo em seguida, é descrita a federação de identidade. Também é apresentada a criptografia baseada em atributos seguida de conceitos iniciais sobre registros eletrônicos em *e-health*. E, por fim, as considerações finais do capítulo.

2.1 Computação em Nuvem

O termo "computação em nuvem" surgiu no início da década de 1990, geralmente se referindo a grandes redes ATM (MAGGIANI, 2009), mas passou a ser utilizado para um nicho de sistemas específicos, com diferentes propostas de conceitos e definições.

Em decorrência da computação em nuvem fazer uso de vários recursos tecnológicos já consolidados, inicialmente não se deu tanta importância para esse paradigma, onde muitos acreditavam ser apenas uma nova forma de denominar tecnologias antigas. No entanto, apesar de similaridades com tecnologias anteriores e, mesmo com o uso de várias dessas tecnologias (como por exemplo, computação em cluster e grid), computação em nuvem é algo diferente e inovador (GUO, KUO e SAHAMA, 2012).

Esse paradigma consiste em consumidores não terem o custo de criar, manter e gerenciar grandes conglomerados de recursos computacionais. Assim, ela traz uma democratização da computação, com pequenas empresas e empreendimentos conseguindo acesso a recursos de alta capacidade, escaláveis e rapidamente provisionados. O uso de computação em nuvem tem sido frequentemente comparado à contratação de serviços de fornecimento de água e luz, justamente por ser sobre demanda e geralmente ser cobrada segundo o consumo (FOX *et al*, 2009; GOVERNMENT EXECUTIVE, 2014).

Dessa forma, desde o debate inicial acerca da natureza da computação em nuvem e após várias propostas e discussões sobre sua definição, esse termo se consolidou como referente a uma nova tecnologia, com características próprias e que tem crescido em uso e importância. Como resultado, têm-se uma revolução de TI, tanto para o meio corporativo, quanto para usuários finais de sistemas, aplicativos e programas.

A seguir destaca-se a definição e classificação da nuvem, seguida de uma breve análise sobre segurança nessa tecnologia que por sua vez é seguida de uma introdução à plataforma de criação de nuvens chamada OpenStack.

2.1.1 Definição e Classificação

Dentre as definições existentes, a proposta do Instituto Nacional de Padrões e Tecnologia (*National Institute of Standards and Technology* - NIST) do departamento de comércio dos EUA foi largamente aceita e é frequentemente citada (DURKEE, 2010; BHADAURIA, 2012; SAHA e DAS, 2012), se tornando a definição de fato.

O NIST propõe que computação em nuvem essencialmente é um modelo para permitir acesso na rede, de forma ubíqua, conveniente e sobre demanda, a um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente fornecidos e liberados com um mínimo de esforço de gestão e interação com o provedor de serviços (MELL e GRANCE, 2010).

Dessa forma, esse modelo permite que o usuário da nuvem não necessite saber sobre a configuração ou localização física do sistema que está disponibilizando os serviços, fazendo uso dos recursos de hardware e software via conexão de rede e que de outra forma seria muito mais custoso e complexo de implantar e manter.

Bhadauria e Sanyal (2012) apontam alguns recursos que caracterizam a computação em nuvem e que viabilizaram sua criação. Esses recursos são:

- Virtualização - Talvez o principal viabilizador do surgimento da computação em nuvem. A virtualização busca simular a existência de múltiplos servidores funcionando em um único servidor físico, aumentando assim a eficácia do mesmo. Com a virtualização, é possível o oferecimento dinâmico de infraestrutura de TI virtual (máquinas virtuais, armazenamento, entre outros), onde a nuvem fornece

seus recursos quando existe a demanda ou os dispensa quando não mais utilizados;

- *Application Programing Interface* (API) - Com uma API o consumidor da nuvem pode requisitar funcionalidades e receber os serviços providos pela nuvem, além de poder alterar configurações;

- *Web Service* e SOA - Para que serviços sejam disponibilizados através de uma rede (tipicamente a *World Wide Web*), a nuvem faz uso de comunicação via *web service*, que pode ser através de framework REST (*Representational State Transfer*) (FIELDING, 2000) ou SOAP (*Simple Object Access Protocol*) (W3C, 2007). É através de mensagens *web service* que a API realiza as requisições. Além disso, a organização interna dos serviços de uma nuvem segue a SOA (*Service Oriented Architecture*), combinando diferentes serviços para realização de tarefas específicas;

- Web 2.0 e *Mash-up* - A *web* 2.0 tem possibilitado a criação de páginas *web* e envolvimento de usuários de forma interativa e colaborativa. *Mash-up* é uma aplicação web que combina informações de mais de uma fonte em uma ferramenta única e integrada. Com esses dois elementos, aplicativos *web* podem ser executados diretamente da nuvem ou utilizando seus recursos;

Existe uma grande quantidade de tipos de serviços e implantações de nuvens que se adequam a essas características. Tendo isso em vista, há a necessidade de buscar uma padronização semântica e uma classificação de tipos de sistemas de computação em nuvem. Uma proposta que proporciona isso também é oferecida pelo NIST, que caracteriza a nuvem em quatro possíveis modelos de implantação e três modelos de serviço. Essa definição também é largamente aceita e se tornou a principal divisão dos tipos de nuvem computacional. Os modelos de implantação são (MELL e GRANCE, 2010):

- Nuvem Privada - A infraestrutura da nuvem é provida para uso exclusivo por uma única organização que possui vários consumidores (por exemplo, unidades de negócio). Pode pertencer, ser gerenciada e operada pela organização, terceiros ou alguma combinação entre os dois.

- Nuvem Comunitária - A infraestrutura da nuvem é provida para uso exclusivo de uma comunidade específica de consumidores que possuem preocupações comuns. Pode ser gerenciada, operada e pertencer a uma ou várias organizações pertencentes à comunidade, terceiros ou alguma combinação entre os dois.

- Nuvem Pública - A infraestrutura é provida para uso ao público em geral, não restrita a um nicho. Pode ser gerenciada, operada e pertencer a uma empresa, academia científica ou organização governamental, ou então uma combinação entre elas.

- Nuvem Híbrida - A infraestrutura é uma composição entre outras duas ou mais infraestruturas de nuvem (pública, comunitária ou privada) que pertencem a entidades distintas, mas que são unidas através de tecnologias padronizadas ou proprietárias que viabilizam a portabilidade.

Os modelos de serviço demonstrado na Figura 2.1 são descritos abaixo:

- *Software as a Service* (SaaS) - O serviço provido para o consumidor é o de usar um aplicativo fornecido sobre uma infraestrutura em nuvem. Ou seja, um *software* ou aplicativo é fornecido aos usuários e os recursos que esse software utiliza são da própria nuvem. Geralmente esses softwares são acessados através de *web browsers* e internet. Exemplos de SaaS são os editores de texto em *web browser*, algumas redes sociais, dentre outros. Vale ressaltar que existem sistemas SaaS que fornecem armazenamento de dados, mas se enquadram nesse modelo pois fornecem aplicações, geralmente para diversas plataformas, que também estão sendo fornecidas através de um serviço de nuvem. Exemplos desses sistemas são o Dropbox (2015), Google Drive (2015), Microsoft OneDrive (2015), dentre outros.

- *Platform as a Service* (PaaS) - A capacidade provida para o consumidor é a de implantar aplicativos criados por linguagens de programação, bibliotecas e ferramentas. Ou seja, a nuvem fornece uma plataforma ou ambiente completo para realização de desenvolvimento de software.

- *Infrastructure as a Service* (IaaS) - A capacidade provida para o consumidor é a de usar processamento, armazenamento, rede e outros recursos computacionais

onde o consumidor pode também implantar e executar software arbitrário (como sistema operacional ou aplicativos em máquinas virtuais).

Esses três modelos funcionam com certa dependência, onde a IaaS fornece seus recursos para nuvens PaaS, que por sua vez podem fornecer seus recursos para nuvens SaaS.

Figura 2.1 – Modelos de serviço de nuvem.

Fonte: Produção própria.

Dentre os sistemas que fornecem IaaS, alguns se destacam na comunidade acadêmica por serem projetos de código aberto. Esse é o caso dos sistemas Eucalyptos (NURMI *et al*., 2009), Nimbus (2015), OpenNebula (2015) e OpenStack (2015).

Já dentre os sistemas de computação em nuvem da iniciativa privada que proveem serviços do tipo IaaS, um dos mais utilizados é o sistema *Amazon Simple Storage Service* (AMAZON WEB SERVICES, 2013). Ele disponibiliza o serviço de armazenamento de dados, realizando cobrança por quantidade de dados armazenados como também pelas operações de envio e retirada desses dados. Esse serviço chegou à marca de 950 bilhões de objetos armazenados e média de 650 mil requisições por segundo já no final do ano de 2012 (AMAZON WEB SERVICES, 2012).

Com a popularização do Amazon S3, a sua interface de disponibilização de serviços passou a ser uma referência para sistemas *open source* de armazenamento em nuvem, como o Walrus e Cumulus, utilizados pelo Eucalyptus e Nimbus respectivamente, cujas interfaces são compatíveis com as do Amazon S3.

2.1.2 Segurança e Computação em Nuvem

Dada à complexidade da computação em nuvem, além de seus benefícios, surgiram novos desafios que ao não serem devidamente tratados se caracterizam como problemas e possivelmente empecilhos para a adoção dessa tecnologia.

Alguns desses desafios são: interoperabilidade, padronizações (de acesso, operação, dentre outros), questões jurídicas e de políticas organizacionais relacionadas à terceirização de serviços de TI, privacidade e segurança, monitoramento, auditoria e confiabilidade. Dentre esses desafios, um dos mais importantes é a segurança e como a nuvem a provê (SABAHI, 2011), pois é um fator determinante para o sucesso de uma nova tecnologia (BHADAURIA e SANYAL, 2012). Como o foco do trabalho está relacionados a nuvens provedoras de armazenamentos de dados, este tópico focará principalmente na segurança desses tipos de nuvens.

Uma das maiores preocupações relacionadas à segurança em nuvem é o armazenamento de dados de consumidores (SAHA e DAS, 2012). Quando esses consumidores utilizam os recursos da nuvem, como armazenamento, os dados deixam de estar apenas sobre o seu controle e passam, também, a estarem sobre o controle do provedor da nuvem.

Esses dados são armazenados em um local desconhecido pelo cliente, em um ou vários servidores localizados em qualquer lugar do mundo. Quando em ambientes de nuvem privada, essa falta de controle é mitigada, uma vez que o consumidor possui domínio sobre a infraestrutura da nuvem.

Com a distribuição do controle dos dados é possível ocorrer problemas de privacidade, que é uma área chave e de principal preocupação dos consumidores. Esses problemas são mais preocupantes se a natureza dos dados for sensível, como informações financeiras ou de saúde (PEARSON, 2009).

Além dos problemas relacionados à privacidade, também surgem novos riscos relacionados à segurança como um todo, com novos cenários de possibilidade de ataque e roubo desses dados.

Esses novos riscos de segurança em nuvem já se provaram perigosos e importantes através de falhas reais em que ocorreram roubos de dados de usuários armazenados em nuvem. Um desses casos ocorreu em 2007, quando o provedor de nuvem Salesforce.com informou que os e-mails e informações de endereço de seus usuários foram roubados (GREENBERG, 2008). Outro exemplo ocorreu em 2010, quando se constatou que um funcionário da Google utilizava seus privilégios de acesso para roubar informações de usuários, quebrando a privacidade dos mesmos, algo aparentemente não raro na empresa (RAM, 2010).

Devido a essa importância da segurança na nuvem, levantamentos de riscos têm sido propostos, buscando maior compreensão das minúcias dessa área de desafio que a nuvem enfrenta.

Um dos primeiros documentos com esse tipo de levantamento foi criado pela Gartner, apresentando sete riscos possivelmente presentes em nuvem (BRODKIN, 2008; TAN e AIB, 2011), focando principalmente no provedor. Esses riscos estão descritos, de forma resumida, abaixo:

1. Usuário com Privilégio de Acesso:
   Dados sensíveis, presentes fora da infraestrutura do consumidor, possuem um nível de risco de segurança inerente. Como por exemplo, a possível existência de usuários (provavelmente funcionários do provedor do serviço) com privilégios de acesso aos servidores da nuvem.

2. Conformidade Regulamentar:
   Provedores de computação em nuvem podem não estar dispostos a realizar auditoria externa ou certificação de segurança.

3. Localização de Dados:
   Quando um usuário armazena seus dados na nuvem, provavelmente não sabe onde esses dados serão hospedados.

4. Segregação de Dados:

Dados na nuvem estão normalmente em um ambiente compartilhado, lado a lado com dados de outros consumidores. Além disso, criptografia no provedor pode gerar problemas de disponibilidade ou inutilizar dados se não for feita corretamente.

5. Recuperação:
   Mesmo sem saber a localização dos dados, os usuários devem ser informados do que irá acontecer com seus dados em caso de algum desastre.

6. Suporte a Investigação:
   Investigação ou auditoria de atividade ilegal ou inapropriada pode ser impossível em computação em nuvem.

7. Viabilidade em Longo Prazo:
   Idealmente, o provedor da computação em nuvem nunca deveria ir à falência ou ser comprado por outra empresa. Usuários devem permanecer com acesso a seus dados mesmo depois de um evento dessa natureza.

Outro levantamento de riscos é feito pela *Cloud Security Alliance* (2010), que divulgou o que considera os sete maiores riscos de segurança em sistemas de computação em nuvem, partindo do ponto de vista de um atacante. Esses riscos estão listados e descritos a seguir:

1. Uso Abusivo e Nefasto de Computação em Nuvem:
   Ao abusar do relativo anonimato provido por sistemas de computação em nuvem, autores de códigos maliciosos, *spammers* e outros criminosos têm conseguido conduzir suas atividades com relativa impunidade.

2. Interfaces e APIs Inseguras:
   Operações sobre a nuvem são realizadas através da sua API. Mecanismos de autenticação, controle de acesso, monitoramento, acesso em anonimato e transmissão segura necessitam ser bem definidos e implementados para que a API não venha apresentar vulnerabilidade possa ser aproveitada por agentes maliciosos.

3. Funcionários Maliciosos:
   Um provedor pode não divulgar como permite o acesso a ativos físicos e virtuais a funcionários e como esses são monitorados.

4. Compartilhamento de Infraestrutura Tecnológica:
   Os componentes da infraestrutura são utilizados por diversos clientes simultaneamente, mas não foram projetados para fornecer propriedades de isolamento forte.

5. Perda de Dados ou seu "vazamento":
   Ameaças de comprometimento de dados aumentam na nuvem devido à suas características únicas. Os dados correm o risco de serem acessados por entidades não autorizadas, ou mesmo perdidos em casos de falhas de máquinas.

6. Apropriação de Serviço ou de Conta:
   A conta ou serviço de um usuário pode vir a ser apropriada através de roubo de suas credenciais. Em posse dessas credenciais, um agente poderá utilizar o poder da reputação da conta do usuário para lançar ataques subsequentes.

7. Perfil de Risco Desconhecido:
   Versionamento de software, *updates* de código, práticas de segurança, perfis de vulnerabilidade, tentativas de invasão e arquitetura de segurança são todos fatores importantes para estimular uma postura de segurança de uma empresa fornecedora de computação em nuvem.

Além desses riscos existe a possibilidade do próprio provedor de nuvem fazer uso indevido dos dados de seus usuários infringindo a privacidade e confidencialidade. Essas informações podem ser comercializadas para fins de publicidade direcionada como bancos ao realizarem avaliações para empréstimos e contratação de funcionários.

2.1.3 Nuvem OpenStack Swift

O OpenStack tem se tornado a plataforma *open source* de maior preferência para a construção de sistemas de nuvem, sendo usado por grandes fornecedores de serviço como a

HP, Rackspace, MercadoLivre, IBM, dentre outras (COMPANIES SUPPORTING THE OPENSTACK FOUNDATION, 2015).

A criação dessa plataforma teve como principais agentes a empresa Rackspace e a NASA (*National Aeronautics and Space Administration*) uma vez que estas instituições não estavam satisfeitas com o uso da principal plataforma de nuvem *open source* até o momento, o Eucalyptos.

É nesse cenário que o OpenStack surgiu sobre a licença Apache, onde ambas as organizações e todos os que desejassem apoiar, se dedicariam ao desenvolvimento de uma plataforma para criação de nuvens privadas ou públicas.

Essa plataforma possui três projetos principais: o OpenStack Nova dedicado ao provisionamento de processamento, o *Openstack Image Service* também chamado de *Glance* e o *Openstack Swift* dedicado ao armazenamento de objetos (OPENSTACK, 2015).

O projeto Swift é responsável pelo armazenamento de dados em nuvem em forma de objeto e pode trabalhar de forma independente dos demais projetos. Dessa forma é possível a criação de nuvens de armazenamento públicas (concorrendo com o Amazon S3) ou privadas (concorrendo com o Eucalyptos Walrus). No Swift, os objetos são armazenados dentro de containers, e esses containers são associados a contas de usuários ou grupo de usuários.

Para alcançar uma melhor disponibilidade, os dados armazenados recebem redundância entre diferentes servidores. Além disso, os servidores podem ser divididos em agrupamentos, formando zonas dentro da nuvem. Caso ocorra um particionamento da nuvem, como por exemplo, a queda da rede de uma zona ou um servidor ficar inativo, outros servidores suprem as requisições.

2.2 Federação de Identidade

Como mencionado no Capítulo 1, uma federação de identidade consiste na colaboração entre diferentes entidades que podem assumir dentre dois papéis: provedores de identidade (IdP) ou provedores de serviços (SP). As interações entre essas entidades se baseia em uma confiança limitada, através de protocolos e políticas estabelecidas e acordadas.

No relacionamento entre SPs e IdPs, o provedor de identidade confia que o provedor de identidade é habilitado para comprovar a identidade do usuário e fornecer atributos

verdadeiros. Por sua vez, o IdP confia que o SP manterá a privacidade dos atributos da identidade dos usuários (SAML SPECIFICATIONS, 2015).

A federação surgiu como alternativa ao modo centralizado de autenticação em decorrência do aumento de serviços providos fora do domínio local dos usuários, principalmente através da internet, e que estão além do seu controle. Dessa forma, a federação de identidade permite a portabilidade da identidade entre domínios diferentes de segurança.

O objetivo maior da federação de identidade é permitir a usuários de um domínio acessarem recursos (sistemas, dados, serviços compartilhados, dentre outros) em outro domínio sem dificuldades e sem redundância na administração de contas de usuário. E isso pode ser conseguido através da Autenticação Única (*Single Sign-On* – SSO). O SSO consiste em um usuário poder se autenticar e acessar um sistema e em seguida acessar outros sistemas sem necessitar se autenticar novamente. Outro significado se refere à possibilidade de usar apenas uma credencial para se autenticar em diversos domínios, eliminando a necessidade do usuário de memorizar inúmeras senhas e nomes (AUTHENTICATION WORLD, 2015).

Apesar da praticidade, o SSO recebe críticas no que se refere a um potencial maior risco em caso de comprometimento da segurança das credenciais de um usuário, resultando em agentes maliciosos poderem acessar diversos domínios com apenas uma credencial de usuário.

Para que esses domínios diferentes possam se comunicar, a federação requer protocolos padronizados. Esses padrões geralmente são abertos ou abertamente divulgados. Existem diferentes padrões e abordagens para implementação de sistemas de gerência de identidade federada. O precursor dessas tecnologias, e ainda largamente utilizado, é o SAML (*Security Assertion Markup Language*). O SAML é um padrão aberto e gerenciado pelo OASIS (2015) e faz uso do formato XML em suas mensagens.

Outro padrão é o oAuth (2015), disponível nas versões 1.0 ou 2.0. O oAuth é um protocolo de delegação de autorização, em que um usuário permite a um SP acessar seus recursos que estão em outro local. Esse protocolo não é exatamente específico para federação de identidade propriamente dita. Apesar disso, seu mecanismo permite uma configuração que obedece a um comportamento de federação de identidade, apesar de limitado. Assim, um usuário pode permitir a um SP acessar seus atributos (recursos) em outra entidade (no papel de IdP).

SAML é um componente da nova arquitetura proposta por esse trabalho. Devido às peculiaridades do oAuth, este é usado em uma variação secundária da arquitetura. Ambos os protocolos são utilizados na prototipagem e são descritos com maiores detalhes no Capítulo 3.

2.3 Criptografia Baseada em Atributos

A criptografia de chave pública (assimétrica) pode garantir o sigilo de dados em um ambiente de compartilhamento simples. De forma a compartilhar um arquivo segundo esse mecanismo, é necessário cifrá-lo com a chave pública da entidade a quem o arquivo se destina e assim apenas essa entidade poderá decifrá-lo através de sua própria chave privada (STALLINGS, 2004).

Para o compartilhamento entre múltiplas entidades, ao utilizar esse tipo de criptografia é necessário cifrar o mesmo arquivo com a chave pública de cada entidade. Isso resulta tanto em um custo computacional elevado como também em custo de armazenamento, ou seja. é necessário armazenar múltiplas cópias de um mesmo arquivo. Isso é inviável para um ambiente com muitos compartilhamentos entre múltiplas entidades.

Uma alternativa para isso é o uso da criptografia simétrica. Nesse mecanismo um usuário encripta um arquivo a ser compartilhado com outro usuário, mas necessita compartilhar a chave privada para que a outra parte possa decifrá-lo. Assim, apesar de a criptografia simétrica ser mais eficiente computacionalmente que a de chave pública, ela requer o compartilhamento prévio de chaves simétricas. Além disso, ela carece de um mecanismo para controle de acesso. Ou seja, para se compartilhar um arquivo é necessário entregar a mesma chave para todos os participantes; isso pode colocar em risco os arquivos compartilhados caso essa chave seja comprometida (ALSHEHRI *et al.*, 2012).

Tanto a criptografia assimétrica quanto a simétrica exigem que as partes envolvidas se conheçam e troquem algum tipo de informação (chave pública ou privada). Isso não atende aos cenários em que o dono do arquivo não sabe previamente a identidade de quem poderá acessar o mesmo. Dentre os diversos cenários possíveis, esse é o particular contexto de registros de saúde. Como por exemplo, um paciente pode desejar que seu registro seja acessível apenas por determinada categoria de profissionais, ao invés de um profissional em específico, ou mesmo que possa vir a decidir com quem compartilhar apenas em outro momento futuro (DIXIT e SURESH, 2013).

Por causa das limitações desses métodos tradicionais, Sahai e Waters (2005) criaram o que chamaram de criptografia baseada em atributos (*Attribute Based Encryption* - ABE) com o objetivo de oferecer segurança e controle de acesso via criptografia. Esse protocolo criptográfico permite que dados sejam cifrados com base em uma lista de atributos. Múltiplos usuários podem decifrar os dados contanto que suas chaves (que são distintas) estejam associadas a um número mínimo, pré-estabelecido, de atributos que também foram utilizados no momento da cifragem. Isso permite um compartilhamento mais intuitivo e sem as dificuldades mencionadas anteriormente em protocolos criptográficos tradicionais.

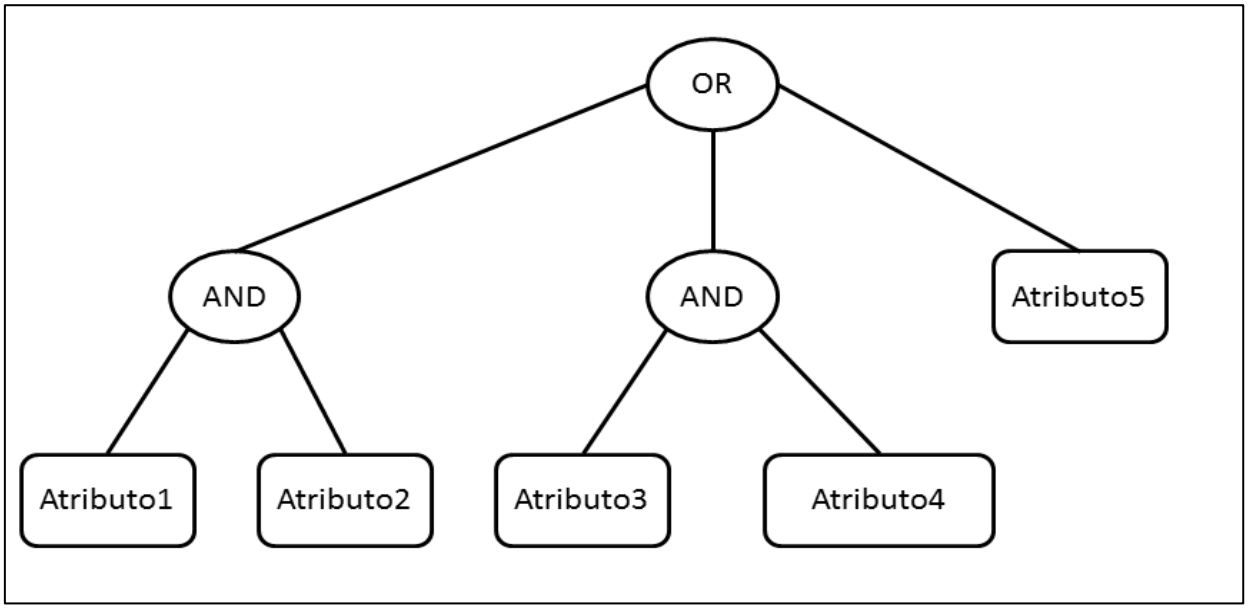
O ABE foi um passo natural após a criptografia baseada em identidade (*Identity Based Encryption* - IBE) e suas variações. Enquanto que na IBE a identidade tem o papel similar ao da chave pública, em que é usada como parâmetro para a cifragem e também criação de chave privada, no ABE o leque de parâmetros se estende para suportar diversos atributos que inclusive podem ser comuns a usuários distintos.

Os atributos utilizados no ABE podem representar características reais de usuários como papeis, perfis, identificadores, profissão ou qualquer outra característica. A disponibilização desses atributos, para que possam ser utilizados em qualquer operação de cifragem de dados, é de responsabilidade de uma Autoridade de Atributos (AA). Essa AA também é responsável pela criação de chaves. Assim, ela é um elemento central do protocolo e requer confiança. Cada usuário recebe uma chave criada pela AA e baseada em seus próprios atributos.

Posteriormente, Bethencourt *et al*. (2007) propuseram uma adaptação para o ABE, chamada de CP-ABE (*Ciphertext-Policy Attribute-Based Encryption*). No CP-ABE ao invés de requerer uma lista de atributos, é necessária uma política de acesso baseada em atributos no momento da cifragem. Nessa proposta, a chave continua sendo associada a uma lista de atributos.

Essa política pode ser representada através de uma árvore, onde cada nó pai representa um operador lógico do tipo AND ou tipo OR, como exemplificada na Figura 2.2. Com isso, um usuário pode definir diferentes conjuntos de combinações de atributos que terão de ser satisfeitos, em uma chave, para que ocorra a decifragem. Assim, a política é utilizada como parte da cifragem e a chave associada a atributos deve satisfazer a política para que possa decifrar.

Figura 2.2 – Exemplo de política de atributos para um cenário de saúde eletrônica.



Fonte: Produção própria.

Essa criptografia é utilizada na forma dos algoritmos: *setup*, *encrypt*, *keygen*, *decrypt*. A seguir são descritos resumidamente como é cada uma dessas etapas do protocolo. Para maiores detalhes, é possível realizar consultar ao trabalho de Sahai e Waters (2005).

Setup. O algoritmo de *setup* é feito pela AA e irá selecionar um grupo bilinear $\mathbf{G}_o$ de ordem prima *p* através do gerador *g*. A seguir, irá selecionar dois componentes randômicos $\alpha,\beta \in \mathbb{Z}_p$. A chave pública é publicada como (sendo *e* um mapa bilinear):

$$PK = G_o, g, h = g^{\beta}, f=g^{1/\beta}, e(g,g)^{\alpha}$$

E a chave mestra é MK = $(\beta,g)^{\alpha}$.

***Encrypt*(PK, M, T)**. Esse algoritmo cifra uma mensagem M segundo uma árvore T de atributos. Inicialmente o algoritmo escolhe um polinomial q_x para cada nó (ou folha) *x* da árvore T. Sendo Y o conjunto de nós folhas em T, o texto cifrado CT é computado com:

$$CT = (T, \hat{C}=Me(g,g)^{\alpha s}, C=h^{s}, \forall y \in Y: C_y = g^{q_y(0)}, C'y = H(att(y))^{q_y(0)}).$$

KeyGen(MK,S). O algoritmo de geração de chaves tem como entrada um set de atributos e tem como resultado uma chave que se identifica com os atributos. O algoritmo primeiro seleciona um $r \in \mathbb{Z}_p$ randômico e então um $r_j \in \mathbb{Z}_p$ randômico para cada atributo $j \in$ S. Então computa a chave como:

$$SK = (D=g^{(\alpha+r)/\beta}, \forall j \in S: D_j = g^{r} \cdot H(j)^{r_j}, \grave{D_j}=g^{r_j}).$$

***Decrypt*(CT,SK)**. O algoritmo para decifrar é recursivo. Para isso, inicialmente é necessário um algoritmo auxiliar *DecryptNode*(CT, SK, *x*) que é definido como (sendo *x* um nó da árvore T):

$$DecryptNode(\text{CT, SK, } x) = e(g,g)^{rq_x(0)}.$$

Então, para cada nó *z* que for filho do nó *x*, é chamado o algoritmo *DecryptNode*(CT, SK, *z*) e seu resultado é armazenado em F_z. Assim, a F_x será alcançada através de:

$$F_x = \prod_{z \in Sx} F_z^{\Delta_{i,S\grave{x}}(0)} \quad onde \;\; i = index\,(z) \; e \; s\grave{x} = \{ index\,(z) : z \in S_x \}$$

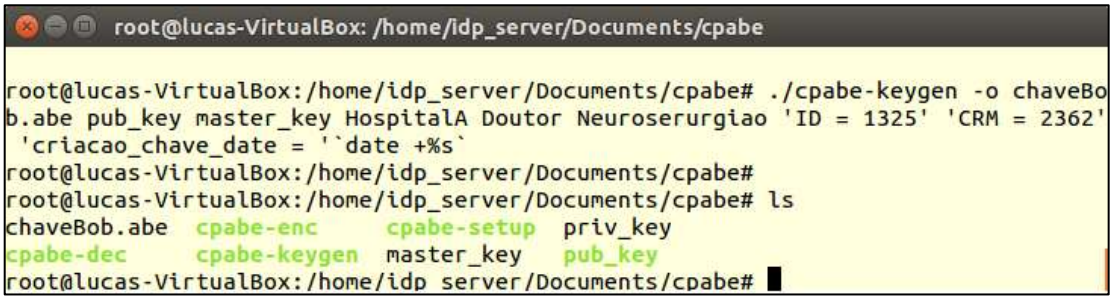
$$F_x = e\,(g,g)^{rq_x(0)}.$$

Com isso, para decifrar basta chamar *DecryptNode* na raiz de *T*. Tomando esse resultado como A, para obter a mensagem M basta computar:

$$\hat{C}/(e(C,D)/A) = C/\,(e(h^s, g^{(\alpha+r)/\beta})\,/e(g,g)^{rs}) = M.$$

Os autores do CP-ABE disponibilizaram ainda uma biblioteca desenvolvida em C para testes desse protocolo criptográfico[1]. Na Figura 2.3 é demonstrada a geração de uma chave ABE através do uso dessa biblioteca. Nesse exemplo é demonstrado os parâmetros para a criação da chave: a chave mestre da autoridade, a sua chave pública e os atributos da pessoa a quem a chave pertence. No exemplo ainda são demonstrados alguns atributos de um profissional de saúde e uma data de criação da chave. Com isso, a chave criada está atrelada a essa lista de atributos.

Figura 2.3 – Exemplo de criação de uma chave para um profissional de saúde.

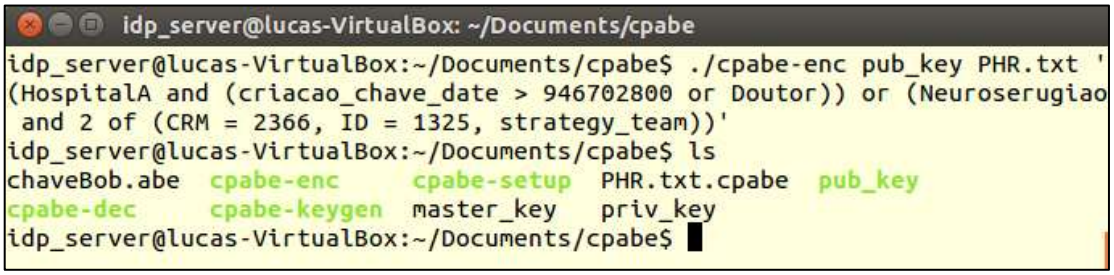


Fonte: Produção própria.

[1] http://acsc.cs.utexas.edu/cpabe/

A Figura 2.4 demonstra o uso da biblioteca para encriptar um registro pessoal de saúde. Como parâmetro se tem o caminho para arquivo a ser cifrado, a chave pública da autoridade e a política ABE que define as relações entre atributos. É essa a política que deve ser satisfeita por uma chave na ocasião da decifragem. Como exemplo de política, no comando percebe-se alguns relacionamentos entre atributos através dos operadores AND e OR.

Figura 2.4 – Exemplo de cifragem de registro de saúde através do uso de uma política ABE.



Fonte: Produção própria.

Além de atributos em cadeia de caractere (ex.: HospitalA), pode-se utilizar também comparação com atributos numéricos (ex.: data menor que um dia específico). Além disso, mais um recurso é o uso do operador "*of*" para estabelecer uma relação de quantidade mínima dentro de um conjunto de atributos (ex.: último parâmetro da Figura 2.4).

A centralização da emissão de chaves e verificação sobre todos os atributos em uma única AA pode representar um risco ao uso do CP-ABE. Caso a AA não se mostre confiável, poderá emitir chaves para acessar indevidamente arquivos cifrados. Além disso, é potencialmente problemático ter apenas uma única AA para se responsabilizar por atributos e usuários que podem pertencer a diferentes entidades, como é caso da federação de identidade.

Outra desvantagem é o gargalo de apenas uma autoridade central, que pode vir a ser sobrecarregada com operações de criação de chaves. A tentativa de replicar uma autoridade centralizada em múltiplas instâncias é uma estratégia que elevaria o risco de exposição da chave mestre.

Por essa razão existem propostas de descentralização de AA. Dentre eles a de Lewko e Waters (2011) permite múltiplas autoridades sobre um modelo CP-ABE (*Multi-Authority ABE* - MA-ABE). Com isso, diferentes AAs são responsáveis por atributos distintos e

emitem chaves para usuários. Esse modelo se enquadra melhor no cenário de *e-health* e federação de identidade. Por exemplo, uma entidade responsável pela regulamentação da profissão médica pode ser a autoridade sobre o atributo “Cirurgião”, enquanto que outra entidade responsável por pesquisas de saúde pode ser responsável pelo atributo “pesquisador”. A proposta desse trabalho permite o uso tanto do CP-ABE quanto do MA-ABE, mas se concentra na forma descentralizada.

Os principais desafios do MA-ABE são a criação de chaves com atributos de diferentes AAs e a criação de políticas também com relações entre atributos de diferentes AAs sem a necessidade de coordenação entre as mesmas. O modelo MA-ABE proposto por Lewko e Waters (2011) atende a esses desafios e ainda permite que novas autoridades ingressem no modelo sem a necessidade de reconfiguração ou sincronia global entre as AAs, com a exceção da criação de alguns parâmetros de referência. Assim, esse protocolo criptográfico é escalável e mais seguro, já que autoridades se tornam mais independentes e caso alguma seja corrompida não irá afetar o funcionamento de outras autoridades.

Diferentes AAs podem emitir chaves privadas para um mesmo usuário referentes aos atributos aos quais são responsáveis. Por essas autoridades poderem mesmo nem ter conhecimento da existência uma das outras, usa-se um identificador global por usuário. Isso permite unir os segmentos de chaves em uma única chave privada para o usuário (conjunto de chaves ABE), contendo representação de atributos de diferentes domínios.

O MA-ABE é composto por cinco algoritmos. São eles: configuração global, que cria parâmetros globais; configuração de AA, que estabelece a autoridade; encriptar arquivo, que cifra arquivos com base em políticas; geração de chave, para criar chaves privadas com base em atributos; e decifrar, que usa chave privada para decriptografar um arquivo. Cada um desses algoritmos é utilizado em etapas da arquitetura proposta por esse trabalho e serão mais detalhados no próximo capítulo.

2.4 Registros Eletrônicos em *E-health*

Como mencionado na introdução, os registros médicos evoluíram da forma tradicional em papel para sua forma eletrônica em documentos digitais. A princípio esses registros apenas tinham migrado de mídia, possuindo as mesmas características da forma anterior. Posteriormente, novas facilidades e funcionalidades foram propostas a ponto de tornar ainda

mais vantajoso à adoção de Registros Eletrônicos de Saúde, também chamados de EHR (*Eletronic Health Record*).

Dentre os benefícios desses novos registros em relação ao de papel está à facilidade de atualização de dados e a convergência de informações. Dessa forma, os dados de pacientes deixam de estar espalhados, repetidos e muitas vezes inconsistentes entre diferentes locais para estarem centralizados em um único registro. Isso permite também a disponibilidade via rede sem a necessidade de transporte físico do registro, aumentando a velocidade de recuperação da informação (ROBINSON *et al*., 2012).

A manutenção de EHR também é facilitada por permitir a atualização em um único registro, mesmo que por diferentes profissionais de saúde. Além disso, informações podem ser atualizadas ou inseridas de forma automatizada, como por exemplo, informada por um sistema de coleta de resultados de exames laboratoriais.

O uso de EHR promete melhorar a qualidade do cuidado médico e consequentemente a saúde de pacientes. Um exemplo é a possibilidade de profissionais de saúde diferentes não precisarem solicitar o mesmo exame para um paciente, uma vez que o resultado já está no registro eletrônico compartilhado. Assim é possível evitar gastos adicionais com procedimentos já realizados e também poupar o paciente do esforço e custo extra (DIXIT e SURESH, 2013; FERNÁNDEZ-ALEMÁN *et al*., 2013).

Uma limitação do EHR é a dificuldade em incorporar informações de todas as fontes relevantes (ROBINSON *et al*., 2012). Isso é decorrente de nem todos provedores de assistência médica estarem associados de forma integrada e interoperável ao EHR e, também, de geralmente o próprio paciente ter acesso limitado ao registro. Dessa forma, um paciente pode se consultar com um médico que não possui acesso ao EHR ou estar tomando alguma medicação por conta própria e que por essa razão não esteja registrado no EHR. Como consequência pode-se levar a criação de múltiplos EHRs espalhados em diferentes provedores de assistência médica, com exames e consultas redundantes. Isso potencialmente elimina o proposito do EHR e sua principal vantagem em relação aos registros em papel.

A restrição de o próprio paciente não poder contribuir para o seu EHR, foi a principal motivação que levou a proposta do Registro Pessoal de Saúde (*Personal Health Record* – PHR). O PHR se diferencia do EHR devido o registro estar sobre a responsabilidade do próprio paciente a quem se refere. Nessa forma de registro a premissa é que o dono da informação seja o próprio dono do registro, a quem cabe o poder de decisão quanto ao

compartilhamento e às permissões de visualização e alteração. Ainda que um provedor de assistência médica ou outra entidade seja quem mantenha o PHR em nome do paciente, deve garantir a soberania do proprietário do registro (ROBINSON *et al*., 2012).

Apesar dos benefícios, tanto o PHR quanto o EHR apresentam desafios de integração entre sistemas, portabilidade entre provedores de assistência médica, usabilidade em manipulação e organização de dados no registro, disponibilidade, segurança e privacidade, entre outros. Dentre estes, a falta de confiança na segurança e privacidade é o fator que impede uma maior adoção de *e-health* por pacientes (GOLDMAN e ZOE, 2000).

O termo segurança engloba a proteção de informação contra acessos indevidos ou não autorizados. Já o termo privacidade possui uma definição mais nebulosa, sujeita a variáveis culturais. Aqui, no contexto de EHRs e PHRs, o termo está atrelado à privacidade da informação de saúde, e se refere à habilidade de um indivíduo de deter o controle de informações pessoais em custódia de terceiros (SAHAMA, SIMPSON e LANE, 2013).

O casamento de *e-health* e computação em nuvem também possui outros desafios. Um significativo exemplo é a falta de padrões específicos para a interoperabilidade e migração de dados entre sistemas de saúde como também entre provedores de nuvens. Isso pode afetar diretamente pacientes, que ficam presos a um provedor de plano de saúde, por exemplo, e a um provedor de nuvem.

Outro desafio é o campo jurídico. Servidores de uma nuvem podem estar espalhados por diferentes países que podem ter ou não legislação para reger o uso e proteção de dados dos clientes. Ainda no âmbito jurídico, a definição de proprietário dos dados pode variar entre países. Um exemplo foi quando a Holanda buscava implantar um sistema de registros de saúde e surgiu o cenário de os dados estarem fisicamente armazenados nos EUA. Com isso se levantou o debate de o governo dos EUA poder ou não acessar os dados. Por fim o projeto foi encerrado pelo ministro de saúde por não atender as normas de privacidade.

A questão de confiança tem sido um motivo para usuários estarem relutantes em usar serviços de nuvem como parte da assistência médica (EVERETT, 2009). Portanto, para que seja viável o uso de cenários desse tipo, são necessários mecanismos que satisfaçam os requisitos de segurança e privacidade, a fim de justificar a confiança no sistema e proteger os dados sensíveis dos pacientes. Com esse objetivo, pesquisas científicas têm buscado meios satisfatórios de garantir a segurança e proteção de privacidade em registros eletrônicos armazenados na nuvem (vede subseção 1.5 dessa dissertação).

2.5 Considerações Finais

Apesar das vantagens do uso de registros de saúde com nuvem, o ponto crítico do uso desse serviço é a elevação do risco de quebra de segurança e perda de privacidade de pacientes, algo já temido no contexto de PHR e EHR por si só. O acesso indevido aos sensíveis dados de saúde e a sua divulgação pode ser catastrófico para pacientes e potencialmente irreparável. Para que essa integração com a nuvem seja viável, se faz necessário mecanismos que garantam a privacidade dos dados armazenados.

A federação de identidade, apresentada nesse capítulo, demonstra uma forma alternativa de se gerenciar identidade e atributos de usuários ao modo tradicional utilizado em nuvem e em sistemas de saúde. A federação permite colaboração entre domínios diferentes de entidades ou organizações parceiras para o compartilhamento de serviços.

Criptografia baseada em atributos é uma alternativa à criptografia tradicional simétrica ou assimétrica e com aplicação intuitiva e conveniente ao contexto de saúde. Pacientes e profissionais de saúde podem ter chaves correspondentes as suas próprias características (ex.: identificação, profissão, entre outros) e criptografarem arquivos de acordo com uma política de acesso.

Pesquisas têm sido desenvolvidas em busca de proporcionar segurança e privacidade para PHR e EHR, inclusive com o uso de computação em nuvem. No entanto possuem limitações.

Os elementos apresentados nesse Capítulo são os componentes fundamentais da arquitetura proposta no próximo capítulo. Cada um deles possui um papel singular para a criação do mecanismo que propõe o armazenamento seguro de registros eletrônicos de saúde.

3 A NOVA ARQUITETURA

Este capítulo trata do principal resultado da pesquisa apresentada nesta dissertação, que aborda a proposta de uma nova arquitetura para assegurar a proteção à privacidade de registros de saúde armazenados em um ambiente de nuvem, bem como, das ferramentas utilizadas para obter esses resultados.

3.1 Considerações Iniciais

Como apresentado no Capítulo 1, existem diversas pesquisas com propostas que envolvem controle de acesso e armazenamento seguro de arquivos na nuvem. No entanto, a solução aqui apresentada possui vantagens quando comparada com elas. Essa solução promove compartilhamento seguro e gerenciado pelo usuário, revogação de acesso e segurança no armazenamento de dados. Em particular, ela provê comprovação de atributos de usuários perante AAs e gerência de chaves.

A fim de permitir o armazenamento seguro e compartilhamento de dados na nuvem, a nova solução é composta pelas seguintes partes: plataforma de nuvem, autoridades de atributo (AA), provedores de identidade (IdP), prestadores de serviços em nuvem (SP), e um conjunto de usuários. Os usuários podem ter os papeis de dono do arquivo ou colaborador (com quem o arquivo é compartilhado).

Para a proposta, devem-se considerar as seguintes suposições de segurança. Os IdPs e AAs são confiáveis e organizações que prezam por interesses dos usuários devem estabelecê-las em uma federação. Os provedores de nuvem devem seguir o protocolo corretamente, isto é, desempenham seu papel como definido. No entanto, prestadores de tais serviços podem ameaçar a privacidade e segurança dos usuários ao tentar ler os registros de saúde. Se não seguirem o esquema, isso implica em não funcionamento do serviço corretamente e a federação pode cancelar o contrato ou parceria com tais provedores. A terceira premissa é a

utilização de um canal de comunicação seguro para a troca de mensagens entre as partes, como o uso de SSL ou TLS.

A seguir serão descritas inicialmente as ferramentas utilizadas para a construção do sistema e posteriormente os elementos da nova arquitetura em si.

3.1.1 Protocolos para Federação de Identidades

Como citado no Capítulo 2, os dois protocolos utilizados neste trabalho são o SAML e o oAuth que serão descritos, agora, com mais detalhes.

3.1.2 *Security Assertion Markup Language*

Através do SAML é possível realizar, no ambiente web, autenticação e autorização entre diferentes domínios de segurança, alcançando também SSO (*Single-Sign On*). Para isso, o padrão faz uso de *tokens* que são trocados entre uma autoridade SAML, no papel de provedor de identidades (IdP), e um consumidor no papel de provedor de serviço (SP).

Cada *token* trocado entre o IdP e o SP contém asserções, isto é, conjuntos de declarações sobre uma entidade ou sujeito (ex.: usuário). Um IdP pode emitir três tipos de asserção: autenticação, declarando que o sujeito foi autenticado; atributos, declarando atributos associados ao sujeito; ou autorização, aceitando ou declinando o pedido de acesso a um recurso (SAML SPECIFICATIONS, 2015).

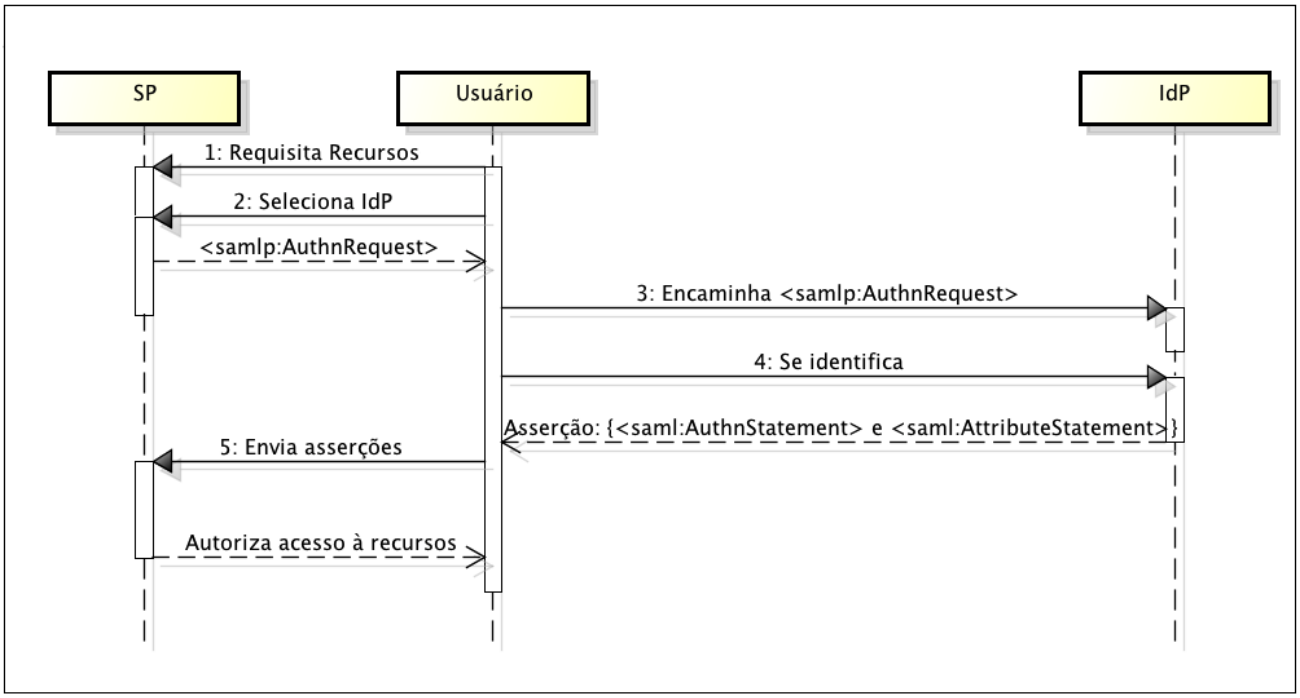
A Figura 3.1 apresenta uma simplificação da forma mais comum de se realizar autenticação em um SP de serviço Web através de um IdP. Apesar de o padrão SAML suportar algumas outras formas, essa é a que será considerada neste trabalho. Além disso, aqui se utiliza o HTTP como forma de envio de mensagens, apesar de a especificação não restringir a esse meio, podendo ser utilizado, como por exemplo, SOAP.

Ainda na Figura 3.1, o fluxo de autenticação é iniciado pelo usuário que deseja fazer uso de recursos em um SP (passo 1). Como o SP não gerencia autenticação, ele criará uma requisição de autenticação através de uma mensagem SAML do tipo <samlp:AuthnRequest>. Essa requisição é codificada em base 64. Ela é enviada através de um HTTP *Redirect* pelo usuário para o IdP que escolheu (passo 2). Em geral essas mensagens são pequenas e por isso

podem ser enviadas através do *Redirect*. Mensagens no formato HTTP Post também podem ser enviadas nessa etapa e não possuem limitação de tamanho.

Após receber a mensagem SAML de requisição (passo 3), o IdP irá realizar a operação de autenticação com o usuário, tipicamente através de nome e senha (passo 4). Dessa forma, o IdP construirá uma mensagem SAML, também codificada em base 64. Essa mensagem é do tipo <samlp:response> e contém as asserções de autenticação e de atributos. Ela será então repassada através do usuário para o SP (passo 5) em mensagem HTTP do tipo Post.

Figura 3.1 – Diagrama de sequência de autenticação federada.



Fonte: Produção própria.

Por fim, ao receber as asserções de atributos e de autenticação com sucesso, o SP poderá criar um contexto de segurança, na forma de sessão para o usuário. Com base nos atributos, ele também poderá realizar a autorização, através de controle de acesso, e liberar o recurso desejado pelo usuário.

Vale ressaltar que o SP requer confiança nos IdPs. Uma vez que IdPs são autoridades na federação e confiáveis no sentido de comprovar ou não a identidade e atributos de seus próprios usuários, essa relação de confiança é algo natural na federação. Apesar disso, a confiança recíproca é limitada à visualização dos atributos, uma vez que o SP é o consumidor das asserções em um domínio externo.

Como um SP tem total controle sobre a função de autorização (liberar o recurso), é de sua responsabilidade realizar o controle de acesso após a autenticação. Em um cenário de SP de nuvem pública de armazenamento de dados, isso se apresenta como um risco aos usuários por ser um serviço terceirizado e possivelmente não confiável. O provedor da nuvem possui a capacidade de infringir contra a segurança e confidencialidade de membros da federação por deter o poder sobre o controle de acesso e a posse dos dados armazenados. Assim, um SP de nuvem mal-intencionado poderá acessar e ler os dados, como também liberá-los de forma não autorizada.

3.1.3 oAuth 2.0

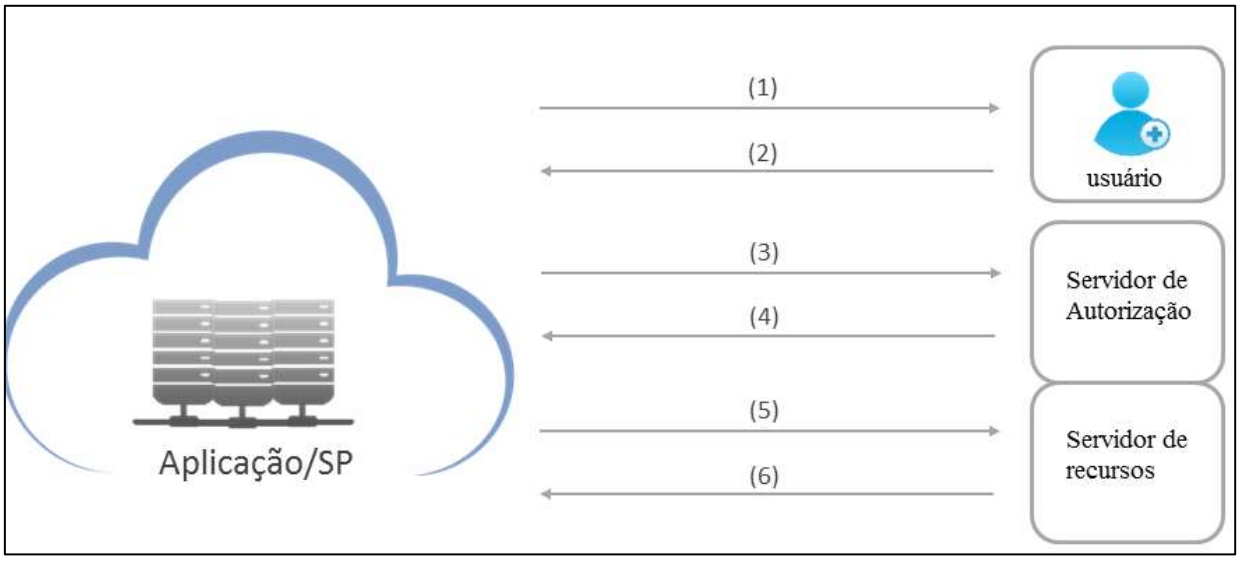
O oAuth versão 2 é um framework de autorização regido pela IETF através do RFC 6749. Ele especifica um processo que permite a donos de recursos autorizarem a terceiros o acesso a seus recursos sem a necessidade de compartilharem suas credenciais. Diferentemente do SAML, o oAuth usa exclusivamente HTTP como forma de envio de suas mensagens.

A Figura 3.2 apresenta um diagrama com o processo generalizado de autorização através do oAuth. A seguir estão os passos de forma mais detalhada (IETF RFC 6749, 2015):

1. Requisição de autorização: uma aplicação (SP) realiza ao usuário o pedido de autorização de acesso aos seus recursos;
2. Autorização: caso o usuário autorize o acesso, emite uma autorização para a aplicação;
3. Requisição de acesso: a aplicação envia para o servidor de autorização uma prova de sua própria identidade e a autorização concedida pelo usuário;
4. Concessão de acesso: caso a identidade da aplicação e a autorização sejam válidas, o servidor de autorização emite um *token* de acesso. Aqui a autorização é concluída;
5. Requisição do recurso: a aplicação envia o *token* de acesso para o servidor de recursos (geralmente é o mesmo que o servidor de autorização) requisitando o recurso;
6. Envio de recurso: caso o *token* seja válido, o recurso é enviado para a aplicação.

Esse fluxo inicial está representado de forma abstrata, onde a autorização descrita no passo 2 pode ser realizada de quatro diferentes formas, dependendo da política de autorização e a implementação. Essas diferentes formas são: código de autorização, utilizado para aplicações em servidores; implícito, para aplicativos executados localmente com o usuário (ex.: dispositivos móveis); credenciais, a senha e nome de usuário são fornecidos diretamente para a aplicação; e credenciais de aplicação, quando o recurso é da própria aplicação e não de usuários.

Figura 3.2 – Diagrama de passos gerais de autorização via oAuth2.



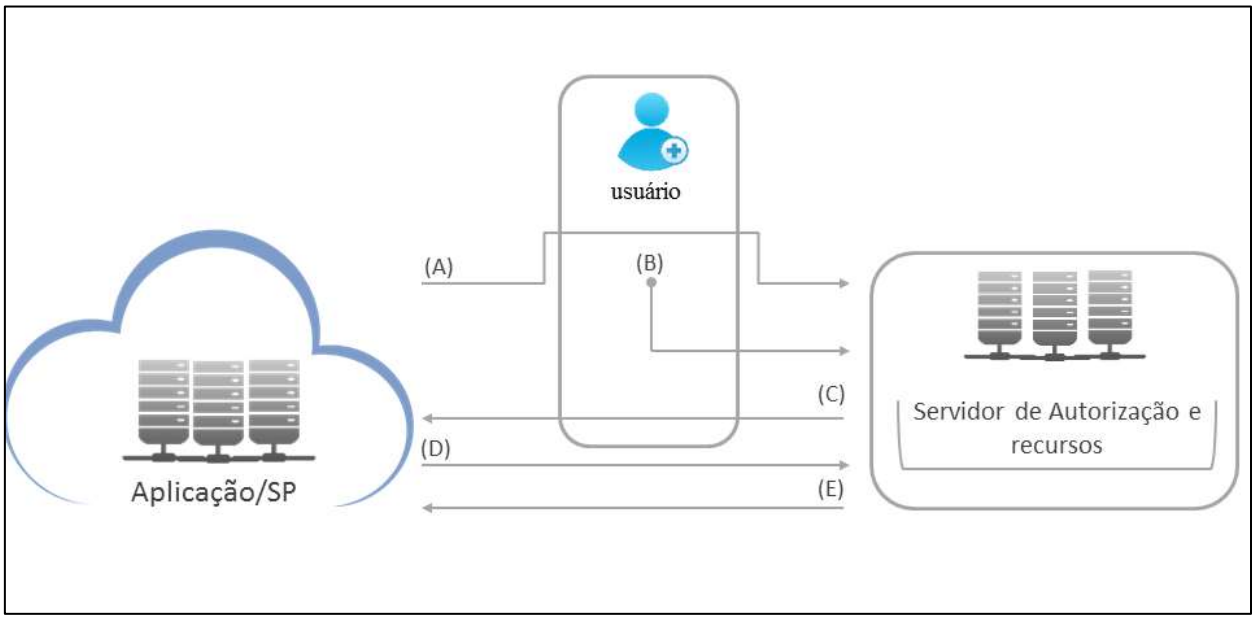
Fonte: Produção própria.

A autorização mais comum no oAuth, e utilizada nesse trabalho, é a com uso de código de autorização. Nesse fluxo, a autorização é baseada em redirecionamentos através do usuário. Essa forma é detalhada a seguir e representada pelo diagrama da Figura 3.3:

A. A aplicação redireciona o usuário para o servidor de autorização, informando sua identidade e que tipo de acesso deseja para os recursos;
B. O usuário se autentica (se já não estiver autenticado) perante o servidor de autorização, decidindo então quais tipos de acesso deseja permitir para a aplicação;
C. Caso o usuário tenha autorizado à aplicação no passo anterior, o servidor de autorização redireciona o usuário de volta para a aplicação incluindo um código de acesso;

D. A aplicação envia para o servidor de autorização um pedido de autorização de acesso, incluindo o código recebido;
E. Caso o código de acesso seja válido, o servidor de autorização responde com um *token*. Assim a aplicação pode usar esse *token* para acessar o recurso de acordo com o escopo.

Figura 3.3 – Diagrama de oAuth 2 com código de autorização.



Fonte: Produção própria.

A segurança no oAuth se baseia principalmente no uso de comunicação segura através de SSL ou TLS. Enquanto que no SAML as entidades utilizam certificados digitais e par de chaves para comprovarem ser realmente quem dizem ser, no oAuth há o uso de um segredo compartilhado entre a aplicação e o servidor de autorização para comprovar identidade.

3.2 Configuração Inicial

De modo a implantar a arquitetura, as partes interessadas primeiro estabelecem as políticas da FIM. Como já mencionado, estas partes são organizações que podem ser provedoras de assistência médica, entidades governamentais, organizações sem fins lucrativos, entre outros. As políticas FIM especificam exatamente quais as organizações estão

colaborando para formar a FIM, quais serão AAs ou IdPs, e também determinar, via contrato ou parceria, quais nuvens prestarão serviços à federação sob o papel de SPs.

De forma que não seja possível a entrada de um falso IdP ou SP na federação, nessa etapa os componentes estabelecem suas assinaturas eletrônicas. Com isso cada IdP possui um registro de SPs e outros IdPs com acesso a suas chaves públicas. Assim, caso alguém tente se passar por um IdP ou SP da federação, será detectado através da assinatura eletrônica que mensagens de tais locais não são permitidas por não serem genuínas da federação.

Nessa fase também há o registro de usuários em seus respectivos IdPs. Eles associam os usuários a seus atributos e a uma identificação global baseada no número da identificação única (como o registro geral, RG). IdPs também mantêm um mapeamento para AAs e seus respectivos atributos gerenciados. Por sua vez, cada AA conhece e confia nos IdPs que pertencem ao FIM. Os atributos aos quais uma AA é responsável por gerenciar e os procedimentos para adicionar novas AA deve satisfazer a política organizacional da federação.

Além desses procedimentos, as partes interessadas devem executar os seguintes algoritmos. Aqui se usa o protocolo proposto por Lewko e Waters (2011) (ver Subseção 2.3), todavia, é possível utilizar outros protocolos ABE.

> **CinfiguracaoGlobal (λ)**: as partes executam esse algoritmo para estabelecer os parâmetros globais do ABE. Ele tem como entrada um parâmetro de segurança (λ) e tem como resultado os parâmetros globais (PG). Os PG são utilizados durante todo o resto da arquitetura.
>
> **ConfiguracaoAA (PG):** esse algoritmo gera os elementos criptográficos específicos para cada AA. Ele tem como entrada os PG e tem como saída uma chave privada (SK) e uma chave pública (PK). ConfiguracaoAA deve ser executado por cada AA e uma única vez.

Uma organização pode se responsabilizar pela implantação de AA e IdP simultaneamente. Alguns atributos podem ser genéricos e comuns a muitos usuários de diferentes organizações, como por exemplo, “médico cardiologista”, “farmacêutico”, “enfermeiro”, dentre outros. Instituições governamentais ou organizações sem fins lucrativos podem gerir AAs para esses atributos.

Por outro lado, alguns atributos são específicos de um contexto ou mesmo de uma organização, como por exemplo, a identificação do plano de saúde, locais de trabalho, um

hospital ou laboratório. Esses atributos devem ser geridos pela própria organização. Não obstante, AAs podem ainda ser implementadas de forma independente do IdP, e vice-versa.

3.3 Armazenamento em nuvem

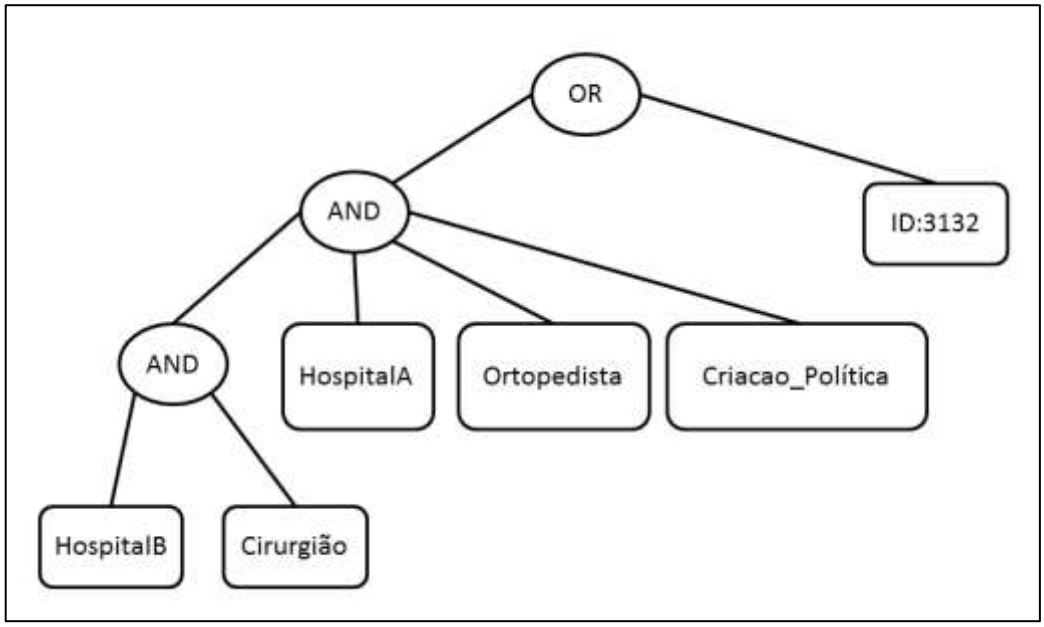
A fim de armazenar um registro eletrônico na nuvem, o seu dono deve executar dois algoritmos, o CifrarRS e CifrarABE. O dono do registro deve executar o primeiro algoritmo para criptografar o registro com uma chave simétrica e em seguida criptografá-lo com uma política ABE. Os algoritmos são:

> **CifrarRS (K, RS):** esse algoritmo tem como entrada uma chave simétrica K e um registro de saúde (RS). Ele cifra o RS utilizando K através de um algoritmo de criptografia simétrica, como o AES. Como resultado, se tem um texto cifrado (TC).
>
> **CifrarABE(K, P_{ABE}, PG, {PK}):** esse algoritmo tem como entrada a chave simétrica K, uma política baseada em atributos (P_{ABE}), os parâmetros globais e um conjunto de chaves públicas das AAs envolvidas (AAs responsáveis por atributos presentes em P_{ABE}). O algoritmo cifra a chave K através do ABE de acordo com a política e tem como saída o texto cifrado TC_{ABE}.

A política P_{ABE} necessariamente deve conter o identificador global do dono do registro. Este atributo de identificação garante que o proprietário do arquivo é capaz de acessar seu arquivo no futuro. Um exemplo de política está na Figura 3.4, em que estão presentes os atributos de identificação independente das relações entre os outros atributos. A política ABE deve também incluir um atributo Criacao_Política referente à data de criação da política. Tal data nas políticas é necessária para o mecanismo de revogação de acesso.

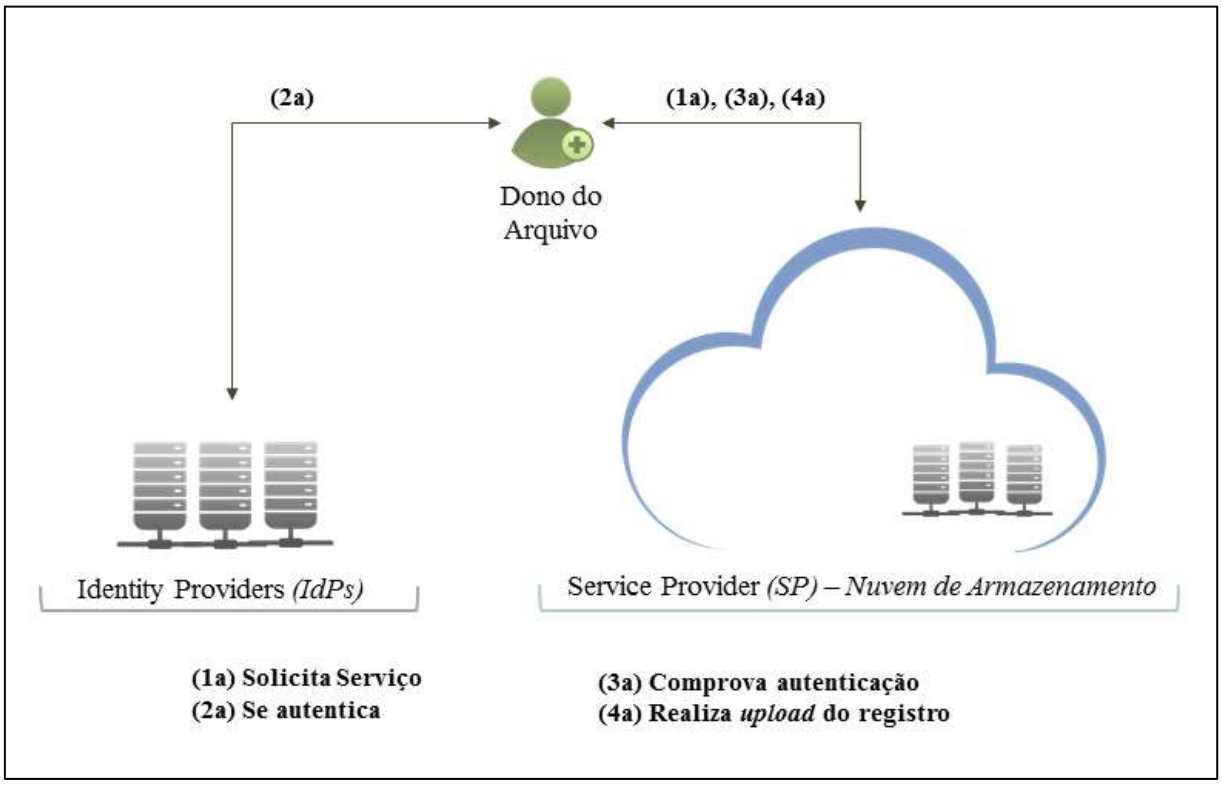
Depois de executar os algoritmos, o dono do arquivo é agora capaz de armazenar seu registro de saúde criptografado na nuvem SP, conforme ilustrado na Figura 3.5. Para isso, ele solicita ao SP o serviço de armazenamento (passo 1a) e escolhe se autenticar por meio da FIM, selecionando um IdP específico (onde possui cadastro). Isso aciona uma mensagem de requisição SAML enviada pelo SP para o IdP escolhido. Essa requisição SAML é redirecionada via usuário e está exemplificada no Quadro 3.1. Nele é identificado o requerente (a nuvem) através do elemento <saml:Issuer> e o desejo de receber uma asserção de autenticação e de atributos.

Figura 3.4 – Exemplo de política de atributos para um cenário de saúde eletrônica.



Fonte: Produção própria.

Figura 3.5 – Passos para realizar armazenamento em nuvem na nova arquitetura.



Fonte: Produção própria.

Ao receber a requisição, o IdP solicita e verifica as credenciais do usuário (passo 2a,

Figura 3.5), é retornado uma resposta SAML para o SP redirecionado através do usuário (passo 3a, Figura 3.5). Essa resposta SAML é uma asserção que contém uma declaração de autenticação. Essa mensagem é exemplificada, de forma resumida, no Quadro 3.2. Nela se destaca a presença do <saml:AuthnStatement> que contém o contexto de autenticação.

Quadro 3.1 Exemplo de mensagem de requisição de autenticação.

```
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="xxxx-781246-5314-231z-133t"
    Version="2.0"
    IssueInstant="2004-12-05T09:21:59"
    AssertionConsumerServiceIndex="0"
    AttributeConsumingServiceIndex="0">
    <saml:Issuer>https://sp.nuvem.com/SAML2</saml:Issuer>
    …
  </samlp:AuthnRequest>
```

Quadro 3.2 Exemplo de mensagem de resposta de autenticação.

```
<saml:Response>
…
<saml:Assertion>
   <saml:Subject>
     …
   <saml:AuthnStatement
     AuthnInstant="2015-7-05T09:22:00"
     SessionIndex="b07b804c-7c29-ea16-7300-4f3d6f7928ac">
     <saml:AuthnContext>
       <saml:AuthnContextClassRef>
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
      </saml:AuthnContextClassRef>
     </saml:AuthnContext>
   </saml:AuthnStatement>
</saml:Assertion>
</saml:Response>
```

Isso permite que o usuário armazene o conjunto de textos cifrados {TC, TC_{ABE}} na

nuvem (passo 4a, Figura 3.5). Em conjunto com o upload, algumas informações são inseridas nos metadados de armazenamento: a P_{ABE}, a identificação do proprietário do arquivo e a data de criação. É também desejável que a nuvem informe a URL do registro armazenado. Este URL é usado para facilitar o compartilhamento de registros de saúde.

3.4 Compartilhamento de registros de saúde

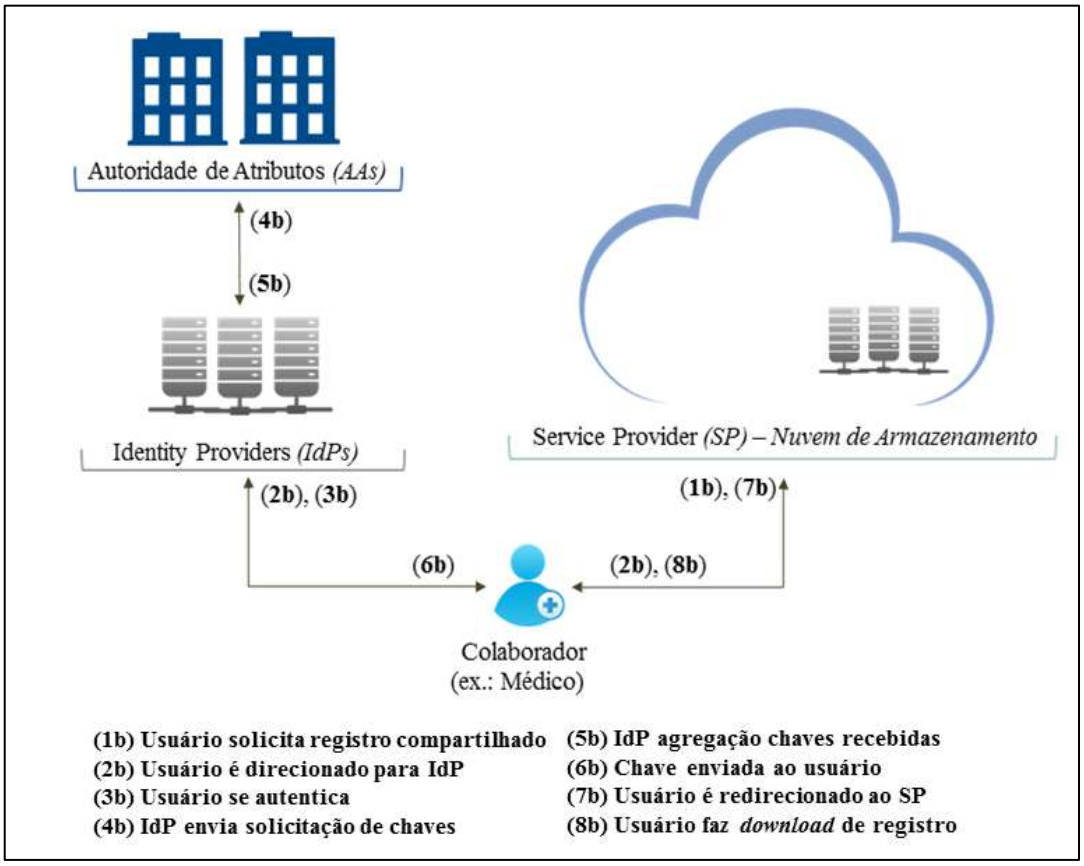
A fim de compartilhar informações de forma segura, não há necessidade de qualquer outra operação de delegação de acesso além de estabelecer relações entre atributos na política ABE. Uma vez que o proprietário do arquivo criptografa seu registro através do ABE com uma política P_{ABE}, o protocolo criptográfico assegura o controle de acesso. Isso resulta em colaboração entre os usuários através da partilha segura de registro de saúde.

A Figura 3.6 apresenta o processo de compartilhamento de registros de saúde. Este processo se inicia quando um usuário solicita acesso (passo 1b) a um dado registro armazenado em um SP (possivelmente através da URL). O SP então cria uma solicitação SAML do tipo <samlp: AuthnRequest> que é enviada para o IdP escolhido pelo usuário (passo 2b). O SP também envia o P_{ABE} (salvo nos metadados) para o IdP na mesma mensagem, assim o IdP fica ciente que a requisição é de compartilhamento e não apenas autenticação.

Ao receber a mensagem de requisição do SP, juntamente com P_{ABE}, o IdP exige as credenciais do usuário, como nome de usuário e senha (passo 3b, Figura 3.6). Depois da comprovação da identidade, o IdP verifica se os seus atributos (ligada ao seu registro) satisfazem a política de ABE. Se isso não acontecer, o IdP responde uma mensagem de acesso negado informando quais atributos não possui e seriam necessários.

É válido mencionar que essa verificação inicial da política no IdP não é essencial para garantir a segurança, pois o protocolo ABE garante o controle de acesso. No entanto, sem esta verificação, um usuário bem-intencionado só descobre que não têm privilégios de acesso o suficiente depois de fazer as operações criptográficas e não for possível ler o arquivo. Isto levaria a perda de tempo e de recurso computacional. Portanto, por meio da verificação de política no IdP, evitam-se essas operações desnecessárias quando o acesso é negado.

Figura 3.6 – Passos para realizar compartilhamento em nuvem na nova arquitetura.



Fonte: Produção própria.

Uma vez que o IdP identifica que o usuário possui atributos que satisfazem P_{ABE}, são mapeadas as AAs correspondentes a cada atributo necessário. É então solicitada (passo 4b, Figura 3.6) à respectiva AA a chave correspondente ao atributo. Cada AA então executa o algoritmo KeyGen. Este algoritmo tem o objetivo de criar uma chave de acordo com o atributo informado. No caso da AA ser separada do IdP, pode-se enviar uma mensagem na forma de uma asserção SAML do tipo que lista os atributos do usuário. Essa asserção é exemplificada no Quadro 3.3 que possui o elemento <saml:AttributeStatement> onde é informado cada elemento <saml:Attribute> contendo um valor para um atributo do usuário. O algoritmo é descrito a seguir:

> **KeyGen (GID, PG, *i*, SK):** Este algoritmo recebe como entrada o ID global do usuário (GID), parâmetros globais do sistema (PG), o atributo escolhido (*i*), e a chave secreta do AA (SK). Ele gera uma chave correspondente ao atributo *i*.

Depois de executar o algoritmo, cada AA retorna (passo 5b, Figura 3.6) as chaves individuais para o IdP, que então irá agregá-las em um conjunto chamado de K_{ABE} (incluindo um atributo de data atual) que aqui será tratado como uma única chave para o usuário. Para fins de otimização, o IdP pode solicitar uma chave a partir de um conjunto mínimo de atributos necessários para satisfazer a política, em vez de um conjunto que abrange todos os atributos do usuário.

Quadro 3.3 Exemplo de mensagem de fornecimento de atributos.

```
<saml:Assertion …>
    …
   <saml:AttributeStatement>
     <saml:Attribute
xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
       x500:Encoding="LDAP"

       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
       FriendlyName="Profissao">
       <saml:AttributeValue
         xsi:type="xs:string">médico</saml:AttributeValue>
       <saml:AttributeValue
xsi:type="xs:string">ortopedista</saml:AttributeValue>
     </saml:Attribute>
   </saml:AttributeStatement>
 </saml:Assertion>
```

O IdP, em seguida, retorna K_{ABE} para o usuário (passo 6b, Figura 3.6) juntamente com a resposta SAML <samlp: response>. Enquanto que a chave permanece apenas com o usuário, a resposta é encaminhada automaticamente para o SP (passo 7b, Figura 3.6) para provar o sucesso da autenticação e criar sessão de SSO. Ao receber esta prova, o SP permite o *download* de {TC, TC_{ABE}} (passo 8b, Figura 3.6). Para obter o registro de saúde de forma legível, o usuário precisa executar os seguintes algoritmos na respectiva ordem:

DecifraABE(TC_{ABE} , PG, {K_{ABE}}): Este algoritmo tem como entrada um texto cifrado ABE (TC_{ABE}), os parâmetros globais do sistema (GP), e a chave K_{ABE}. O algoritmo decifra TC_{ABE} e obtém a chave simétrica K. Se a chave K_{ABE} não satisfizer a

política ABE utilizada no momento da cifragem, o processo de descodificação não retorna K.

DecifraRS(TC, K): A entrada deste algoritmo é o registro de saúde cifrado (TC) e a chave simétrica (K) obtida após a execução do algoritmo DecifraABE. Ele decifra TC e retorna o registro de saúde.

Para atualizar o registro armazenado, o usuário criptografa o RS utilizando K e então faz o *upload* para o SP. Para atualização de controle de acesso, a política P_{ABE} precisa ser alterada com as novas regras de acesso e atualizada nos metadados de armazenamento, como também seu atributo de data de criação atualizado. Além disso, K deve ser recriptografada com a nova política e retransmitido para o SP. Apenas o dono do arquivo é permitido alterar K cifrado (que é o TC_{ABE}) e P_{ABE} armazenados na nuvem. O dono do arquivo também é capaz de transferir a posse do arquivo, mudando isso nos metadados. Portanto, além de partilha, profissionais de saúde podem também transferir a posse de EHR para outros.

3.5 Revogação de Acesso

Os procedimentos apresentados permitem que um proprietário de arquivo armazene e compartilhe registros de saúde de forma segura. No entanto, em um cenário realista, esses procedimentos não são suficientes. Um proprietário de arquivo ainda precisa de uma forma de revogar o acesso a seus registros. Um paciente pode necessitar dessa operação quando muda de médico e não deseja que o anterior continue acessando seus dados do PHR, por exemplo. Isto é, ele permite que o novo médico acesse seus registros, enquanto que desautoriza o antigo médico. Além disso, um proprietário de arquivo precisa revogar o acesso a seu registro no caso de outro usuário (por exemplo, um enfermeiro) ter perdido atributos.

Em outras palavras, revogação de acesso é necessário em dois casos: o proprietário de um registro deseja alterar quem tem acesso ao seu registro de saúde; ou um usuário perdeu atributos que tinha anteriormente e, portanto, não satisfaz mais a política ABE de um determinado registro de saúde.

No mecanismo, um usuário é capaz de revogar o acesso ao seu registro para desautorizar a outro usuário. Para tal, é necessário que altere a política utilizada para criptografar a chave simétrica K com uma nova configuração de atributos. Isso é, o usuário executa novamente o algoritmo CifrarABE com o parâmetro K e uma nova política P_{ABE}. O

algoritmo retorna como saída o texto cifrado TC_{ABE}. O dono do arquivo faz então o upload de TC_{ABE} substituindo na nuvem com a versão atualizada.

Ao alterar o acesso através de mudanças na política, p proprietário do registro não necessita alterar o registro de saúde cifrado já armazenado na nuvem, uma vez que esteja usando a chave simétrica K. Essa chave pode ser obtida decifrando TC_{ABE} antes da atualização. Caso contrário, o registro de saúde precisa ser encriptado novamente utilizando um novo K com o algoritmo EncripritaRS. Com isso é criado um novo TC (registro de saúde encriptrado) que é reenviado para a nuvem.

No caso de usuários que perderam atributos, eles perdem acesso aos registros cujas políticas não são mais satisfeitas. Como IdPs verificam se usuários possuem atributos que satisfazem políticas P_{ABE} durante o compartilhamento de arquivos, eles não recebem uma chave válida e resposta SAML. Como consequência, não conseguirá acesso ao arquivo armazenado na nuvem. Ou seja, a revogação, nesse caso, acontece de forma natural.

Em um cenário de a nuvem cooperar de forma maliciosa com um usuário que anteriormente possuía acesso, porém perdeu atributos, é possível que esse usuário obtenha o arquivo ainda que não seja permitido. Mesmo ele tendo guardado uma chave antiga, que satisfazia P_{ABE}, não conseguirá obter o registro de saúde. O atributo de data de criação (ou atualização) contido na política ABE é suficiente para tornar a chave antiga ineficiente, uma vez que possui atributo de data de criação anterior à data exigida na política atualizada.

3.6 Adaptação da arquitetura com oAuth

O uso de oAuth é uma alternativa ao SAML. Apesar de ter limitações, a pilha de operações do oAuth é menor. No entanto, não possui a mesma adaptabilidade para um cenário de múltiplas entidades em colaboração como o SAML, que permite mais facilmente uma implantação com múltiplos IdPs.

O oAuth tem sido popularizado pela provisão de SSO e autorização de acesso a recursos por terceiros. Além disso, serviços populares de redes sociais, como o Google e Facebook, têm fornecido interface oAuth para uso com outros serviços. Inclusive é possível

obter do Facebook alguns atributos de usuários (que autorizarem), como nome, local de trabalho, formação e inclusive fotos[2].

No mecanismo proposto por este trabalho é possível utilizá-lo tanto como uma alternativa ao SAML nos IdPs, como também uma forma de separar AAs de IdPs em outro provedor de serviço para uma FIM.

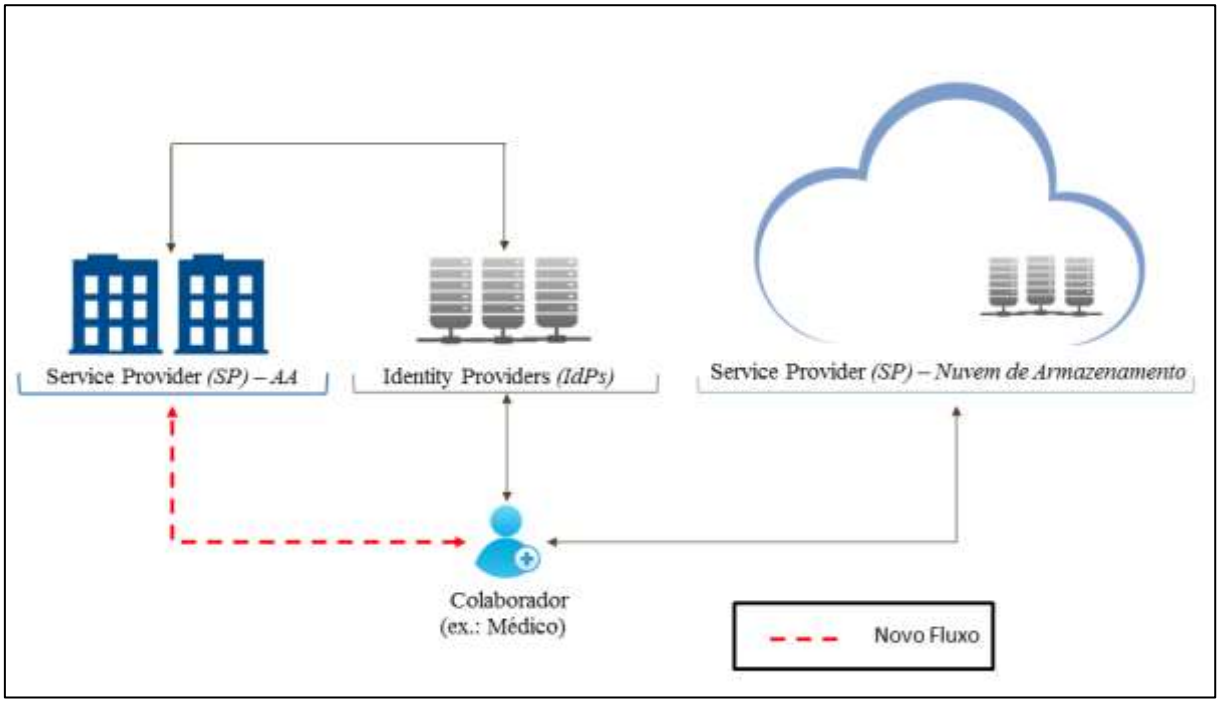
Na primeira opção, SPs de nuvem passam a se comunicar utilizando oAuth em substituição às mensagens SAML descritas na etapa anterior. Apesar disso, os passos do fluxo do mecanismo continuam os mesmos. Usuários donos de registros se autenticam em IdPs oAuth para então armazenarem seus registros na nuvem. Usuários colaboradores requisitam acesso a registros armazenados e são redirecionados a IdPs oAuth para autorizarem acesso aos atributos e também receberem sua chave ABE. A diferença está em que o recurso oAuth sendo requisitado nesse cenário é justamente o conjunto de atributos do usuário.

Já na segunda opção, é inserido um terceiro elemento na FIM, um SP cujo serviço é a criação de chaves ABE. Assim, AAs passam a ser SPs oAuth que permitem a usuários solicitarem diretamente chaves ABE correspondentes aos seus atributos. Nessa variação da arquitetura as nuvens continuam com o mesmo papel e fluxo descrito na seção anterior. Já as AAs passam a ter o papel de SP. A Figura 3.7 ilustra essa adaptação.

O fluxo de comunicação entre usuário e SP de nuvem continua seguindo os mesmos passos da arquitetura principal, tanto para donos de arquivos quanto para colaboradores independentemente de uso de mensagens SAML ou oAuth. O fluxo de autenticação de um colaborador em um IdP também permanece com o usuário se autenticando e recebendo chave ABE via IdP. Nesse fluxo ainda, o IdP pode se comunicar via oAuth com AAs (no papel de SPs) para requisitar a chave para usuários. Já no novo fluxo (ligação pontilhada na Figura 3.7), usuários podem requisitar diretamente a AAs (ainda no papel de SPs) a chave ABE. Nesse fluxo, segue-se o processo padrão de autenticação via IdP para que o usuário se autentique e AAs obtenham a lista de atributos para então criar a chave correspondente.

[2] https://developers.facebook.com/docs/graph-api

Figura 3.7 – Fluxos do mecanismo com a adição e adaptação do oAuth.



Fonte: Produção própria.

O proveito da adição desse novo fluxo é a possibilidade de usuários terem ABE como um serviço. No entanto, o grande desafio é a comunicação entre os próprios SPs de AA, uma vez que usuários podem ter atributos que estão sobre responsabilidade de diferentes autoridades. É possível permitir que o IdP seja o agregador das chaves ou usuários repetirem o fluxo para cada AA em modo SP.

3.7 Considerações Finais

Este capítulo apresentou a nova arquitetura, principal objetivo deste trabalho. Os passos para realização de armazenamento em nuvem, compartilhamento e revogação de acesso aos registros de saúde foram detalhados, como também quais algoritmos criptográficos são utilizados e quais mensagens são encaminhadas entre os componentes da arquitetura.

Com esse mecanismo, usuários tem a possibilidade de armazenar seus registros de saúde de forma segura em um ambiente terceirizado. Nuvens maliciosas podem mesmo tentar acessar os dados de registros para algum fim, mas são impossibilitadas de obterem o conteúdo devido ao protocolo ABE, mesmo estando com a posse física do arquivo.

Apesar de os registros estarem criptografados, o mecanismo ainda permite o compartilhamento. Usuários colaboradores podem acessá-los caso satisfaçam a política de acesso estabelecida pelo proprietário do registro, através de suas próprias chaves criptográficas.

Além disso, também foi apresentado uma variação da arquitetura com o uso do protocolo oAuth no papel de IdP ou no papel de um novo serviço para a federação. Isso permite o uso de ABE sobre demanda na forma de um serviço, até mesmo para outros fins além de registros de saúde.

4. PROTÓTIPOS E RESULTADOS

Este capítulo aborda os protótipos gerados, bem como os resultados. Foram criados dois protótipos, um utilizando o protocolo SAML 2.0 e outro com oAuth 2.0. Aqui serão abordados os aspectos técnicos do desenvolvimento desses protótipos, as configurações de implantação dos mesmos e testes realizados.

4.1. O Objetivo da prototipagem

O principal objetivo da criação dos protótipos é demonstrar a viabilidade da proposta, na forma de uma prova de conceito, mesmo que em protótipos limitados e que não representam um funcionamento completo do mecanismo segundo a sua arquitetura.

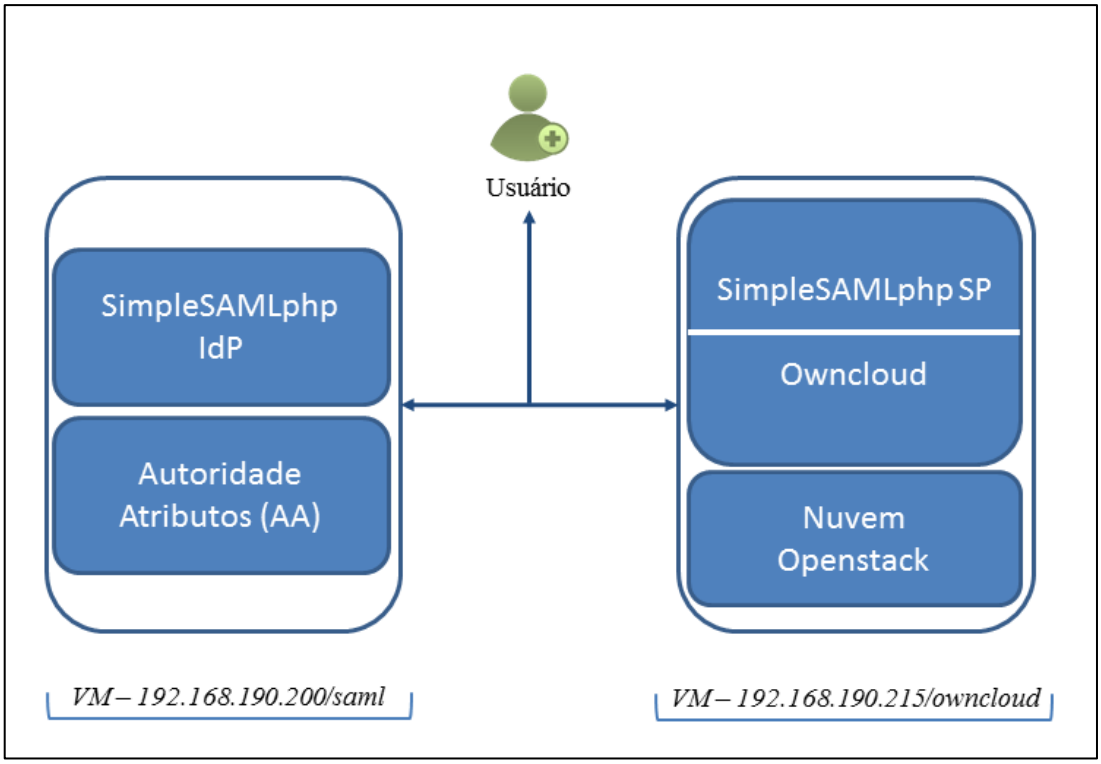
A partir do protótipo busca-se avaliar os fluxos de funcionamento segundo a perspectiva de um usuário. Como o ponto crítico do mecanismo se dá no fluxo de requisição de compartilhamento de registros, esse é o foco da prototipagem desenvolvida, particularmente a etapa de acesso ao armazenamento do registro e aquisição de chave ABE.

4.2. Protótipo com abordagem SAML

Para a elaboração do protótipo SAML seguiu-se a arquitetura representada na Figura 4.1. Os principais componentes de implementação do protótipo são o OpenStack Swift, uma AA, o ownCloud (2015) e o simpleSAMLphp (2015). Conforme ilustrado na Figura 4.1, esses componentes estão em duas instâncias de máquinas virtuais separadas e em rede. O ambiente de implantação consiste em apenas uma máquina física hospedando duas máquinas virtuais (VMs) através do virtualizador VirtualBox (2015). A máquina física possui 8 Gb de memória, processador Intel i5 de 3.2 GHz e sistema operacional Windows 7 de 64 bits. Já cada VM recebeu 4 Gb de memória e sistema operacional Ubuntu 14.04.

O OpenStack Swift é utilizado como nuvem de armazenamento para registros de saúde. A diferença do uso padrão do *swift* para o ambiente do protótipo é que não realiza gerência de usuários, estando encarregado apenas do armazenamento de dados em um único container.

Figura 4.1 – Arquitetura do protótipo com SAML.



Fonte: Produção própria.

O SimpleSAMLphp é uma aplicação em PHP cujo objetivo é o fornecimento de serviços de autenticação. Ele é compatível com diversos protocolos com esse fim, sendo o SAML 2.0 o seu principal foco. Aqui, o SimpleSAMLphp é utilizado nas duas instâncias de VM, tanto no modo de provedor de serviço quanto no modo de provedor de identidades. É no modo IdP dessa aplicação que se gerencia usuários e seus atributos, enquanto que no modo SP é realizado o pedido de autenticação.

Para o protótipo, o código do simpleSAMLphp no modo IdP foi alterado para realizar requisições de criação de chaves à autoridade de atributos. As chaves são então entregues diretamente aos usuários simultaneamente ao redirecionamento de volta à nuvem.

A autoridade ABE é do modo centralizado CP-ABE, uma vez que até o momento do desenvolvimento dessa dissertação implementações de bibliotecas do MA-ABE não estavam disponíveis. Nessa AA, chaves são criadas sob demanda do IdP e de acordo com os atributos repassados.

Outro componente do protótipo é o ownCloud (2015). Ele é uma plataforma que, apesar de possuir versão paga, é de código aberto e pode ser utilizado para a livre criação de pequenas nuvens de uso local ou acesso a nuvens maiores (como OpenStack). O ownCloud possui uma comunidade ativa de colaboradores e tem expandido para incorporar compatibilidade com novas tecnologias e nuvens.

Através de uma extensão no ownCloud é possível utilizar o protocolo SAML para autenticação de usuários através de uma instância simpleSAMLphp em modo SP acoplado ao ownCloud na mesma VM. Quando um usuário se autentica em um IdP pela primeira vez, seus atributos de nome e cota de armazenamento (limite de espaço de armazenamento) são mapeados para um novo usuário no ownCloud que recebe um local de armazenamento no OpenStack Swift dentro do contêiner principal. Após esse primeiro acesso, cada vez que o usuário se autenticar no serviço através do IdP, será mapeado para seus arquivos armazenados.

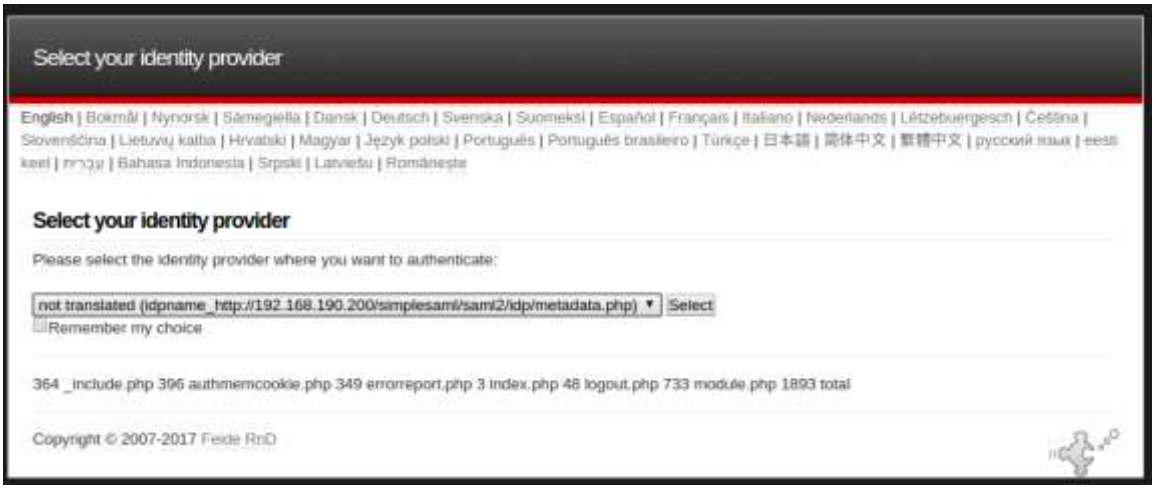
A tela inicial do ownCloud é demonstrada na Figura 4.2. A seguir são instruídos os passos de uso do protótipo.

Figura 4.2 – Tela inicial do ownCloud.

Fonte: Produção própria.

Após escolher se autenticar com SAML, é exibida para o usuário uma tela do simpleSAMLphp modo SP para a seleção de qual IdP ele deseja se autenticar, conforme demonstrado na Figura 4.3.

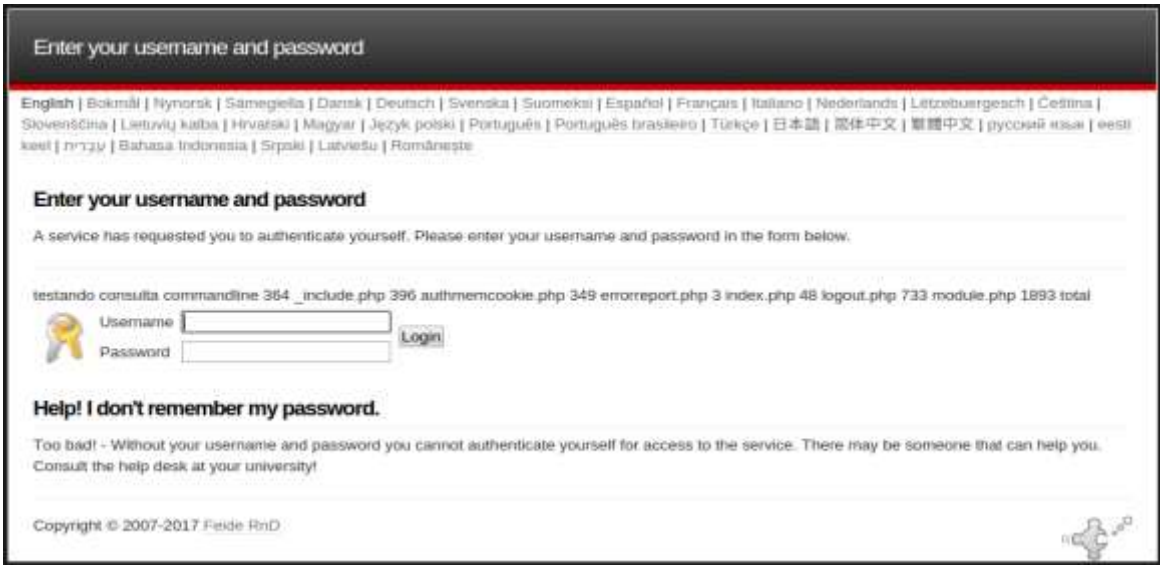
Figura 4.3 – Escolha de IdP.



Fonte: Produção própria.

O IdP (nesse caso, servidor 192.168.190.200) então recebe o pedido de autenticação e exibe uma tela para inserção de credenciais de usuário, conforme a Figura 4.4. Após a inserção de nome e senha de usuário, o IdP verifica a autenticidade e em caso positivo inicia o pedido de K_{ABE} (chave ABE). Esse pedido consiste em executar um programa de criação de chave, que faz uso dos atributos do usuário, além da chave pública e privada da autoridade.

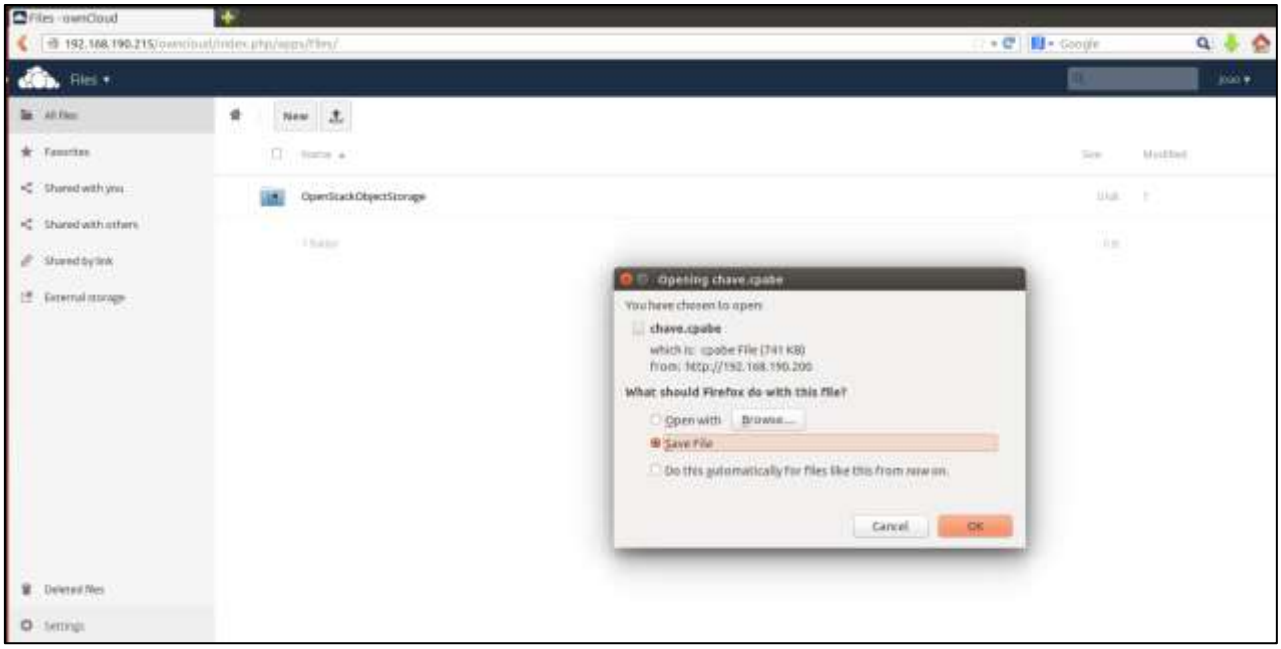
Figura 4.4 – *Login* no IdP.



Fonte: Produção própria.

Após as operações no IdP, o usuário é retornado para o SP, que lhe dá o acesso aos arquivos armazenados na nuvem. De forma simultânea, é oferecida a opção de *download* da chave ABE. Esse *download* é direto do IdP, não estando visível para o SP. A Figura 4.5 apresenta a tela correspondente a esses eventos.

Figura 4.5 – Acesso concedido no SP e *download* de chave ABE.



Fonte: Produção própria.

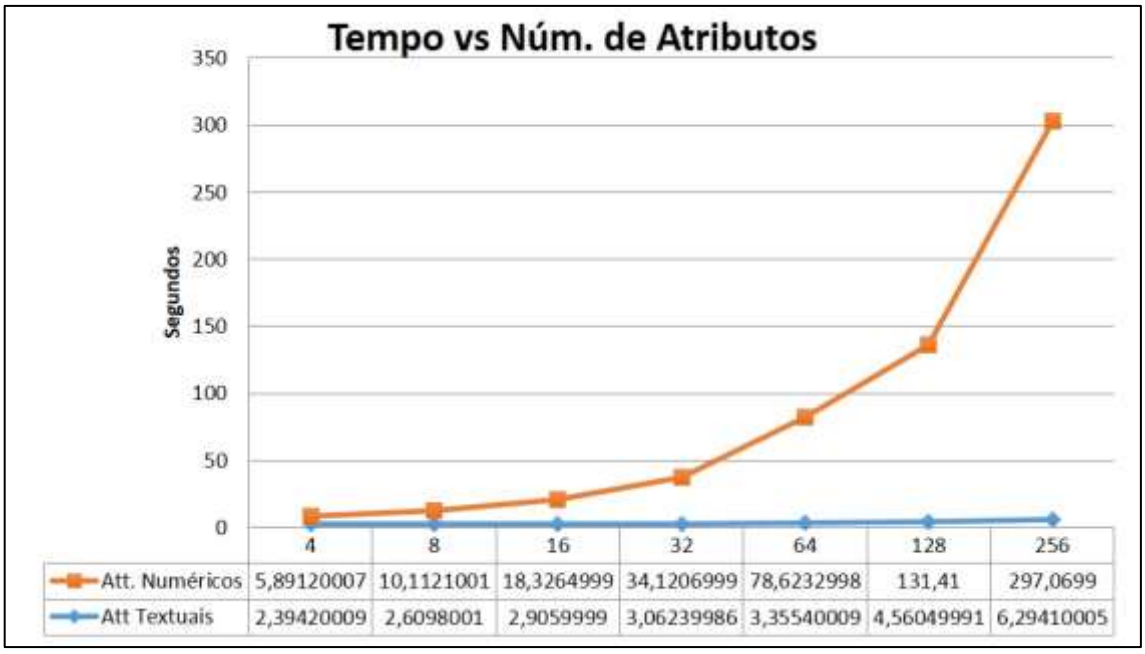
4.2.2. Discussão sobre o protótipo com abordagem SAML

Além de prova de conceito, o protótipo também pôde ser utilizado para a realização de alguns testes. Como a etapa de criação de chave ABE é a que mais demanda recurso computacional, buscou-se analisar o impacto da adição dessa etapa em comparação com a autenticação normal via IdP. Além disso buscou-se analisar o comportamento da proposta através de manipulações de atributos de usuários e consequentemente de chaves ABE.

Em um primeiro teste, coletou-se o tempo de resposta ao usuário da ação inicial de se autenticar até o momento de obter sua chave ABE, desconsiderando o tempo de download da chave. Esse teste foi realizado de duas formas, uma aumentando o número de atributos do tipo texto do usuário (ex.: médico e ortopedista) e outra aumentando o número de atributos

numéricos do usuário (ex.: RG, idade, número de CRM e datas). O resultado, através de médias, está demonstrado pelo gráfico na Figura 4.6.

Figura 4.6 – Impacto do número e natureza de atributos na criação de chaves ABE com até 256 atributos.



Fonte: Produção própria.

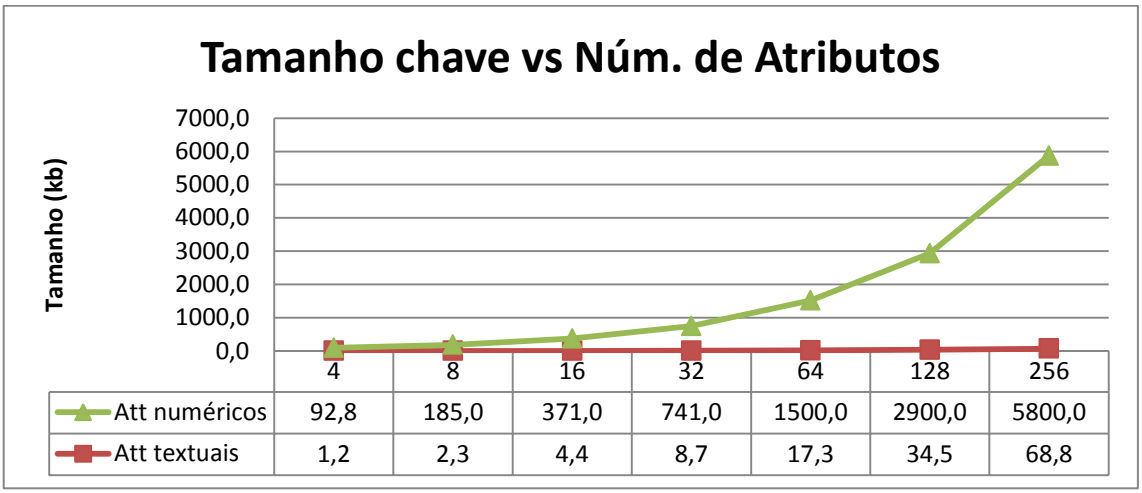
Através desse teste é perceptível o impacto no tempo de espera para usuários obterem suas chaves. Atributos numéricos recebem tratamento diferenciado na implementação utilizada do protocolo ABE, isso para que nas políticas possam ser usadas comparações (como na data de atualização), o que resulta em uma demora de pouco mais de um segundo para cada atributo numérico adicional. Em um cenário real, no entanto, é improvável que usuários cheguem a um número de atributos numéricos elevado. Uma alternativa é tratar os elementos numéricos como texto e assim diminuir o impacto negativo no desempenho, reservando o tipo numérico para atributos que serão utilizados em comparações (ex.: data).

Em contrapartida, atributos textuais possuem pouco impacto ao comparar com o tempo de autenticação federada sem a criação de chave, que em média é de 2.30 segundos (autenticação padrão com IdP sem modificações). Esse impacto ainda é pequeno mesmo quando dobrando sucessivamente o número de atributos. Como atributos textuais são mais

comuns e prováveis para estarem presentes em um cenário real, o impacto da criação de chaves na usabilidade de usuários é mínimo.

O aumento de uso de atributos também tem consequência direta no tamanho das chaves. O gráfico na Figura 4.7 demonstra o crescimento do tamanho das chaves de acordo com o número de atributos. Mais uma vez se faz distinção entre atributos numéricos e textuais.

Figura 4.7 – Impacto do número e natureza de atributos no tamanho de chaves ABE.



	4	8	16	32	64	128	256
Att numéricos	92,8	185,0	371,0	741,0	1500,0	2900,0	5800,0
Att textuais	1,2	2,3	4,4	8,7	17,3	34,5	68,8

Fonte: Produção própria.

É perceptível um comportamento similar ao teste anterior. Atributos numéricos resultam em maior tamanho para chaves ABE, em que é acrescentado em torno de 23 kb no tamanho da chave para cada atributo. Já o acréscimo de atributos textuais resulta em pouco acréscimo no tamanho final da chave. De qualquer forma, mesmo com alto número de atributos numéricos, o tamanho final da chave é desprezível diante da capacidade de armazenamento que dispositivos possuem (5,8 MB para 256 atributos no maior caso).

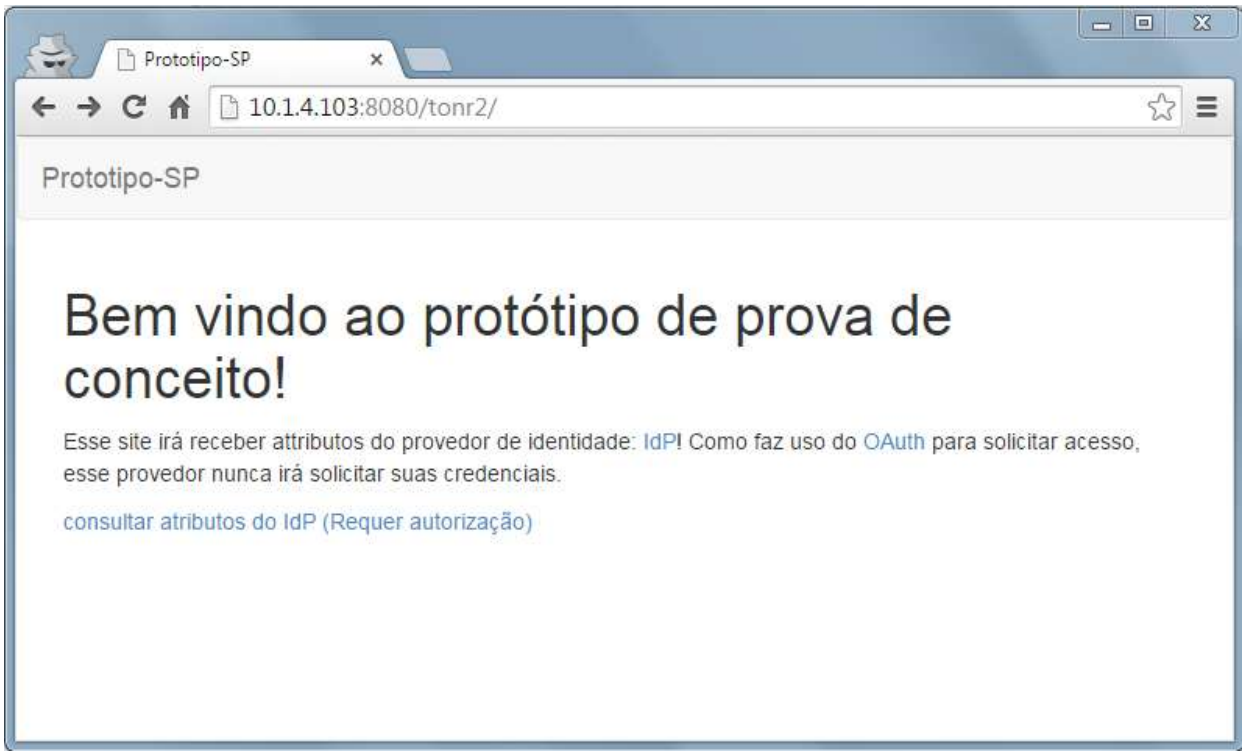
4.3. Protótipo com abordagem oAuth

O protótipo com oAuth segue a abordagem de AAs no papel de SP mencionado na Subseção 3.6. Esse protótipo é mais simples que o anterior e busca apenas demonstrar a viabilidade da proposta com esse protocolo. Sua implantação consiste em duas máquinas

virtuais com a mesma configuração do protótipo anterior. Cada VM possui um servidor web com uma aplicação Java que faz uso da biblioteca Spring (2015) para comunicação oAuth.

Na VM de endereço IP 10.1.4.103 está à aplicação que tem o papel de SP e AA. Essa aplicação é responsável pela criação e provisão de chaves ABE para usuários que solicitarem o serviço. Já na VM de endereço 10.1.4.102 está a aplicação com o papel de IdP que detém os atributos na forma de recursos do oAuth. A Figura 4.8 apresenta a tela inicial do SP.

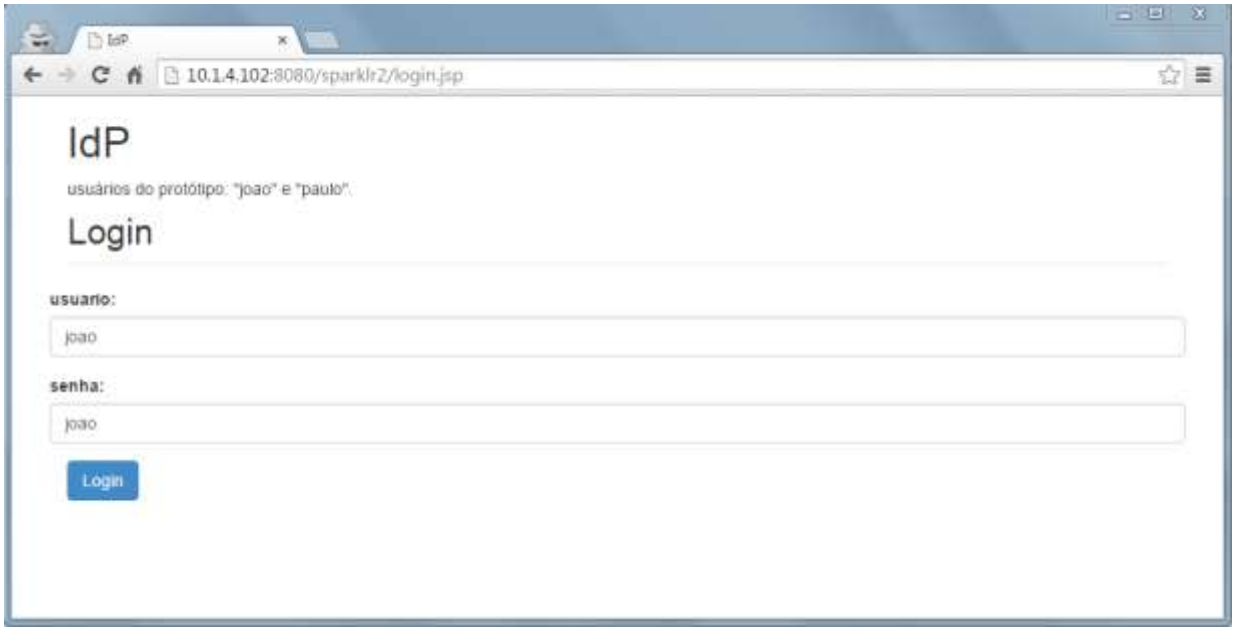
Figura 4.8 – Tela inicial do SP do protótipo oAuth.



Fonte: Produção própria.

Para ser possível solicitar suas chaves, usuários primeiro devem autorizar ao SP o acesso a seus atributos que residem no IdP. Ao selecionarem a opção de consultar seus atributos, o usuário é então redirecionado ao IdP através do protocolo oAuth, onde deve se autenticar (conforme Figura 4.9). Após a autenticação, o usuário autoriza ao SP o acesso aos seus atributos. Por fim são redirecionados ao SP novamente, onde é apresentado a opção de download de sua chave, ilustrado na Figura 4.10.

Figura 4.9 – Tela de autenticação no IdP do protótipo oAuth.



Fonte: Produção própria.

Figura 4.10 – Tela de download da chave ABE no SP oAuth.

Fonte: Produção própria.

O desempenho desse protótipo consiste na média de 1,05 segundos para a autenticação sem a etapa de criação de chaves. Em face dos 2,30 segundos no IdP SAML (protótipo anterior), também sem considerar etapa de criação de chaves, é possível comprovar

o quanto a pilha de protocolo SAML é maior em comparação com o oAuth. Uma vez que os dois protótipos possuem a mesma implementação e configuração de AA, o tempo agregado da etapa de criação de chaves ABE no protótipo oAuth se comporta da mesma forma que o protótipo SAML.

4.4. Considerações Finais

Foram desenvolvidos dois protótipos como prova de conceito da arquitetura proposta por essa dissertação. Esses protótipos são limitados em sua representatividade da arquitetura, no entanto são suficientes para demonstrar o funcionamento do mecanismo na perspectiva do usuário e simular a etapa mais onerosa da arquitetura.

Além disso, através do protótipo SAML também é possível avaliar o impacto das alterações (para se adequar a arquitetura) em uma FIM e SP de nuvem no tempo de resposta ao usuário, e consequentemente também na sua usabilidade. E apesar da criação e entrega de chaves ABE resultar em maior espera de resposta ao usuário, isso é aceitável diante do ganho em segurança. Além disso, uma forma de minimizar esse impacto seria o uso majoritário de atributos textuais em relação aos numéricos, ou mesmo a criação prévia da chave ABE pela autoridade. Nesse último caso, cai sobre a AA a maior responsabilidade e complexidade de gerência dessas chaves.

Conclui-se que o mecanismo proposto no capítulo anterior é viável, inclusive com a adaptação com o protocolo oAuth nas AAs. Apesar disso, ainda é necessário maiores estudos e avaliações sobre um cenário mais robusto e um ambiente de teste com maior capacidade computacional, além de maior número de usuários.

5. CONCLUSÕES

Nesse capítulo são apresentadas as conclusões finais, bem como, as principais contribuições à segurança e privacidade de armazenamento de registros de saúde, as limitações e desafios ainda presentes na solução proposta, como também trabalhos futuros.

5.1. Considerações Finais

E-health tem permitido a área de saúde ser beneficiada pelos avanços tecnológicos para apoiar o cuidado de pacientes e facilitar o serviço de clínicos em geral. Esse é o caso de registros de saúde, com o surgimento dos PHRs e EHRs. Registros eletrônicos de saúde têm possibilitado uma melhoria nos cuidados médicos e gestão de saúde.

O próprio estudo de registros eletrônicos em *e-health* é um assunto complexo e vasto, ainda com muito espaço para aprimoramentos. Os avanços nessa área são constantes e bem vindos para a qualidade de vida das populações em geral. No entanto, mesmo recebendo diversos benefícios, uma preocupação constante é a segurança e privacidade dos dados de pacientes.

Pacientes e usuários de *e-health* em geral não devem ser prejudicados com seus dados sendo manipulados sem consentimento, como em troca de receberem os cuidados médicos. A soberania dos pacientes deve ser garantida e preservada. Prover essa proteção aos dados é um desafio e que merece solução.

Com isso em vista, a proposta dessa dissertação foi a criação de tal solução que viabilizasse o armazenamento de registros de saúde ao mesmo tempo em que seus donos mantivessem o total controle. Através do mecanismo apresentado, usuários tem a capacidade de armazenar seus registros de saúde de forma segura mesmo em um ambiente terceirizado de nuvem, onde não possuem domínio sobre a infraestrutura.

Para apoiar e avaliar a proposta, foram desenvolvidos protótipos de prova de conceito. Esses protótipos demonstraram a viabilidade de uso do mecanismo. Através dos mesmos foi possível visualizar o funcionamento da arquitetura proposta. Como resultado da análise desses protótipos se observou que existe impacto no tempo para a aquisição de chaves criptográficas por parte de usuários, o que potencialmente pode diminuir a usabilidade. No entanto, esse impacto pode ser amenizado através de uso majoritário de atributos em tipo texto. Além do mais, o benefício em segurança justificou o impacto apresentado.

Por fim, a proposta deste trabalho atingiu os seus objetivos secundários e principal. Apesar disso, ainda possui algumas limitações e pontos de melhorias, como descrito na Subseção 5.3.

5.2. Contribuições

A seguir são apresentadas as principais contribuições obtidas durante o desenvolvimento deste trabalho:

- **Nova arquitetura:** uma nova arquitetura que viabiliza armazenamento e compartilhamento seguro de registros de saúde em nuvem, onde a complexidade é delegada para os macros componentes IdP, AA e SP, livrando assim os usuários da gerência de chaves criptográficas e grandes esforços para realizar compartilhamento.
- **Prototipagem:** dois protótipos conceituais foram desenvolvidos e testados, demonstrando a viabilidade da proposta e sua aplicabilidade técnica.
- **Produção de trabalho científico:** houve a publicação de um artigo apresentando a proposta da nova arquitetura (SILVA *et al.*, 2014) no *International Conference on E-health Networking, Applications and Services* (Healthcom).

5.3. Limitações

A principal limitação do uso do mecanismo proposto é que usuários podem decidir ignorar os procedimentos de segurança e escolherem fazer o *upload* de registros não cifrados para a nuvem. O dualismo entre segurança versus usabilidade ainda permanece como um desafio quando lidando com protocolos criptográficos, algo que é componente chave na

arquitetura. As etapas de cifragem e decifragem dos arquivos podem ser vistas como enfadonhas e demoradas para profissionais de saúde e também para pacientes que não compreendem os riscos do armazenamento terceirizado de dados.

Outro desafio está presente na revogação de acesso. Apesar de o mecanismo permitir que um usuário revogue acesso de outros usuários e a data de criação da política ABE ser utilizada para evitar uso de chaves defasadas, a revogação de acesso direcionada a um usuário específico, que ainda satisfaça a política, é limitada. Como por exemplo, não permitir o acesso a um ortopedista específico, enquanto ainda permitir aos outros ortopedistas. O protocolo MA-ABE e a maioria dos outros protocolos ABE não permitem a utilização do operador lógico NOT em relações na política. Revogação de chaves é uma tarefa desafiadora em ABE e tem sido foco de pesquisas.

Mais uma limitação é a ausência de implementações de ABE. Existem algumas bibliotecas, no entanto, que se concentram apenas em CP-ABE, como em JPBC (2015) e em *Advanced Crypto Software Collection* (2015). Mesmo essas são limitadas e em fase de testes. Até o momento da escrita dessa dissertação não se encontrou qualquer implementação de MA-ABE ou mesmo esforços para tal.

Por fim, a própria criação de uma FIM para *e-health* é tecnicamente complexa e possui barreiras políticas. O requisito de colaboração entre as instituições membros agrega vantagens para saúde de pacientes e população em geral, além de vantagens para as próprias entidades através do compartilhamento de serviços. No entanto, interesses comerciais podem motivar empresas de plano de saúde, por exemplo, a não buscarem interoperabilidade e colaboração. Além disso, a FIM tem o potencial de facilitar migração de pacientes entre provedores de assistência de saúde, o que pode desmotivar financeiramente a adesão dessas entidades na federação. A regulamentação de sistemas de saúde e criação de normas de padronização são exemplos de esforços que facilitam a criação de uma FIM para *e-health*.

5.4. Trabalhos Futuros

Espera-se que a proposta aqui apresentada contribua para o avanço da segurança em *e-health*, inclusive motivando novos trabalhos a respeito do mesmo tema e que solucionem os desafios e problemas ainda presentes. Assim, esta seção sugere alguns possíveis avanços no trabalho aqui apresentado.

5.4.2. Aprimoramento da prototipagem

Os protótipos se limitaram a uma etapa essencial do mecanismo proposto. No entanto, outras etapas do mecanismo podem ser avaliadas. Além disso, seria interessante incorporar novas funcionalidades relacionadas à manipulação de registro por parte de usuários, a fim de possibilitar um teste mais aplicado ao uso de EHR e PHR.

5.4.3. Estudo de caso em cenário real

Para avaliar o comportamento e adequabilidade do mecanismo, é desejável a realização de um estudo de caso contextualizado em um cenário real de uso por pacientes e profissionais de saúde. O desafio de usabilidade e questões práticas do dia-a-dia do contexto de saúde seriam assim mais bem compreendidas para a avaliação da proposta. Em conjunto com a prototipagem, esse estudo permitiria a coleta de resultados e análise da arquitetura, posteriormente levando a aprimoramentos.

5.4.4. Estudo e análise de protocolos ABE

A Arquitetura possui um nível de generalização para o uso de criptografia ABE. Apesar disso, poderia se beneficiar com uma análise mais profunda e comparações entre as diferentes propostas de ABE, inclusive as de mesma categoria. Como por exemplo, identificar qual protocolo de MA-ABE se comporta melhor no mecanismo.

5.4.5. Clientes inteligentes

Em apoio à proposta dessa dissertação é necessária a criação de clientes inteligentes, que auxiliam usuários nas operações descritas no mecanismo. Por usuários abrangerem uma vasta gama de perfis (idosos, debilitados, etc.), são necessárias funcionalidades que apoiam a usabilidade. Operações de criptografia, seleção de atributos para políticas ABE e a própria

criação de políticas ABE são exemplos de itens que devem ser facilitados e beneficiados por programas clientes.

REFERÊNCIAS BIBLIOGRÁFICAS

ADVANCED CRYPTO SOFTWARE COLLECTION. (2015). Disponível em: <http://acsc.cs.utexas.edu/cpabe/>. Acesso em 20 de Agosto de 2015.

ALSHEHRI, S., RADZISZOWSKI, S. P., & RAJ, R. K. (2012). **Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption**. IEEE 28th International Conference on Data Engineering Workshops (ICDEW) (pp. 143-146), 2012.

AMAZON WEB SERVICES. AMAZON SIMPLE STORAGE SERVICE. (2015). Disponível em: <http://aws.amazon.com/s3/>. Acesso em 20 de Agosto 2015.

AMAZON WEB SERVICES. **Amazon S3 - 905 Billion Objects and 650,000 Requests/Second. Amazon Web Services Blog** (2012). Disponível em: <http://aws.typepad.com/aws/2012/04/amazon-s3-905-billion-objects-and-650000-requestssecond.html>. Acesso em: 20 de Agosto 2015.

AUTHENTICATION WORLD. **What is SSO?**. Disponível em <http://www.authenticationworld.com/Single-Sign-On-Authentication/>. Acesso em: 28 de Setembro, 2015.

BETHENCOURT, J., SAHAI, A., WATERS, B. (2007). **Ciphertext-policy attribute-based encryption**. IEEE Symposium on Security and Privacy, SP'07. (pp. 321-334), 2007.

BHADAURIA, R.; SANYAL,S.. **A Survey on security issues in cloud computing and associated mitigation techniques**. International Journal of computer applications, vol. 47, 2012.

BRODKIN, J. GARTNER: **Seven cloud-computing security risks**. Networkworld, (2008). Disponível em: <http://www.networkworld.com/news/2008/070208-cloud.html>. Acesso em: 20 Agosto 2015.

CAFe. **Comunidade academica federada**. Disponível em: <http://portal.rnp.br/web/servicos/cafe>. Acesso em 20 de Agosto de 2015.

CHADWICK, D. W., SIU, K., LEE, C., FOUILLAT, Y., e GERMONVILLE, D.. (2013). **Adding federated identity management to openstack**. Journal of Grid Computing, pp. 1–25, 2013.

CLOUD SECURITY ALLIANCE. **Cloud Security Alliance: Top Threats to Cloud Computing V1.0**. CloudSecurityAlliance (2010). Disponível em: <https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf>. Acesso em 20 de Agosto de 2015.

COMPANIES SUPPORTING THE OPENSTACK FOUNDATION. (2015). Disponível em: <http://www.openstack.org/foundation/companies/>. Acesso em 20 de Agosto de 2015.

DIXIT, G. N., & SURESH, M. B. (2013). **Patient Centric Frame Work For Data Access Control Using Key Management In Cloud Server**. In International Journal of Engineering Research and Technology (Vol. 2, No. 4 (2013). ESRSA Publications.

DROPBOX. (2015). Disponível em: <https://www.dropbox.com/>. Acesso em 20 de Agosto de 2015.

DURKEE, D. (2010). **Why cloud computing will never be free**. Communications of the ACM, New York, v. 53, n. 5, p. 62-69, Maio 2010.

EVERETT, C. (2009**). Cloud computing–A question of trust**. Computer Fraud & Security, 2009(6), 5-7.

FERNÁNDEZ-ALEMÁN, J. L., SEÑOR, I. C., LOZOYA, P. Á. O., e TOVAL, A. (2013). **Security and privacy in electronic health records: A systematic literature review**. Journal of biomedical informatics, 541-562.

FORMISANO,C., KOLODNER, E. K., SHULMAN-PELEG,A., TRAVAGLINO, E., VERNIK,G., e VILLARI, M. (2014). **Delegation across storage clouds: onboarding federation as a case study.** Scalable Computing: Practice and Experience, vol. 14, no. 4, 2014.

FOX, A. et al. (2009) **Above the clouds: A Berkeley view of cloud computing**. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley.

GAVRILOV, G., e TRAJKOVIK, V. (2012). **Security and privacy issues and requirements for healthcare cloud computing**. ICT Innovations, 143-152.

GILBERT, S.; LYNCH, N. (2002). **Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services**. ACM SIGACT News, Nova York, v. 33, n. 2, p. 51-59, Junho 2002. ISSN 0163-5700.

GOLDMAN, JANLORI, e ZOE HUDSON. (2000). **Virtually exposed: privacy and e-health**. Health Affairs 19.6 (2000): 140-148.

GOOGLE APP ENGINE. (2015). Disponível em: <https://developers.google.com/appengine/>. Acesso em 20 de Agosto de 2015.

GOOGLE DRIVE. (2015). Disponível em: <https://drive.google.com/>. Acesso em 20 de Agosto de 2015.

GOVERNEMENT EXECUTIVE (2014). **Why Cloud computing is like water**. Disponível em: <http://www.govexec.com/contracting/2014/03/why-cloud-computing-water/80641/>. Acesso em 20 de Setembro de 2015.

GREENBERG, A. (2008). **Cloud Computing's Stormy Side**. Forbes Magazine, 19 Fevereiro 2008. Disponível em: <http://www.forbes.com/2008/02/17/web-application-cloud-tech-intel-cx_ag_0219cloud.html>. Acesso em: 27 Agosto 2015.

GUO, Y., KUO, M. H., E SAHAMA, T. (2012). **Cloud computing for healthcare research information sharing**. In Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on (pp. 889-894). IEEE.

InCommon. **Security, privacy and trust for the research and education community**. Disponível em <https://incommon.org/>. Acesso em 20 de Agosto de 2015.

JPBC (2015). Disponível em <http://gas.dia.unisa.it/projects/jpbc/index.html#.VddxB_lViko>. Acesso em 20 de Agosto de 2015.

LI, M., YU, S., ZHENG, Y., REN, K., & LOU, W. (2013). **Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption**. Parallel and Distributed Systems, IEEE Transactions on, 24(1), 131-143.

LI, M., YU, S., REN, K., E LOU, W. (2010). **Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings**. In Security and Privacy in Communication Networks (pp. 89-106). Springer Berlin Heidelberg.

LIU, F., RIJNBOUTT, E., ROUTSIS, D., VENEKAMP, N., FULGENCIO, H., REZAI, M., e VAN DER HELM, A. (2013). **What challenges have to be faced when using the cloud for e-health services?.** System 6.

LÖHR, H., SADEGHI, A. R., & WINANDY, M. (2010). **Securing the e-health cloud**. In Proceedings of the 1st ACM International Health Informatics Symposium (pp. 220-229). ACM.

LEE, C. C., CHUNG, P. S., & HWANG, M. S. (2013). **A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments**. IJ Network Security, 15(4), 231-240.

LEWKO, A., e WATERS, B. (2011). **Decentralizing attribute-based encryption**. Advances in Cryptology–EUROCRYPT 2011 (pp. 568-588), Springer, 2011.

MAGGIANI, R. (2009). **Cloud computing is changing how we communicate**. In Professional communication conference, 2009. IPCC 2009. IEEE international (pp. 1-4). IEEE.

MELL, P.; GRANCE, T. (2010). **The NIST Definition of Cloud Computing**. Disponivel em: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. Acesso em: 20 Agosto 2015.

MICROSOFT HEALTHVAULT. Disponível em: <http://www.healthvault.com>. Acesso em 20 de Agosto de 2015.

MUSTHAFA, S., STUDENT, M. T., & SUDARSA, D. B. (2013). **Patient–Centric Secure Data Sharing Frame Work for Cloud-Based PHR Systems**. International Journal of Engineering Science Invention, 17-26.

NIMBUS. **Nimbus: Cloud Computing for Science**. Disponível em: <http://nimbusproject.org>. Acesso em 20 de Agosto de 2015.

NIWA, Y., KANAOKA, A., & OKAMOTO, E. (2013). **Construction of a Multi-domain Functional Encryption System on Functional Information Infrastructure**. In Network-Based Information Systems (NBiS), 2013 16th International Conference on (pp. 105-112). IEEE.

NURMI, D. C. et al. (2009). **The Eucalyptus Open-Source Cloud-Computing System. Cluster Computing and the Grid**. [S.l.]: IEEE/ACM. 2009. p. 124 - 131.

NZANYWAYINGOMA, F., & HUANG, Q. (2012). **Securable Personal Health Records using ciphertext policy attribute based encryption**. In e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on (pp. 502-505). IEEE.

OASIS. **Advancing open standards for the information society**. Disponível em:<https://www.oasis-open.org/>. Acesso em 20 de Agosto de 2015.

OAUTH 2. Disponível em:<http://oauth.net/2/>. Acesso em 20 de Agosto de 2015.

OECD. **National strategies and policies for digital identity management in oecd countries**. Disponível: <http://dx.doi.org/10.1787/5kgdzvn5rfs2-en>. Acesso em 20 de Agosto de 2015.

ONEDRIVE. Disponível em: <https://onedrive.live.com/>. Acesso em 20 de Agosto de 2015.

OPENNEBULA. **Enterprise Cloud and Datacenter Virtualization**. Disponível em: <http://opennebula.org/>. Acesso em 20 de Agosto de 2015.

OPENSHIFT Redhat (2015). Disponível em: <https://www.openshift.com/>. Acesso em 20 de Agosto de 2015.

OPENSTACK. **Open source software for building private and public clouds.** Disponível em: <http://www.openstack.org/>. Acesso em 20 de Agosto de 2015.

OWNCLOUD. Disponível em:<https://owncloud.org/>. Acesso em 20 de Agosto de 2015.

PEARSON, S. (2009). **Taking Account of Privacy when Designing Cloud Computing Services**. Software Engineering Challenges of Cloud Computing (CLOUD '09). Vancouver: IEEE. 2009. p. 44-52.

ROBISON, J., BAI, L., MASTROGIANNIS, D. S., Tan, C. C., & Wu, J. (2012). **A survey on PHR technology**. In e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on (pp. 184-189). IEEE.

RAM, S. **Insider Attack – No one is safe**. Maravis. (2010). Disponível em: <http://www.maravis.com/insider-attacks-no-one-is-safe>. Acesso em 20 de Agosto de 2015.

SABAHI, F. (2011). **Cloud Computing Security Threats and Responses**. International Conference on Communication Software and Networks (ICCSN). Xi'an. p. 245-249.

SAHA, A.; DAS, A. (2012). **A Detailed Analysis of the Issues and Solutions for Securing Data in Cloud**. IOSR Journal of Computer Engineering (IOSRJCE), Índia, v. 4, n. 4, p. 11-18, Set-Out. 2012. ISSN 2278-0661.

SAHAI, A., WATERS, B. (2005). **Fuzzy Identity Based Encryption**. In Advances in Cryptology – Eurocrypt, volume 3494 de LNCS, páginas 457–473. Springer, 2005.

SAHAMA, T., SIMPSON, L., e LANE, B. (2013). **Security and Privacy in eHealth: Is it possible?**. In e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on (pp. 249-253). IEEE.

SAML SPECIFICATIONS. **SAML Specifications**. Disponível em: <http://saml.xml.org/saml-specifications>. Acesso em 20 de Setembro de 2015.

SECURITY FOR BUSINESS INNOVATION COUNCIL. (2013). **Information Security Shake-up: Disruptive Innovations to Test Security's Mettle in 2013**. Disponível em: <http://www.emc.com/collateral/industry-overview/h11391-rpt-information-security-shake-up.pdf>. Acesso em 20 de Agosto de 2015.

SEMPOLINSKI, P.; Thain, D. (2010). **A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus**. 2nd IEEE International Conference on Cloud Computing Tecnology and Science. [S.l.]: [s.n.]. 2010.

SILVA, E. L.; MENEZES, E. M. (2001). **Metodologia da Pesquisa e Elaboração da Dissertação**. 3. ed. rev. Atual – Laboratório de Ensino a Distância da UFSC. Florianópolis, Brasil, 2001.

SIMPLESAMLPHP. Disponível em: <https://simplesamlphp.org/>. Acesso em 20 de Agosto de 2015.

SPRING. Disponível em: <http://spring.io/>. Acesso em 20 de Agosto de 2015.

STALLINGS, W. **Cryptography and Network Security Principles and Practices**. 4. ed. [S.l.]: Prentice Hall, 2005. 31 p.

TAN, X.; AIB, B. (2011). **The Issues of Cloud Computing Security in High-speed Railway**. International Conference on Electronic & Mechanical Engineering and Information Technology (EMEIT). p. 4358 - 4363.

TASSANAVIBOON, A., & GONG, G. (2011). **Oauth and ABE based authorization in semi-trusted cloud computing: aauth**. In Proceedings of the second international workshop on Data intensive computing in the clouds (pp. 41-50). ACM.

TSENG, F. K., E CHEN, R. J. (2012). **Enabling Searchable Dynamic Data Management for Group Collaboration in Cloud Storages**. In Proc. 2012 Int'l Conf. on Security and Management (SAM'12).

WAINER, JACQUES.(2007). **Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação**. Atualização em Informática. Org: Tomasz Kowaltowski; Karin Breitman. Rio de Janeiro: Ed. PUC-Rio (2007).

WINDOWS AZURE. Disponível em: <http://azure.microsoft.com/en-us/>. Acesso em 20 de Agosto de 2015.

VIRTUALBOX. Disponível em: <https://www.virtualbox.org/>. Acesso em 20 de Agosto de 2015.

ZHU, S., YANG, X., WU, X. (2013). **Secure Cloud File System with Attribute Based Encryption**. International Conference on Intelligent Networking and Collaborative Systems (INCoS)(pp. 99-102), 2013.