UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MECANISMO DE GARANTIA DE PRIVACIDADE PARA APLICAÇÕES EM REDES ORIENTADAS A CONTEÚDO

ROAN SIMÕES DA SILVA

ORIENTADOR: PROF. DR. SÉRGIO DONIZETTI ZORZO

São Carlos - SP
Abril/2016

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MECANISMO DE GARANTIA DE PRIVACIDADE PARA APLICAÇÕES EM REDES ORIENTADAS A CONTEÚDO

ROAN SIMÕES DA SILVA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos e Redes.
Orientador: Dr. Sérgio Donizetti Zorzo

São Carlos - SP
Abril/2016

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

Silva, Roan Simões da
S586m Mecanismo de garantia de privacidade para aplicações em redes orientadas a conteúdo / Roan Simões da Silva. -- São Carlos : UFSCar, 2016.
79 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2016.

1. Redes orientadas a conteúdo. 2. Named-data networking. 3. Privacidade. 4. Criptografia baseada em atributos. I. Título.

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Roan Simões da Silva, realizada em 08/03/2016:

Prof. Dr. Sergio Donizetti Zorzo
UFSCar

Prof. Dr. Helio Crestana Guardia
UFSCar

Prof. Dr. Rodrigo Palucci Pantoni
IFSP - Sertãozinho

Aos meus pais.

AGRADECIMENTO

Agradeço ao meu orientador, Dr. Sérgio Donizetti Zorzo pela oportunidade e confiança depositada em mim como seu aluno.

Agradeço aos meus pais e a Lilian, que sempre me apoiaram e me deram força para superar os momentos difíceis.

Agradeço também os colegas do laboratório Priv&Person e de curso que ajudaram com ideias e companhia nos estudos.

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

José de Alencar

RESUMO

A utilização atual da Internet difere muito em relação à sua concepção inicial. Em geral, os usuários a utilizam para acessar e compartilhar conteúdos e não se importam pela localidade física dos dados associados. As Redes Orientadas a Conteúdo surgem como uma proposta para modificar o modo de operação da arquitetura atual da Internet, onde o roteamento passa a ser baseado no conteúdo e não no endereçamento. As Redes Orientadas a Conteúdo buscam tornar a Internet mais eficiente e segura, mas por estarem ainda em desenvolvimento, deixam em aberto a solução para muitas questões de privacidade. O direito à privacidade dos usuários deve ser respeitado pelas aplicações e seu conceito pode abranger diferentes aspectos. Este trabalho considera a privacidade como o direito do usuário controlar quem poderá acessar seus dados. Deste modo, este trabalho tem como objetivo propor um mecanismo de garantia de privacidade para aplicações em Redes Orientadas a Conteúdo, com o intuito de permitir que um usuário publicador defina quais usuários poderão acessar seus conteúdos. Para garantir a confidencialidade destes conteúdos e, consequentemente, a privacidade do usuário publicador, o mecanismo proposto faz uso de uma técnica de criptografia baseada em atributos, chamada CP-ABE, que permite que sejam definidas políticas de acesso que são armazenadas no próprio conteúdo. Para viabilizar a revogação de privilégios imediata, é inserido um servidor *Proxy* que atua no processo de descriptografia. Como prova de conceito da viabilidade do mecanismo proposto, foi desenvolvida uma aplicação de troca de arquivos criptografados de acordo com o mecanismo, adotando uma política de acesso que limita a descriptografia somente a usuários autorizados. Esta aplicação foi executada em um simulador da arquitetura *Named-Data Networking*, chamado ndnSIM, visando a análise da viabilidade de sua implementação em termos de desempenho. Testes de desempenho em relação às principais funções do sistema foram realizados, com o intuito de determinar a viabilidade e limitações do mecanismo. Os testes analisaram o tamanho dos arquivos após a criptografia, o tempo de processamento e o consumo de memória RAM. Com os testes conclui-se que o mecanismo é viável em termos de desempenho.

Palavras-chave: Redes Orientadas a Conteúdo, *Named-Data Networking*, Privacidade, Criptografia Baseada em Atributos.

ABSTRACT

The current use of the Internet differs greatly in relation to its initial design. Internet users are becoming interested in accessing and sharing content regardless of their physical location. For future Internet, information-centric networking is considered a potential solution to many of its current problems. Information-centric networking treats content as the main element in the architecture rather than the host location. Information-centric networking is intended in becoming Internet most efficient and safe, however, as it is still under development, it leaves open the solution to many privacy issues. The privacy concept may cover many different aspects and must be respected by applications. In this work the privacy is considered as the right of the user to control who can access your data. Thus, this work aims to propose a mechanism for applications in Information-centric networking that allow a publisher user to define which users can access their content. To ensure the content confidentiality and hence the user's privacy publisher, the proposed mechanism uses an attribute-based encryption technique, called CP-ABE, which allows the use of access policies that are defined and stored in the content. To enable the immediate revocation of privileges, it is inserted a proxy server that operates in the decryption process. As a proof of concept of the feasibility of the proposed mechanism, an application to share encrypted file was developed. The application adopts an access control policy that limits the decryption only by authorized users. This application was performed on a simulator of the Named-Data Networking architecture, called ndnSIM. Performance tests against major system functions have been performed in order to determine the feasibility and limitations of the mechanism. The tests analyzes the file size after encryption, processing time and RAM memory consumption. The tests concluded that the mechanism is viable in terms of performance.

Palavras-chave:. Information-Centric Networking;, *Named-Data Networking*, Privacy, Atributte-based encryption.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABE - *Attribute-Based Encryption*

API - *Application Programming Interface*

CCN - *Content Centric Networking*

CP-ABE - *Ciphertext-Policy Attribute-Based Encryption*

CS - *Content Store*

DRM - *Digital Rights Management*

FIB - *Forward Information Base*

IP - *Internet Protocol*

KP-ABE - *Key-Policy Attribute-Based Encryption*

NDN - *Named Data Networking*

NSF-FIA - *National Science Foundation – Future Internet Architecture*

OSPF - *Open Shortest Path First*

OSPFN - *Open Shortest Path First for Named Data*

P2P - *Peer-to-Peer*

PIT - Pending Interest Table

ROC - Rede Orientada a Conteúdo

SDE - *Searchable Data Encryption*

TCP - *Transmission Control Protocol*

TLS - Transport Layer Security (TLS)

VP - *Virtual Private Comunity*

SUMÁRIO

Capítulo 1

INTRODUÇÃO

Este capítulo apresenta o contexto onde a proposta deste trabalho está inserida e apresenta a motivação para o seu desenvolvimento, os objetivos e a forma como este trabalho está organizado.

1.1 Contexto

A arquitetura da Internet foi inicialmente desenvolvida para suprir propósitos muito diferentes dos que direcionam o seu uso atual. O objetivo original era interligar poucos pontos fixos em um ambiente confiável, prioritariamente para a transferência de arquivos, os quais em sua maioria continham apenas texto (WETHERALL; TANEMBAUM, 2011).

Porém, a flexibilidade do conjunto de protocolos TCP/IP possibilitou a adaptação dessa arquitetura por várias décadas, permitindo a inclusão de uma diversidade de aplicações que não haviam sido planejadas inicialmente, mas que atualmente são executadas sobre essa antiga arquitetura.

Aplicações, como as ferramentas colaborativas da chamada Web 2.0, permitiram uma revolução no compartilhamento de conhecimento, uma vez que qualquer usuário com acesso à Internet pode publicar seus próprios conteúdos e torná-los acessíveis a outros usuários (O'REILLY, 2007).

Além disso, o modelo par-a-par (*Peer-to-Peer - P2P)* introduziu aplicações, como o *bittorrent*, que permitem que usuários compartilhem conteúdo entre si, sem depender de um servidor intermediando toda a transmissão.

Segundo relatório da empresa Cisco (2012), até 2018 o tráfego global somente de dados móveis deve alcançar 190 Exabytes, tornando-se 190 vezes maior que todo o tráfego IP (incluindo fixo e móvel) gerado no ano de 2000. A maioria deste tráfego se refere a conteúdo multimídia, como imagens e vídeos, publicados e compartilhados pelos próprios usuários.

Considerando este volume de tráfego o atual esquema de roteamento da Internet, baseado em endereçamento, passou a ser uma barreira ao acesso do que realmente interessa ao usuário, o conteúdo e não sua localização (DIBENEDETTO ET AL., 2012 e BRITO; VELOSO; MORAES, 2012).

Este problema ocorre, pois, para um usuário ser capaz de acessar algum conteúdo publicado na Internet, é necessário descobrir primeiramente o local onde o mesmo foi publicado, para só então obter acesso ao conteúdo em si. Para que isso ocorra, é necessário realizar antes mapeamentos de endereço de enlace e de rede, além da resolução de nomes. Em redes móveis, o problema se agrava ainda mais, pois o esquema de infraestrutura da Internet foi planejado para pontos fixos.

Outro problema recorrente da arquitetura atual da Internet está relacionado com a segurança dos dados e, mais especificamente, com a privacidade dos usuários. Como a previsão inicial da Internet era de ser um ambiente confiável, a segurança não foi uma preocupação de seus criadores (PAUL ET AL., 2011).

Diante destes problemas, diversas soluções para uma nova arquitetura da Internet passaram a ser pesquisadas nos últimos anos. Entre essas propostas, as Redes Orientadas a Conteúdo (ROC) têm sido alvo de estudo por diversos grupos de pesquisa em todo o mundo, e considerada uma solução viável para o futuro da Internet (PASSARELA, 2012).

1.2 Motivação e Objetivos

Entre as principais características necessárias para as novas arquiteturas da Internet estão a segurança e privacidade desde a sua concepção, sendo estas premissas dos principais projetos de arquiteturas para a Internet do futuro. Tais pontos também são requisitos dos projetos subsidiados pela NSF-FIA (Fundação Nacional

de Ciência – Arquitetura da Internet do Futuro, ou no original em inglês, *National Science Foundation – Future Internet Architecture*).

Acs et al. (2013) e Chaabane et al. (2013) destacam que as pesquisas dos problemas e soluções sobre o tema privacidade ainda estão incipientes em relação às ROC's. Este fato é corroborado em pesquisas como as de Mohaisen et al. (2015) e Tourani et al. (2016) que demonstram a existência de diversos riscos à privacidade em ROC´s, os quais ainda precisam de soluções.

Este trabalho tem como objetivo principal propor um mecanismo de garantia de privacidade para aplicações em ROC. O mecanismo proposto deve permitir a um usuário publicar conteúdos que serão legíveis somente a outros usuários que tenham sido previamente autorizados pelo usuário publicador.

O mecanismo deve ser capaz de garantir a confidencialidade do conteúdo dos pacotes, tornando os dados ilegíveis aos usuários não autorizados. Consequentemente, o mecanismo permitirá à aplicação manter a privacidade do usuário publicador, pois irá lhe garantir o direito de decidir quem tem privilégio para acessar seus dados.

Este mecanismo de garantia de privacidade funcionará no nível de aplicação de uma ROC e fará uso de um esquema de criptografia baseada em atributos.

Como requisito final, o mecanismo permitirá que sejam acrescidos recursos de auditoria caso necessário, uma vez que a aplicação terá controle sobre o processo de criptografia, o que impede a distribuição totalmente anônima de conteúdos.

Como prova de conceito da viabilidade do mecanismo, foi desenvolvida uma aplicação a ser executada na arquitetura *Named-Data Networking* (NDN), através do simulador ndnSim. A viabilidade foi aferida através de testes de desempenho nas principais funções do mecanismo, através da medição do tamanho dos arquivos após a criptografia, o tempo de processamento e o consumo de memória RAM.

1.3 Organização do Trabalho

Este trabalho está dividido em seis capítulos, que serão descritos resumidamente a seguir.

No capítulo 1 há uma introdução e contextualização do ambiente onde está inserida a proposta, além da motivação para a sua realização.

No capítulo 2 é apresentado um levantamento bibliográfico sobre os conceitos principais deste trabalho. Entre os tópicos, estão as redes orientadas a conteúdo, através da arquitetura CCN e do projeto NDN, o conceito de privacidade aplicado ao contexto de desenvolvimento do trabalho, e a criptografia baseada em atributos, uma técnica de criptografia que será utilizada pelo mecanismo proposto.

No capítulo 3 são apresentados os trabalhos existentes na literatura que se correlacionam com o que é proposto por este projeto. O intuito do capítulo é contextualizar e identificar a complementaridade deste trabalho com o estado da arte.

No capítulo 4 o mecanismo de garantia de privacidade proposto é descrito.

No capítulo 5, são demonstrados a forma como foi realizada a prova de conceito, bem como os resultados obtidos com os testes.

No capítulo 6 é apresentada a conclusão do trabalho.

Capítulo 2

REDES ORIENTADAS A CONTEÚDO, PRIVACIDADE E CRIPTOGRAFIA

Neste capítulo são discutidos conceitos primordiais para o desenvolvimento do trabalho, como as redes orientadas a conteúdo, a privacidade e o uso da criptografia como técnica de segurança para garantir a confidencialidade dos dados.

.

2.1 Considerações Iniciais

O objetivo deste capítulo é apresentar os conceitos básicos dos principais tópicos que são tratados neste trabalho.

Inicialmente, na seção 2.2, é apresentada a conceituação das Redes Orientadas à Conteúdo (ROC) e a arquitetura do projeto *Named-Data Networking* (NDN), que é o contexto onde o mecanismo de garantia de privacidade proposto neste trabalho será aplicado. A descrição do funcionamento do simulador ndnSIM, utilizado no estudo de caso, encerra a discussão sobre as redes ROC.

Em seguida, a abordagem considerada por este trabalho em relação aos aspectos de privacidade e segurança é apresentada na seção 2.3.

Na seção 2.4, são apresentados conceitos básicos sobre criptografia e, nas subseções seguintes, a criptografia baseada em atributos, que é utilizada como recurso de segurança pelo mecanismo de garantia de privacidade proposto neste trabalho.

Por fim, a seção 2.5 apresenta as considerações finais, demonstrando como os tópicos apresentados anteriormente se relacionam entre si neste trabalho.

2.2 Redes Orientadas a Conteúdo

As redes orientadas a conteúdo (ROC) constituem uma promissora arquitetura para tornar as redes de computadores mais flexíveis, eficientes e seguras (PAUL; PAN; JAIN, 2011).

A motivação da proposta das ROC se baseia no fato de que a principal utilização da Internet é a distribuição de conteúdo, independente da sua localização. Isso tem resultado na proliferação de serviços par-a-par (*Peer-to-Peer - P2P)* e de Redes de Distribuição de Conteúdos (*Content Delivery Networks - CDN)* (AHLGREN ET AL., 2013).

Deste modo, uma ROC passa a tratar o conteúdo como elemento principal da rede, ao invés do endereçamento, como na Internet atual. Na Figura 2.1, é apresentada uma comparação das arquiteturas da Internet com uma proposta de ROC.

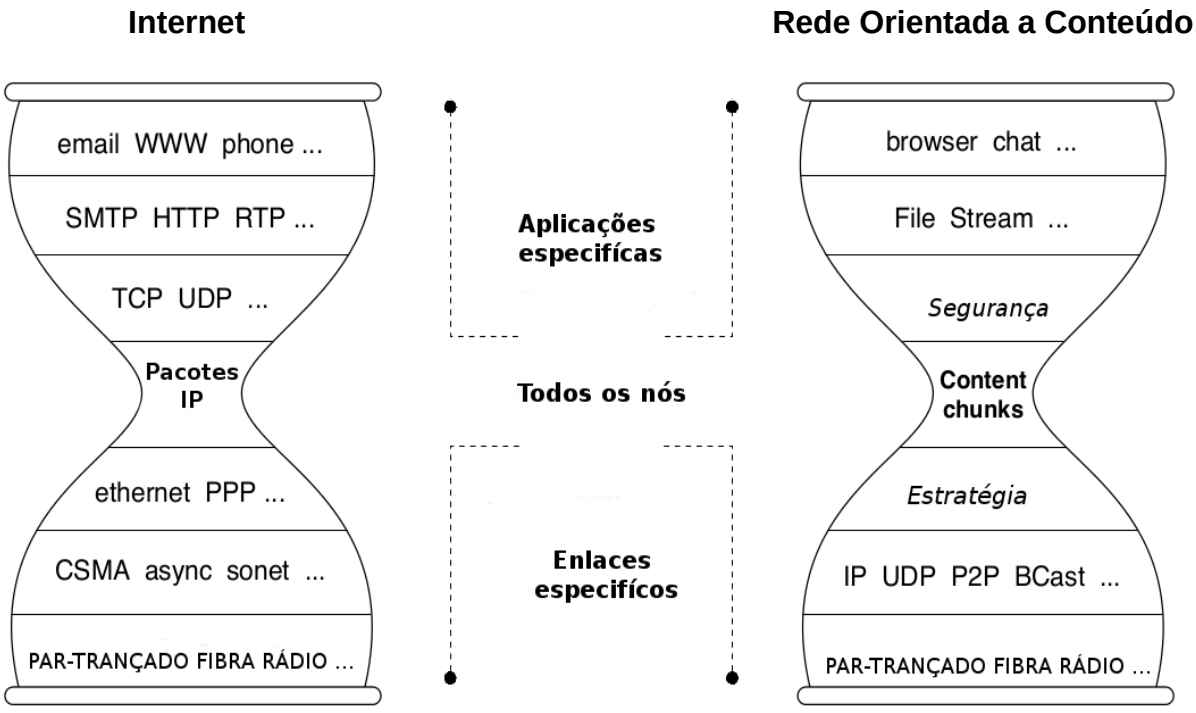

Figura 2.1 - Comparação da arquitetura da Internet com uma ROC

Fonte: Adaptado de Zhang et al. (2014)

Como é possível visualizar na Figura 2.1, as aplicações e os meios físicos continuam sendo, respectivamente, as extremidades superior e inferior em ambas as arquiteturas. Entretanto, o centro das arquiteturas muda do foco no protocolo IP, na Internet atual, para o conteúdo, em um ROC.

Entre as arquiteturas ROC existentes, destaca-se a *Content Centric Networking* (CCN), proposta por Jacobson et al. (2009), e que tem sido objeto de estudo por inúmeros grupos de pesquisa no mundo todo. CCN é a arquitetura utilizada por este trabalho e foi descrita nas próximas seções.

2.2.1 Estrutura de nomes CCN

Na arquitetura CCN é utilizada uma estrutura de nomes hierárquica, onde cada pedaço de conteúdo, denominado *chunk*, deve ser nomeado utilizando identificadores legíveis. Os nomes precisam ser únicos dentro de um segmento local, porém, não precisam ser únicos globalmente, pois a estrutura hierárquica evita a ambiguidade na identificação ao analisar o caminho completo (ZHANG ET AL., 2010).

Os roteadores CCN não interpretam o nome do conteúdo, permitindo que a aplicação decida a estrutura que melhor se ajuste às suas necessidades. É possível, inclusive, a aplicação fornecer informações sobre o conteúdo diretamente em seu nome de identificação, como por exemplo, formato, versão, etc. Um exemplo de nomeação pode ser visto na Figura 2.2.

/ufscar/dc/ppgcc/video.avi/1/2

Figura 2.2 - Exemplo de nome de conteúdo

Considerando que um roteador CCN não interpreta o significado do nome do conteúdo para realizar o encaminhamento, uma aplicação poderia utilizar o exemplo de nome ilustrado pela Figura 2.2, de modo a representar hierarquicamente a segunda parte, da versão 1, do conteúdo “video.avi”, disponibilizado pelo publicador “ppgcc”, que faz parte do departamento “dc”, da instituição “ufscar”.

O símbolo “/” é utilizado para separar os componentes do nome, mas não faz parte do nome e, portanto, é interpretado pelo roteador CCN, diferentemente dos nomes em si.

2.2.2 Funcionamento básico da arquitetura CCN

A arquitetura CCN utiliza dois tipos de pacotes, chamados *Interest* e *Data* (também chamado de *Content Object*). O pacote do tipo *Interest* é enviado por um usuário quando está interessado em algum conteúdo e irá conter o nome do conteúdo desejado. O pacote do tipo *Data* contém o conteúdo em si e é enviado ao usuário consumidor como resposta ao pacote do tipo *Interest* (ZHANG ET AL., 2010).

Deste modo, há uma relação de um-para-um na transmissão dos pacotes, pois um pacote *Data* só é enviado ao usuário caso um pacote *Interest* tenha sido enviado anteriormente.

Os pacotes do tipo *Data* são tratados independentemente do local de publicação original ou do destinatário final, podendo ser armazenado em *cache* pelos roteadores que o receberam para atender futuras requisições. Na Figura 2.3, são apresentados os pacotes *Interest* e *Data*.

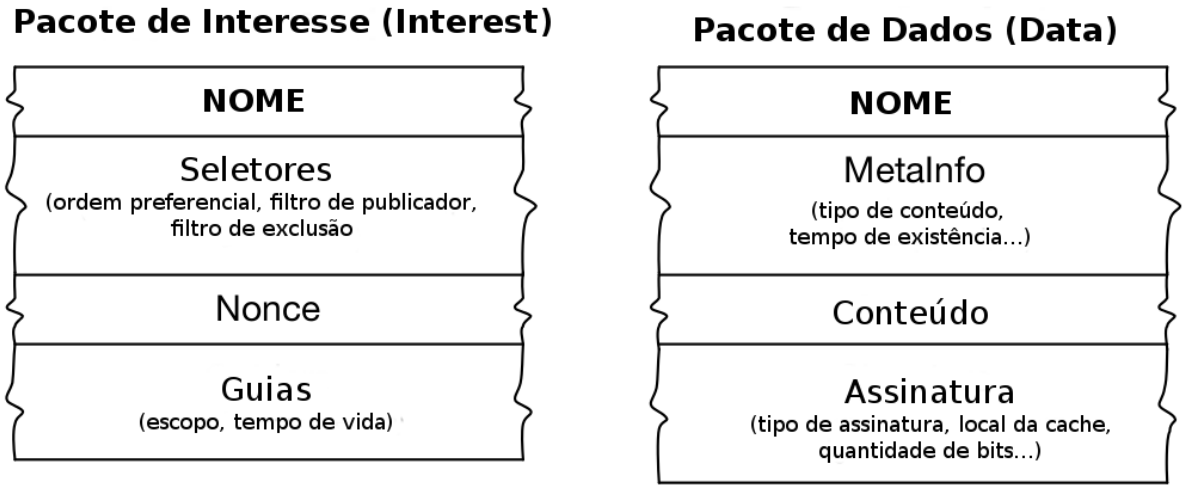

Figura 2.3 - Pacotes Interest e Data
Fonte: Adaptado de Zhang et al. (2014)

Como é possível visualizar na Figura 2.3, além do nome do conteúdo e do conteúdo em si, alguns parâmetros extras são utilizados para suportar outras funcionalidades.

No pacote *Interest* podem ser definidos seletores para ordem de preferência, filtro de publicadores, regras de exclusão, entre outros. Há ainda um campo *Nonce*,

que permite descartar duplicatas de um mesmo pacote, e um campo *Guiders*, que permite definir o escopo e o tempo de vida da requisição.

Já o pacote *Data* também possui campos para parâmetros relacionados à assinatura, que permitem autenticar o conteúdo do pacote, além de meta-informações, que permitem descrever dados como o tipo do conteúdo, data da última alteração, entre outros.

2.2.3 Segurança dos conteúdos

Enquanto a segurança nas redes IP tem como foco o canal por onde os dados trafegam, a arquitetura CCN opera com a premissa de que a segurança deve ser estabelecida no próprio conteúdo, independente do canal utilizado (ZHANG ET AL., 2014).

Para este fim, existem os campos relacionados à assinatura (apresentados na Figura *2.3*), permitindo que em qualquer ponto da rede seja verificada a autenticidade dos dados.

Adicionalmente, conteúdos sensíveis podem ser criptografados pela própria aplicação, que deve ficar responsável por realizar a cifragem dos dados e o gerenciamento das chaves utilizadas. Deste modo, a aplicação pode escolher o mecanismo que for mais conveniente para as suas necessidades (MOISEENKO; ZHANG, 2014).

Estes aspectos de segurança inerentes à arquitetura CCN permitem que alguns problemas costumeiros na Internet sejam minimizados. Como exemplo, ataques *man in the middle* que alterem os dados de um pacote e a prática de *spoofing* se tornam mais complexos devido à autenticação pública poder ser realizada em qualquer nó da rede. Outro exemplo, é que se torna mais fácil a proteção em relação a ataques de negação de serviço, uma vez que pacotes do tipo *Interest* para o mesmo conteúdo não precisam ser encaminhados diversas vezes, uma vez que todas as requisições são mantidas na PIT e serão atendidas quando o conteúdo estiver disponível no roteador.

2.2.4 Roteamento

Para um conteúdo disponibilizado em uma rede CCN é possível que existam diversas rotas diferentes. Isso ocorre pois como não há uma relação obrigatória entre conteúdo e endereço do publicador, o conteúdo pode ser mantido em *cache* em qualquer lugar, portanto, pode haver interfaces de saída distintas para um mesmo conteúdo, permitindo, por exemplo, que um determinado arquivo tenha *chunks* armazenados em localidades distintas e que mesmo assim possam ser enviados ao usuário requisitante para formar o conteúdo completo.

Essa característica é especialmente interessante no quesito mobilidade, pois como não há uma dependência de localidade física, não é necessário um mapeamento entre endereços de rede e de enlace, permitindo que um usuário migre entre diferentes roteadores e reestabeleça a conexão rapidamente (ZHANG ET AL., 2014). Além disso, permite que o algoritmo de roteamento utilizado opte pela melhor rota no momento do encaminhamento, utilizando políticas que permitam, por exemplo, não sobrecarregar outro roteador que esteja lento, visando melhorar o desempenho da rede.

Devido à similaridade entre o roteamento IP e a estrutura hierárquica de nomes da CCN na questão de agregação de prefixos, qualquer algoritmo de roteamento IP pode ser adaptado para a CCN, como por exemplo, o OSPF (*Open Shortest Path First)* que tem uma variação OSPFN (*Open Shortest Path First for Named Data)* (WANG ET AL., 2012).

O conceito da CCN deu origem ao projeto Named-Data Networking (NDN), uma implementação da arquitetura CCN, que provê *testbed*, aplicações e simuladores (ZHANG ET AL., 2014). A NDN é abordada em mais detalhes nas próximas seções.

2.2.5 Named Data Networking

A *Named-Data Networking* (NDN) é um projeto que aplica os conceitos da arquitetura CCN proposta por Jacobson et al. (2009). A rede NDN utiliza um esquema hierárquico de nomes de conteúdo, onde cada *chunk* recebe um nome único de identificação.

Esta estrutura de nomes tem o objetivo de substituir o esquema de roteamento baseado em localização do TCP/IP, permitindo que a arquitetura seja executada como

um *overlay* sobre a camada de enlace da Internet. A arquitetura da NDN é apresentada na Figura 2.4.

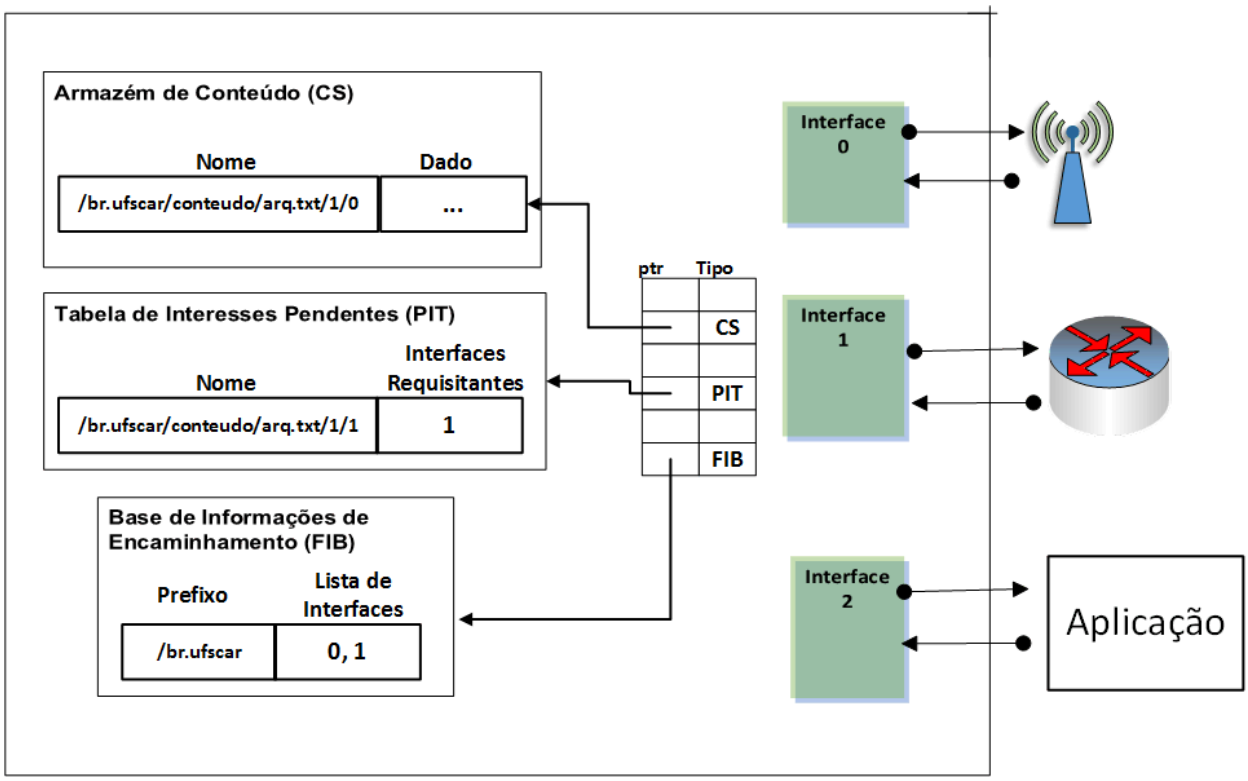

Figura 2.4 - Componentes Básicos da Arquitetura NDN

Fonte: Adaptado de Zhang et al. (2010)

Como apresentado na Figura 2.4, a arquitetura NDN possui três componentes principais, o Armazém de Conteúdos (*Content Store* - CS), a Tabela de Interesses Pendentes (*Pending Interest Table* - PIT) e a Base de Informações de Encaminhamento (*Foward Information Base* - FIB).

FIB é uma tabela de encaminhamento similar às existentes nos roteadores IP. Ela contém o mapeamento entre os prefixos de nomes de conteúdo, que identificam cada conteúdo individualmente, e uma ou mais interfaces para encaminhamento, permitindo a utilização de múltiplas fontes de forma nativa.

CS implementa o recurso de *cache* de arquivos nos roteadores CCN, permitindo que os conteúdos frequentemente utilizados sejam armazenados o mais próximo possível do usuário.

O objetivo do armazém é otimizar o desempenho da rede e economizar largura de banda, uma vez que um nó próximo ao usuário poderá fornecer o conteúdo, sem

precisar consultar outros nós distantes. Deste modo, diferentemente dos roteadores IP que possuem espaço relativamente pequeno de *buffer* e não armazenam os pacotes após a transmissão, um roteador NDN é capaz de armazenar os pacotes para reutilizar para atender novas requisições ao mesmo conteúdo.

Para determinar quais conteúdos e por quanto tempo ficarão armazenados, podem ser utilizadas diversas políticas de armazenamento, as quais são definidas por algoritmos similares aos existentes nas redes IP. É possível, inclusive, cada roteador utilizar políticas diferentes um do outro para decidir o que ficará em *cache* (ZHANG ET AL., 2014).

Quando um usuário envia um pacote *Interest* para um conteúdo que já está disponível na CS, a requisição é respondida imediatamente pelo roteador, que enviará o pacote *Data* com o conteúdo ao usuário solicitante.

Porém, quando o roteador recebe um pacote *Interest* para um determinado conteúdo que não está disponível em seu armazém, ele irá registrar a *interface* de origem da requisição na PIT. Este registro do interesse na PIT é necessário para que seja possível identificar por onde o conteúdo solicitado deve ser enviado quando estiver disponível no roteador.

Caso ocorra uma nova requisição de interesse para um mesmo conteúdo já solicitado anteriormente e ainda não disponível na CS, a nova interface de origem também é armazenada na PIT. Os registros são mantidos na PIT até que o pacote *Data* seja recebido ou o tempo de aguardo expire, para que uma solicitação não fique pendente eternamente.

Após o registro na PIT, o *Interest* é encaminhado à FIB, onde o algoritmo de roteamento utilizado irá verificar, de acordo com o prefixo, por qual interface de saída a requisição deverá ser encaminhada. Caso não exista uma entrada correspondente, a requisição será descartada.

2.2.6 ndnSim

O ndnSIM (AFANASYEV ET AL., 2012) é um módulo para o simulador ns-3, desenvolvido com o objetivo de prover uma infraestrutura para simulação de aplicações na arquitetura NDN de Jacobson et al. (2009).

O simulador foi desenvolvido em camadas distintas e apresenta um modelo de camada de rede alternativo, que pode ser executado sobre as camadas de enlace

tradicionais da arquitetura TCP/IP. O modo como os componentes são distribuídos em camadas e suas relações com camadas do modelo TCP/IP estão apresentados na Figura 2.5.

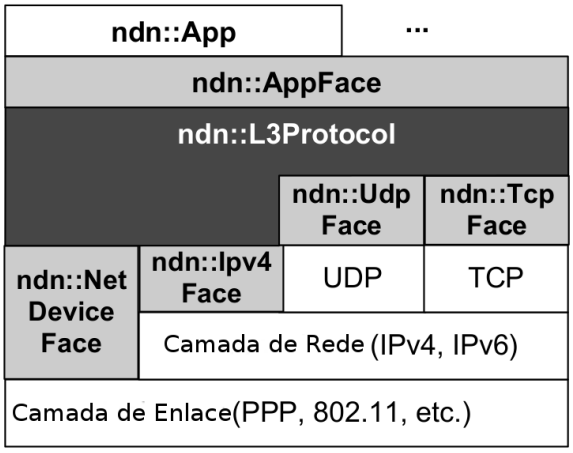

Figura 2.5 - Relação entre os componentes do ndnSIM

Fonte: Adaptado de Afanasyev et al., 2012.

Os diversos componentes do ndnSIM permitem a simulação de redes de uma maneira prática, sem que precisem ser desenvolvidas as funcionalidades essenciais da arquitetura NDN. Deste modo, o simulador já provê recursos como encaminhamento, requisição e armazenamento dos pacotes. Os principais componentes existentes no simulador estão descritos a seguir:

- ndn::L3Protocol: implementação das principais interações do protocolo NDN, incluindo recebimento dos pacotes de Interesse e de dados das camadas adjacentes;
- ndn::Face: abstração para habilitar comunicações entre as aplicações e os nós da rede, com suporte opcional para serviços da camada de enlace, como por exemplo, controle de congestionamento;
- ndn::ContentStore: abstração para armazenamento de pacotes de dados;
- ndn::Pit: abstração para a Pending Interest Table (PIT);
- ndn::Fib: abstração para a Forwarding Information Base (FIB);
- ndn::ForwardingStrategy: abstração e implementação principal do encaminhamento de pacotes de Interesse e de dados.

O processo de comunicação entre os principais componentes do ndnSIM é apresentado na Figura 2.6, e permite ver como os mesmos se relacionam.

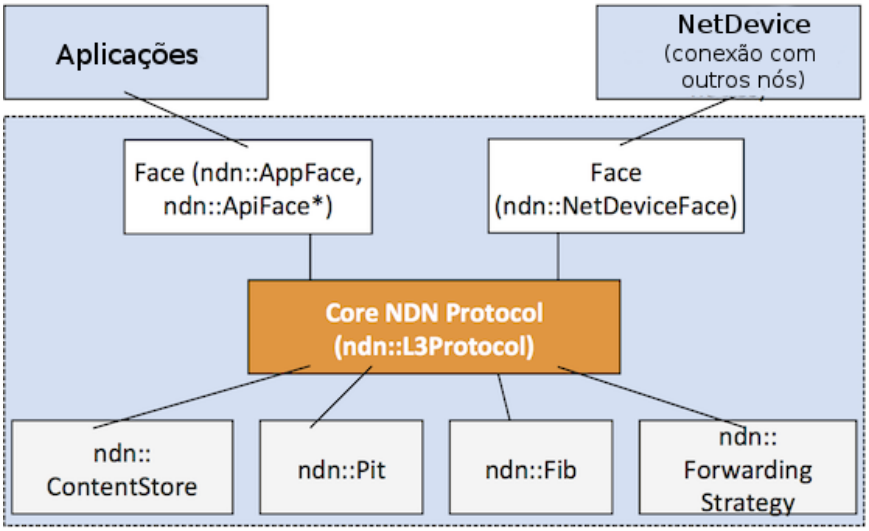

Figura 2.6 - Comunicação entre os componentes do ndnSIM

Fonte: Afanasyev et al., 2012.

É possível verificar na Figura 2.6 que a camada de rede (ndn::L3Protocol) representa o componente principal do simulador.

As aplicações e os demais nós da rede, que constituem os elementos externos ao simulador, se relacionam com esta camada utilizando a interface chamada *Face*. A camada *Face*, portanto, é o mecanismo que intermedeia a requisição dos serviços da CS, PIT e FIB e das estratégias de encaminhamento, para os elementos externos.

2.3 Privacidade

O conceito de privacidade é discutido há muito tempo, tendo como importante marco a publicação do “Direito a Privacidade” (WARREN; BRANDEIS, 1890), considerado o precursor da área de leis de privacidade.

Em sua publicação, Warren e Brandeis (1890) definem o termo privacidade como o direito de ser deixado sozinho. Além disso, os autores reforçam que o direito à propriedade abrange não somente os bens tangíveis, mas também os intangíveis.

Atualmente, estes conceitos envolvem os documentos e dados disponibilizados eletronicamente e compartilhados pelos usuários em redes de computadores.

Hughes (1997) conceitua a privacidade como o poder de uma pessoa revelar informações pessoais aos outros de maneira seletiva, ou seja, com a capacidade de determinar quem poderá ter conhecimento das informações disponibilizadas.

Baseada nestes conceitos, a privacidade no contexto dos sistemas computacionais pode ser definida como o controle do usuário sobre suas informações pessoais, tendo a capacidade de determinar quais informações de si poderão ser utilizadas e quem irá acessá-las.

Chen e Zhao (2012) afirmam que o desejo por privacidade pode variar de acordo com o país, cultura ou jurisdição do usuário, e que a privacidade se relaciona com as fases de coleta, uso, divulgação, armazenamento e destruição de dados.

Deste modo, ao garantir a privacidade de um usuário, uma aplicação deve disponibilizar meios para que o próprio usuário defina com quem deseja compartilhar os seus dados. Além disso, como os dados geralmente são armazenados em um ambiente fora do controle do usuário, a aplicação também deve garantir os mecanismos técnicos que assegurem que os dados não serão interceptados por outros usuários em nenhuma das fases citadas.

Qualquer aplicação que registre dados pessoais deveria ter a preocupação com a privacidade, desde sistemas comerciais simples, onde há apenas um cadastro de clientes, até sistemas de recomendação que utilizam complexos algoritmos de classificação para organizar, analisar e identificar padrões a partir dos dados de um usuário, para lhe oferecer serviços personalizados.

No contexto das redes CCN, Arianfar et al. (2011) afirmam que por atuarem no nível de conteúdo, ao invés de bits como a Internet atual, ocorre um grande problema em relação à privacidade. Isso se deve ao fato das CCN armazenarem as requisições dos usuários e, em caso de monitoramento, permitem a identificação do solicitante e do conteúdo acessado.

Caso um invasor obtenha privilégios nos roteadores da rede, teria acesso às requisições dos usuários, invadindo as suas privacidades.

Além disso, outro problema em relação à privacidade do usuário ocorre no caso da infraestrutura da rede estar sob controle de uma única entidade ou governo. Deste modo, além de estarem aptos a monitorar os usuários, poderiam ainda realizar

bloqueios e filtros de conteúdo, em um processo de censura que impediria o acesso a determinados assuntos.

Os principais problemas em relação à privacidade em CCN estão relacionados ao *cache* de dados existente nos armazéns de conteúdo. Apesar de serem úteis para melhorar o desempenho no acesso aos conteúdos, também são suscetíveis a ataques, que poderiam tanto violar o sigilo dos eventos, ou seja, identificar o que o usuário está acessando, quanto acessar o conteúdo propriamente dito (LAUINGER ET AL., 2012; CHABANE ET AL., 2013; MOHAISEN ET AL., 2013; MOHAISEN ET AL., 2015, TOURANI ET AL., 2016).

A principal solução aplicada aos problemas de segurança descritos acima é o uso de criptografia, que será abordado na próxima seção.

2.4 Criptografia

Segundo Stallings (2008), a criptografia pode ser entendida como um processo de converter um texto claro (mensagem original legível, inteligível à máquina ou ao homem) em texto cifrado (mensagem embaralhada, ilegível, tanto para a máquina quanto para o homem).

O uso da criptografia tem sido o principal recurso de segurança utilizado para garantir alguns dos princípios básicos da segurança da informação:

- confidencialidade – a informação só deve ser acessada por quem tem direito de acesso;

- autenticidade – garante que o emissor da informação seja realmente quem diz ser;

- integridade – protege a informação contra alterações indevidas;

- privacidade – permitir ao usuário decidir quem irá ter acesso aos seus dados.

De maneira geral, os métodos de criptografia têm como objetivo proteger comunicações diversas (mensagens, transferência de dados e arquivos, transações financeiras *online,* acesso a sistemas, etc.), do acesso por pessoas indesejadas.

Como a transmissão dos dados geralmente é realizada sobre canais não seguros, é impossível para a aplicação garantir que os dados não serão interceptados.

Deste modo, mesmo que uma mensagem seja interceptada em algum momento, graças à criptografia, o conteúdo estará ilegível.

Existem diversas técnicas para realizar a criptografia, entretanto, é possível dividir os algoritmos em dois grupos: simétrico e assimétrico.

A criptografia de chave simétrica implica na utilização de uma única chave, que irá tanto cifrar, quanto decifrar o conteúdo. Neste caso, se a chave for roubada por um intruso, este poderá criar novos arquivos cifrados para enviar como se fossem legítimos, ou acessar o conteúdo de arquivos cifrados por outros usuários. Portanto, o envio da chave entre os usuários torna-se um processo crítico, devendo ter atenção especial (KUROSE; ROSS, 2010).

O conceito de criptografia de chave assimétrica consiste basicamente em utilizar uma chave para cifrar a mensagem e outra chave diferente para decifrar a mensagem. Neste caso, são utilizadas duas chaves relacionadas, uma privada e outra pública, sendo que uma mensagem cifrada com uma chave privada, só pode ser aberta pela chave pública e uma mensagem cifrada pela chave pública, só pode ser aberta pela chave privada (KUROSE; ROSS, 2010).

Um exemplo de como as chaves pública e privada se relacionam pode ser extraído do algoritmo RSA, que funciona da seguinte maneira (STALLINGS, 2008):

1. São selecionados dois números primos aleatórios, *p* e *q*.
2. É calculado o valor de *n*, sendo
$$n = pq$$
3. A função totiente de Euler, é calculada para *n*, de modo que:
$$n: \emptyset(n) = (p-1)(q-1)$$
4. Um inteiro *e* é escolhido, tal que $1 < e < \emptyset(n)$, de forma que e e $\emptyset(n)$ sejam primos entre si.
5. Utilizando o algoritmo de Euclides estendido, o valor de *d* é computado de modo que $de \equiv 1\ (mod \emptyset(n))$ ou seja, *d* deve ser o inverso multiplicativo de *e* em $\left(mod \emptyset(n)\right)$.
6. Como resultado, serão obtidas:

   Chave pública: (n,e)

   Chave privada: (p,q,d)

2.4.1 Criptografia baseada em atributos

A Criptografia Baseada em Atributos (*Attribute Based Encryption* – ABE) é um esquema de criptografia assimétrica, originalmente proposto por Sahai e Waters (2005). A ABE permite que um usuário seja identificado por um conjunto de atributos descritivos, os quais irão determinar se o usuário possui permissão de acesso a determinado conteúdo.

A ideia da ABE surgiu a partir da criptografia baseada em identidade (*Identity-Based Encryption* – IBE) (SHAMIR, 1985). Entretanto, diferentemente da IBE, onde todos os atributos são utilizados obrigatoriamente para realizar a criptografia dos dados, na ABE um ou mais atributos podem ser escolhidos como critérios para realizar a decifragem, permitindo uma maior flexibilidade em sua aplicação.

O algoritmo utilizado por Sahai e Waters (2005) define um nível mínimo de semelhança entre os atributos escolhidos para cifrar e os atributos que permitirão decifrar o conteúdo. Desta forma, a chave utilizada para decifrar não precisa ter exatamente os mesmos atributos que foram utilizados para cifrar, bastando apenas que a diferença não seja maior do que o nível mínimo definido.

Esta característica permite que usuários com atributos semelhantes, mas não idênticos aos escolhidos na cifragem, tenham acesso ao conteúdo, tornando complexa a existência de políticas mais rígidas.

Com o intuito de aumentar as funcionalidades e corrigir o problema do nível de semelhança da ABE, Goyal et al. (2006) aperfeiçoaram o conceito, permitindo a utilização de políticas mais precisas. Para isso, incluíram a existência de uma política de acesso baseada em uma estrutura de função que está armazenada na própria chave privada. Esta técnica recebeu o nome de *Key-Policy Attibute-Based Encryption* (KP-ABE).

Na criptografia KP-ABE o conteúdo é cifrado com base em um ou mais atributos e as chaves privadas contêm as políticas de acesso. Essas políticas de acesso são armazenadas em uma estrutura em formato de árvore. O usuário que deseja acessar o conteúdo deverá possuir os mesmos atributos que foram especificados na construção da política.

Um problema encontrado no esquema KP-ABE, consiste no fato da entidade responsável por gerar as chaves ter total controle sobre o acesso aos dados, uma vez que a política de acesso fica armazenada na chave gerada por ela.

Com base neste problema, Goyal et al. (2006) propuseram ainda um esquema inverso ao KP-ABE, em que a política de acesso é definida no próprio conteúdo cifrado, ao invés da chave. Esta criptografia foi chamada de *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE).

Na CP-ABE as chaves privadas estão associadas a um conjunto de atributos, que pode conter um único ou mais atributos e o texto cifrado armazena também uma política de acesso. Deste modo, o controle do acesso não fica centralizado na entidade geradora da chave, pois o próprio conteúdo irá decidir quem pode acessá-lo.

2.4.2 Funcionamento do CP-ABE

Bethencourt et al. (2007) desenvolveram um esquema CP-ABE baseado na proposta de Goyal et al. (2006), onde o processo foi separado em quatro algoritmos básicos: *Setup*, *Encrypt*, *Key Generation* e *Decrypt*, além de um quinto algoritmo opcional, chamado *Delegate*.

O algoritmo desenvolvido por Bethencourt et al. (2007), baseia-se no conceito de emparelhamento bilinear, no qual considerando os grupos cíclicos $G_0, G_1 e\ G_2$ de mesma ordem primária *p,* um emparelhamento *e* é uma função $(G_0 \mathrm{X}\ G_1 \rightarrow G_2)$, se $\forall u \in G_0, v\ \in G_1$ e $a, b\ \in Z_p, e(u^a, v^b) = e(u, v)^{ab}$ e $(\mathrm{g0}, \mathrm{g1}) \neq 1$, então *e* é um emparelhamento bilinear. Caso $G_0 = G_1$ trata-se de em emparelhamento simétrico, e caso $G_0 \neq G_1$, trata-se de um emparelhamento assimétrico.

O algoritmo *Setup* é executado pelo sistema gerador de chaves, e tem como objetivo fornecer a chave mestre de criptografia (MK), que será utilizada na geração das demais chaves, bem como uma chave pública (PK). No algoritmo desenvolvido por Bethencourt et al. (2007), as chaves são geradas considerando um grupo de ordem primária G_0 e dois expoentes aleatórios $\alpha, \beta\ \in Z_p, G_0 = < g >$ de tal modo que as saídas serão:

$$PK = G_0, g, h = g^{\beta}, f = g^{1/\beta} e(g, g)^{\alpha}$$

$$MK = (\beta, g^{\alpha})$$

O algoritmo *Encrypt* é responsável por realizar a criptografia dos dados, recebendo como entrada a, a chave pública (PK), a mensagem com o conteúdo original (M) e a política de acesso. A política de acesso é composta por atributos e

parâmetros que irão formar a estrutura de acesso (T) armazenada em formato de árvore.

A política de acesso no algoritmo proposto por Bethencourt et al. (2007) é construída em uma estrutura em formato de árvore, onde cada nó que não seja um nó folha constitui uma condição “AND” ou “OR” (este nó é chamado de *threshold gate* ou portão limiar) para um ou mais valores de atributo que serão descritos nos nós folhas a seguir (*threshold* ou limiar).

Considerando num_x como o número de filhos de um nó x e k_x o seu valor de limiar, então $0 < k_x \leq num_x$. Quando $k_x = 1$, o *threshold gate* funciona como um “OR” (pois se apenas um dos atributos descritos nos nós folhas coincidir, será aceito), e quando $k_x = num_x$, funciona como um “AND” (pois neste caso, todos os atributos dos nós folha devem coincidir). Todo nó folha deve ser descrito por um atributo, fornecido pelo usuário publicador que cifrou os dados, e um limiar de valor $k_x = 1$.

Exemplificando, considere a seguinte situação: um usuário criptografa um arquivo que só poderá ser acessado por usuários que possuam os seguintes atributos: “amigo” ou “mestrado” e “DC”. A estrutura de acesso teria como regra: (“amigo” OR (mestrado AND “DC”). A Figura 2.7 apresenta um exemplo de árvore de acesso baseada neste exemplo.

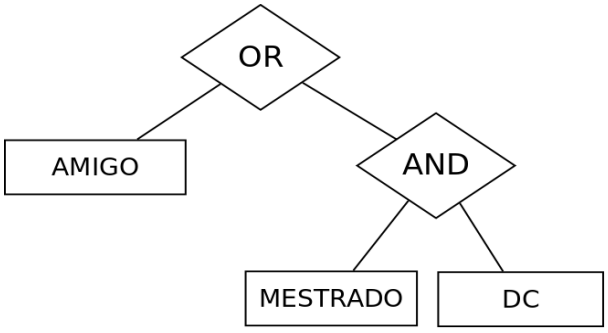

Figura 2.7 - Exemplo de estrutura de acesso

O algoritmo *Encrypt* de Bethencourt et al. (2007) realiza a criação da estrutura de acesso em sentido *top-down* à partir do nó raiz (R). É definido então um polinômio q_x para cada nó x (seja este um nó folha ou não). Para cada nó x contido na árvore, é definido um grau d_x do polinômio q_x, considerando K_x como sendo o valor limiar para *x*, então $d_x = k_x - 1$. À partir de *R*, é definido um valor aleatório para $s \in Z_p$ e define $q_R(0) = s$. Sendo, *Y* um conjunto de nós folhas de *T*, o algoritmo *Encrypt* retorna

a mensagem cifrada (MC), que inclui o conteúdo e também a política de acesso, à partir de:

$$MC = (T, = Me(g,g)^{\alpha s}, C = h^{s},$$
$$\forall_{y} \in Y\colon\ C_{y} = g^{qy(0)}, C'_{y} = H\big(att(y)\big)^{qy(0)})$$

O algoritmo *Key Generation* tem como objetivo gerar as chaves privadas de cada usuário. Para realizar esta tarefa, o algoritmo de Bethencourt et al. (2007) recebe como entrada um conjunto de atributos (S) que o usuário possui.

Inicialmente o algoritmo estabelece um valor aleatório para $r \in Z_p$, para em seguida definir $r_j \in Z_p$ para cada atributo $j \in S$. A saída do algoritmo retorna uma chave privada (SK) onde estão armazenados os atributos que o usuário possui, de modo que não seja capaz forjar a posse de um atributo inexistente.

$$SK = (D = g^{\frac{\alpha+r}{\beta}},$$
$$\forall j \in S : D_j = g^{r}\, H(j)^{rj}, D'_j = g^{rj})$$

Decrypt é o algoritmo que irá realizar a decifragem dos dados. Caso o conjunto de atributos que o usuário possui (os quais estão definidos na chave privada do usuário) atenda a política de acesso, o conteúdo será descriptografado.

Bethencourt et al. (2007) definiram um algoritmo recursivo chamado *DecryptNode* que irá analisar a estrutura de acesso com o intuito de verificar se a chave privada do usuário atende à política de acesso definida durante a criptografia. Para isso, o algoritmo *DecrytpNode* recebe como entrada a mensagem cifrada (MC), a chave privada do usuário (SK) e um nó *x* da estrutura de acesso *T*. Caso o nó *x* seja um nó folha de *T*, então: $i = att(x)$ para $i \in S$, então:

$$DecryptNode(MC, SK, x) = \frac{e(D_i, C_x)}{e(D'_i, C'_x)}$$
$$= \frac{e\big(g^{r}.H(i)^{ri}, h^{qx(0)}\big)}{e(g^{ri}, H(i)^{qx(0)})}$$
$$= e(g,g)^{rq_x(0)}\ .$$

Caso $i \notin S$, então:

$$\mathrm{DecryptNode}(\mathrm{MC}, \mathrm{SK}, \mathrm{x}) = \perp.$$

Para cada nó x que não seja folha, a interpolação de Lagrange será aplicada para k_x, de modo que $e(g,g)^{rq_{z_j}(0)}$ à partir dos filhos $\{z_j\}$ para calcular $e(g,g)^{rq_x(0)}$.

Concluída a execução do algoritmo *DecryptNode*, o algoritmo *Decrypt* inicia a execução no nó raiz *r* da árvore *T*. Então, é definido que $A = \quad A = \mathrm{DecryptNode(MC, SK, r)} = \ e(g,g)^{rq_R(0)} = e(g,g)^{rs}$.

A descriptografia é realizada por:

$\tilde{C}/(e(C,D)/A) = \ \tilde{C}/(e(h^s, g^{\frac{a+r}{\beta}}),/e(g,g)^{rs}) \ = M$,sendo M a mensagem original antes da criptografia.

Por fim, há ainda um quinto algoritmo opcional, chamado *Delegate*, que gera novas chaves privadas a partir de um conjunto novo de atributos especificado. Como entrada o algoritmo de Bethencourt et al. (2008) recebe a chave privada do usuário (SK), que é composta por um conjunto *S* de atributos e um conjunto de atributos $\tilde{S}$, de modo que $\tilde{S} \subseteq S$. Considerando que SK possui o formato $\mathrm{SK} = \left(\mathrm{D}, \forall \mathrm{j} \in \mathrm{S} : \mathrm{D_j}, \mathrm{D'_j}\right)$, o algoritmo define um valor aleatório para $\tilde{r}$ e $\tilde{r}_k \forall k \in \widetilde{S}$. O algoritmo cria então as chaves como:

$$\widetilde{SK} = (\widetilde{D} = \ Df^{\tilde{r}},$$
$$\forall k \ \in \ \tilde{S}: \widetilde{D}_k = D_k g^{\tilde{r}_k} H(k)^{\tilde{r}_k}, \widetilde{D}'_k = D'_k = D'_k g^{\tilde{r}_k})$$

As chaves privadas criadas pelo algoritmo são geradas randomicamente de modo que não podem ser combinadas, tornando o algoritmo resistente a ataques de conluio.

2.4.3 Revogação de privilégios

Apesar de representar um mecanismo flexível ao atribuir restrições de acesso diretamente no conteúdo, a proposta de Bethencourt et al. (2007) possui um problema em relação à revogação dos direitos de acesso.

Para retirar o acesso de um usuário que perdeu o privilégio é necessário desativar todas as chaves antigas, gerar novas chaves, realizar novamente a criptografia do conteúdo e por fim, republicá-lo para que os usuários o acessem novamente.

Apesar de ser uma situação aceitável em alguns tipos de aplicações, no contexto deste trabalho, o recurso de *cache* da CCN iria manter os conteúdos cifrados antigos por tempo indeterminado.

Para resolver este problema da revogação de direitos, Jahid e Borisov (2012) propuseram um esquema no qual um *proxy* é inserido. Este *proxy* tem como responsabilidade intermediar o acesso e impedir que usuários com privilégios revogados continuem acessando um conteúdo ao qual tinham direito de acesso anteriormente.

O *proxy* é parcialmente confiável, ou seja, não é capaz de decifrar o conteúdo nem de adicionar privilégios a outros usuários sozinho. Além disso, utilizando o *proxy*, não é necessário que os usuários troquem suas chaves privadas, nem que o conteúdo tenha que ser criptografado novamente.

O esquema aplicado pelo *proxy* incluído por Jahid e Borisov (2012) é baseado no trabalho de Naor e Pinkas (2001), o qual é dividido em duas fases, Inicialização e Revogação.

Na fase de inicialização, executada uma única vez, é gerado um polinômio aleatório *P* de grau *t* sobre Z_p, de modo que cada usuário *u* com identidade I_u, receba uma chave pessoal $\langle I\ u\ , P\ (I\ u\ )\rangle$.

Na fase de revogação, é necessário que o proxy obtenha a lista de usuários a serem revogados (RL), contendo as suas identidades, em seguida, utilizando um valor aleatório *a*, é definida uma nova chave como $g^{rP(0)}$, que permanecerá desconhecida para o usuários que foram revogados. A chave então é enviada aos usuários não revogados quando requisitada ao proxy, para que combinada com sua chave privada possam proceder com a descriptografia.

Para que fosse possível realizar essa nova funcionalidade, foram necessárias algumas alterações na proposta de Bethencourt et al. (2007). O algoritmo chamado *easier*, desenvolvido por Jahid e Borisov (2012) utiliza emparelhamento bilinear assimétrico.

Para cada usuário além de ser gerada uma chave privada, também é gerado um arquivo que contém informações de sua identidade. Essa identidade será utilizada no processo de validação do acesso. Para realizar essa alteração, Jahid e Borisov (2012) destacam a inclusão de um componente extra no algoritmo KeyGen, chamado D_j'', que permite a criação do *id* do usuário. Para $H: \{0,1\} * \rightarrow G_1\ e\ h_j = \log_{g1} H(j)$:

$$Sk = (D, \forall j \in S : (D_j, D'_j, D''_j)), \text{ onde}$$

$$D = g_1^{a+\frac{r}{\beta}}, D_j = g_1^r . H(j)^{rj^{P(0)}} = g_1^{r+h_j r_j P(0)},$$

$$D'_j = G_0^{rj}, D''_j = \left(D'_j\right)^{P(uk)} = g_o^{rj\hat{}P(uk)}$$

O algoritmo *Encrypt*, ao criptografar um arquivo, gera uma estrutura de acesso que não permite que o usuário decifre o arquivo diretamente com sua chave. Essa estrutura deverá ser complementada pelo *proxy*. Basicamente ele foi mantido por Jahid e Borisov (2012) quase idêntico ao desenvolvido por Bethencourt (2007), porém, foi adapatado para utilizar emparelhamento bilinear assimétrico, de modo que:

$$MT = (\tau, \tilde{C} = Me(g_0, g_1)^{\alpha s}, C = h^s = g_0^{\beta s},$$

$$\forall y \in Y : C_y = g_0^{y(0)}, C'_y = H\left(att(y)\right)^{qy(0)} = g_1^{h_y q_y(0)})$$

Cada vez que um usuário é revogado, é necessário gerar uma nova chave para o *proxy (PXK)*. Este processo é feito pelo algoritmo *ProxyRekey,* incluído por Jahid e Borisov (2012). Este algoritmo recebe como entrada a chave pública (PK), chave mestre (MK) e uma lista de usuários revogados (RL) que possui as identidades dos usuários (u_i) de modo que $i \in \{1, \dots, t\}$.

$$PXK = \forall u_i \in RL : (u_i, P(u_i))$$

Para completar a decifragem, antes do algoritmo *Decrypt* realizar sua tarefa, o *proxy* deve realizar a conversão da estrutura de acesso. O algoritmo *Convert* foi inserido por Jahid e Borisov (2012) para permitir ao *proxy* realizar estava atividade, e recebe como entrada $\mathrm{PXK}, \forall \mathrm{y} \in \mathrm{Y}: \mathrm{C_y}, \mathrm{u_k}$, para em seguida realizar:

$$\lambda_i = \frac{u_k}{u_k - u_i} \prod_{j \neq i} \frac{u_j}{(u_j - u_i)}, \forall i, j \in \{1, \dots, t\}, k \notin \{1, \dots, t\}$$

$$\forall y \in Y: C''_y = \left(C'_y\right)^{\sum_{i=1}^{t} \lambda_i P(u_i)} = g_1^{h_y q_y(0) \sum_{i=1}^{t} \lambda_i P(u_i)}$$

Comparado ao algoritmo original de Bethencourt et al. (2007), o algoritmo *Decrypt* desenvolvido por Jahid e Borisov (2012) possui um passo a mais em cada nó folha contido na estrura de acesso. Nessa versão do algoritmo *DecryptNode*, o que se diferencia em relação a versão original, é que considerando *S* um conjunto de atributos

para a chave privada do usuário (SK), em cada nó folha *x*, com $i = att(x)$, se $i \in S$, então:

$$DecryptNode(MC, SK, x)$$

$$= \frac{e(C_x, D_i)}{e(D_i'', C_x')^{\lambda_k} e(D_i', C_x'')}$$

$$= \frac{e(g_0, g_1)^{rqx(0)+h_i r_i P(0) q_x(0)}}{e(g_0, g_1)^{r_i h_i q_x(0) \lambda_k P(u_k)} e(g_0, g_1)^{r_i h_i q_x(0) \sum_{j=1}^{t} \lambda_j P(u_j)}}$$

$$= \frac{e(g_0\ , g_1)^{rq_x(0)+h_i r_i P(0) q_x(0)}}{e(g_0, g_1)^{r_i h_i q_x(0)} \left(\sum_{j=1}^{t} \lambda_j P(u_j) + \lambda_k P(u_k)\right)}$$

$$= e(g_0, g_1)^{rq_x(0)}$$

Para realizar a revogação é utilizado um polinômio para cada um dos atributos, os quais são ilegíveis dentro da chave privada do usuário (SK), sendo que a chave do proxy (PXK) contém a lista de usuários revogados. Sendo Y' a lista de atributos, Y o conjunto de atributos a ser revogados, e $Y \subseteq Y'$, e t_y o número de usuários que terão o atributo revogado:

$$MK = \beta, g_1^{\alpha}, \forall y \in Y' P_y(0)$$

$$SK\ = (D, D_j = g_1^r \cdot H(j)^{r_j P_j(0)}, D_j''\ = \left(D_j'\right)^{P_j(u_k)})$$

$$PXK = \forall y \in\ Y, \forall i \in\ t_y : \{\langle u_i, P_y\ (u_i)\rangle\}$$

2.5 Considerações Finais

As redes orientadas a conteúdo representam uma promissora arquitetura para a Internet do Futuro, entretanto, muitos aspectos de segurança e privacidade, características primordiais para não repetir os mesmos problemas da atual Internet, ainda não foram suficientemente explorados.

A arquitetura CCN dispõe de recursos para garantir a autenticidade dos conteúdos, porém, delega à aplicação a responsabilidade pela segurança do conteúdo em caso de necessidade de sigilo.

Deste modo, a aplicação pode aplicar a técnica mais conveniente para cada situação e como de praxe, a criptografia surge como um importante recurso de segurança a ser utilizado para garantir a confidencialidade.

Entretanto, essas novas necessidades não são atendidas de modo ideal pelas técnicas de criptografia já consolidadas. Nesta lacuna surge a criptografia baseada em atributos como uma alternativa interessante a ser explorada no âmbito das redes orientadas a conteúdo.

No próximo capítulo são apresentados trabalhos que exploraram conceitos de privacidade no uso da arquitetura CCN e a utilização da criptografia ABE.

Capítulo 3

TRABALHOS RELACIONADOS

Este capítulo apresenta uma visão geral de trabalhos que abordam os temas privacidade e redes orientadas a conteúdo.

3.1 Considerações iniciais

Há na literatura diversos trabalhos que abordam os conceitos de privacidade em Redes Orientadas a Conteúdo (ROC). Entretanto, a abordagem do conceito de privacidade possui diferentes aspectos.

Os trabalhos de Aryanfar et al. (2011), Lauinger et al. (2012), Mohaisen et al. (2013), Acs et al. (2013) e Chaabane et al. (2013) têm como enfoque descrever os principais riscos à privacidade em ROC. Além da privacidade em relação ao próprio conteúdo, há um enfoque maior sobre os riscos à violação do sigilo dos eventos, ou seja, na garantia de que invasores não descubram os tipos de conteúdos que os usuários estão acessando.

Nestes trabalhos é enfatizada a existência de ataques ao *cache* que poderiam obter informações sobre os conteúdos que um usuário está acessando, mesmo que o roteador CCN não seja invadido. Como complemento, são apresentadas possíveis soluções e técnicas que podem ser utilizadas para minimizar os impactos.

O trabalho de Massawe, Du e Zhu (2013) propõe a alteração do roteamento das redes CCN para um esquema que faz uso de *Bloom Filters*. A proposta altera o modo como as palavras-chave de uma consulta são manipuladas, de modo a garantir o seu sigilo. Neste trabalho não é abordada a segurança do próprio conteúdo.

Em Hamdane et al. (2012) é proposta a utilização de uma solução de segurança que aplica uma infraestrutura de chaves públicas e um esquema hierárquico de Criptografia Baseada em Identidade. O objetivo do trabalho é garantir a segurança tanto do conteúdo quanto da estrutura de nomes.

Os trabalhos de Ion, Zhang e Schooler (2013), Papanis et al. (2013) e Kim et al. (2012), abordam conceitos de privacidade ou soluções mais próximas deste trabalho e, portanto, serão apresentados em maiores detalhes nas seções seguintes.

3.2 Privacidade em ROC: Roteamento e Criptografia Baseada em Atributos

Ion, Zhang e Schooler (2013) incluíram no contexto de ROC, uma camada de aplicação publicador/assinante (*publisher/subscriber*) com suporte a privacidade, que tem como base o trabalho anterior de Ion, Russelo e Crispo (2012). Com essa proposta, os autores realizaram alterações no formato de descrição, encriptação e roteamento de conteúdo de uma ROC.

De acordo com a proposta, os usuários ao enviar pacotes do tipo *Interest* poderão utilizar expressões booleanas que aplicam restrições em relação aos nomes e atributos. Esses parâmetros adicionais contendo restrições permitem que conteúdos inadequados sejam desprezados pela própria rede. Além disso, os dados podem ser criptografados pelo usuário publicador utilizando ABE, que irá estabelecer a política de acesso (ION; ZHANG; SCHOOLER, 2013).

Considerando que a proposta modifica o modo como a CCN realiza o roteamento, foi necessária a inclusão de um *Broker* de roteamento. Este *Broker* possui a responsabilidade pelo gerenciamento das chaves e também da verificação das requisições em relação às restrições estabelecidas, com o intuito de impedir o encaminhamento de pacotes não autorizados.

Além de garantir a confidencialidade dos conteúdos, a proposta visa garantir a privacidade dos usuários em relação ao sigilo de seus interesses por conteúdos. Para isso, as restrições estabelecidas são também criptografadas, o que exige um servidor *proxy* que aplique técnicas de busca em dados criptografados (*Searchable Date*

Encryption – SDE). A Figura 3.1 apresenta a arquitetura proposta aplicada a uma rede CCN e utilizando KP-ABE (ION; ZHANG; SCHOOLER, 2013).

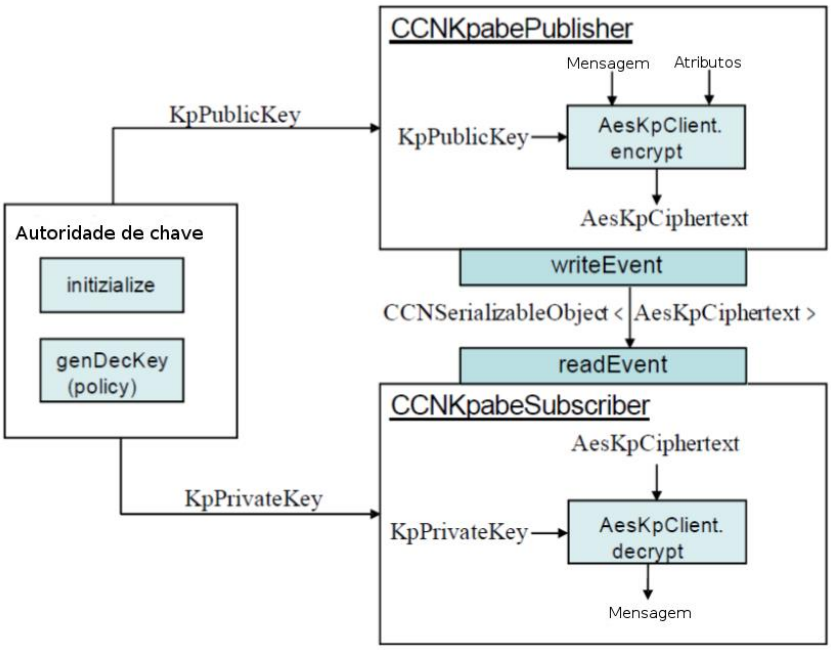

Figura 3.1 - Arquitetura para privacidade em ROC
Fonte: Adaptado de Ion, Zhang e Schooler (2013)

Como é possível visualizar na Figura 3.1, a *Key Authority* possui tanto o papel de *Broker* quanto de servidor *proxy* e, além de gerenciar a distribuição das chaves, intermedia as requisições para impedir o roteamento indevido de pacotes.

3.3 Utilização de ABE para proteção de conteúdo multimídia em ROC

Papanis et al. (2013) desenvolveram uma arquitetura para fornecer distribuição de vídeo sob demanda em uma rede NDN, utilizando ABE para garantir a segurança, juntamente com um esquema de gestão de direitos digitais (*Digital Rights Management* - DRM).

Na proposta dos autores há a descrição do processo de aquisição de licença, o qual faz uso da ABE para garantir a confidencialidade e com possibilidade de existência de múltiplas autoridades de licenciamento.

Neste trabalho também é abordado um problema recorrente da ABE, especialmente para serviços de conteúdo por assinatura, que é a revogação de privilégios. No contexto do trabalho, trata-se de um tópico extremamente importante, pois a necessidade de realizar a criptografia dos conteúdos novamente é inviável em relação ao *cache* realizado em uma ROC.

Papanis et al. (2013) apresentam duas possibilidades de revogação de privilégios para os usuários no contexto de distribuição de vídeo sob demanda.

Na primeira abordagem, os usuários recebem uma chave para acesso que possui um tempo de expiração, que poderia estar relacionado com o período de assinatura, por exemplo.

Na segunda abordagem, que é efetivamente utilizada pelos autores, há uma Autoridade de Revogação, que periodicamente envia uma atualização da chave aos usuários não revogados. Com a chave renovada, os usuários não revogados continuam tendo acesso aos conteúdos, enquanto os usuários revogados perdem o acesso.

O servidor de licenciamento criptografa a licença com base em um atributo do usuário que determina a última atualização de chave enviada pela Autoridade de Revogação. Deste modo, somente os usuários que ainda possuem privilégio serão capazes de acessar o conteúdo.

A Figura 3.2 apresenta a proposta, incluindo o esquema de revogação de privilégios.

Como apresentado na Figura 3.2, o usuário deve se comunicar com o Sistema de Gerenciamento de Assinante (*Subscriber Management System*), que controla os conteúdos em que o usuário está interessado.

Para acessar um conteúdo restrito, o usuário deve obter a licença junto ao Servidor de Licenciamento (*Licence Server*), o qual utiliza a criptografia CP-ABE para enviar a licença ao usuário.

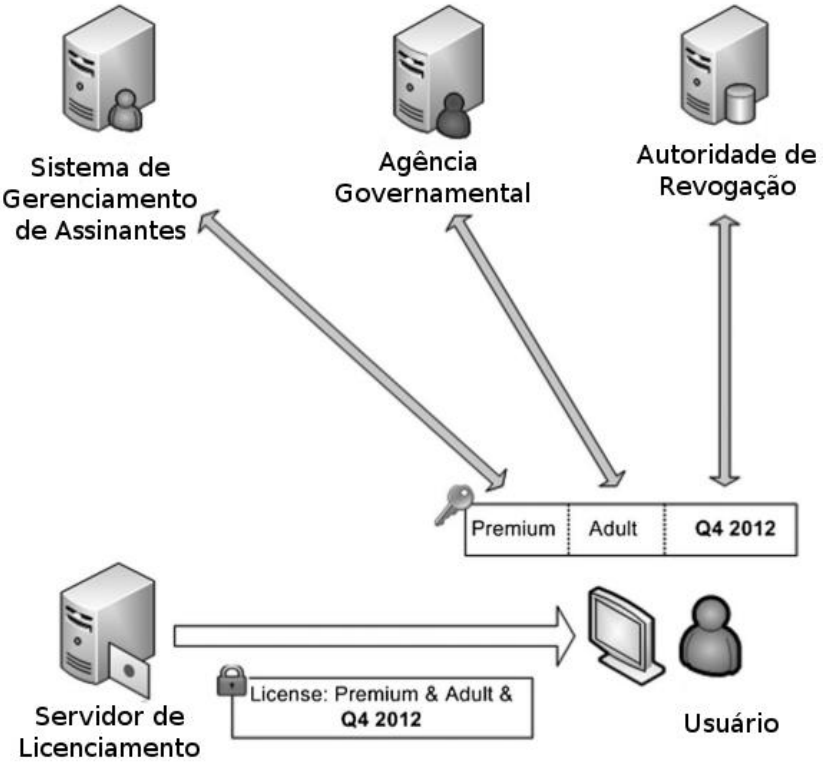

Figura 3.2 - Processo de revogação de privilégios
Fonte: Papanis et al. (2013)

A revogação dos privilégios é feita pela entidade Autoridade de Revogação. A proposta ainda prevê a possibilidade de auditoria e regulação do conteúdo por meio de uma Agência Governamental.

3.4 Compartilhamento de conteúdos com proteção a privacidade utilizando ROC

Kim et al. (2012) propuseram um ambiente de compartilhamento de arquivos protegido e o implementaram utilizando a plataforma Android. A proposta é baseada no uso de uma Comunidade Virtual Particular (*Virtual Private Community – VPC*), a qual permite criar um grupo fechado de usuários, que podem ser gerenciados e organizados hierarquicamente pelo administrador.

Entretanto, o uso da VPC não garante o nível de segurança desejado. Para este fim, os autores incluíram também um esquema hierárquico de pares de chave por grupo.

Kim et al. (2012) baseiam sua proposta no aumento da utilização de dispositivos móveis e outros dispositivos com recursos multimídia e, consequentemente, no crescente tráfego de arquivos multimídia compartilhados entre usuários, não somente dentro do ambiente doméstico.

Os autores afirmam que muitos usuários não confiam em serviços de nuvem para armazenar seus conteúdos pessoais. Deste modo, preferem manter seus arquivos armazenados em um ambiente próprio, especialmente por problemas relacionados à privacidade.

Apesar de ser relativamente simples gerenciar e compartilhar esse conteúdo internamente ao ambiente doméstico, o compartilhamento dos conteúdos com dispositivos externos a este ambiente se torna mais complexo, além de possuir mais riscos.

Ao combinar o esquema de criptografia com o gerenciamento hierárquico de conteúdo provido pela VPC, os usuários são capazes de compartilhar conteúdos, e poderão estabelecer políticas de acesso que deverão ser respeitadas para que seja possível acessá-los.

Na Figura 3.3 é apresentada a arquitetura proposta por Kim et al. (2012).

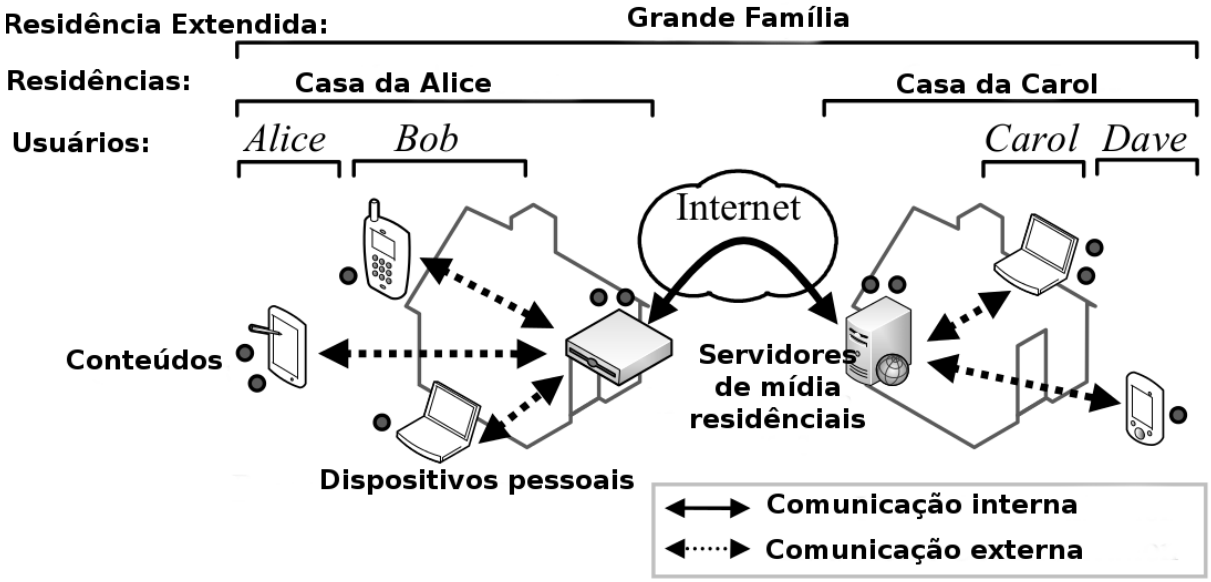

Figura 3.3 - Arquitetura *Extended Home*

Fonte: Adaptado de Kim et al. (2012)

Como é possível visualizar na Figura 3.3, o modelo apresenta dois ambientes separados, a “Casa da Alice”, que possui os usuários “Alice” e “Bob”, e “Casa da Carol”, que contém os usuários “Carol” e “Dave”. Juntos, os ambientes “Casa da Alice” e “Casa da Carol” formam um ambiente estendido “Grande Família”. Estes usuários compõem uma VPC, que permite estabelecer políticas de acesso internas a cada ambiente, ou em relação a todos os ambientes que podem realizar o acesso.

3.5 Considerações finais

A proposta de Ion, Zhang e Schooler (2013) apresenta importantes contribuições em relação à segurança e privacidade nas redes CCN, abordando não somente a confidencialidade dos dados, mas também a garantia do sigilo dos eventos no caso de um ambiente que não seja confiável.

Apesar de introduzir uma interessante funcionalidade de buscas mais refinadas, a alteração proposta pelos autores no esquema de roteamento também sobrecarrega o roteador CCN com uma nova função.

É importante ressaltar que o roteamento ainda é um dos principais desafios para tornar a arquitetura CCN viável na prática.

Além disso, a exigência de um servidor *proxy* que utilize técnica de busca em dados criptografados implica em um custo computacional extra que nem sempre é necessário, tornando ainda mais complexa a viabilidade.

O trabalho de Papanis et al. (2013) tem um contexto específico de distribuição de conteúdo multimídia. A criptografia CP-ABE é aplicada na distribuição do licenciamento e não na proteção do conteúdo em si.

Uma importante contribuição apresentada pelo autor está na discussão de possíveis soluções para um problema recorrente na utilização de ABE, que é a revogação dos privilégios.

O trabalho de Kim et al. (2012) também é aplicado a um contexto específico, que envolve o compartilhamento de arquivos pessoais em um ambiente doméstico estendido. Para estabelecer e aplicar as políticas de acesso, os autores utilizam as técnicas de VPC combinadas a um esquema de segurança que gera pares de chaves para cada grupo.

Apesar de ser uma solução viável para o contexto proposto, essa solução seria inviável em um ambiente com um número maior de usuários, devido ao esquema de gerenciamento de chaves aplicado.

A proposta deste trabalho é apresentar um mecanismo de garantia de privacidade em um contexto onde a aplicação é confiável ao usuário. Deste modo o custo computacional extra da busca por dados criptografados aplicada na solução de Ion, Zhang e Schooler (2013), não é interessante.

Além disso, o mecanismo proposto utiliza uma solução de revogação de privilégios diferente das existentes nos trabalhos de Ion, Zhang e Schooler (2013) e Papanis et al. (2013).

Capítulo 4

MECANISMO DE GARANTIA DE PRIVACIDADE

Este capítulo apresenta a proposta de um mecanismo de garantia de privacidade para aplicações em redes orientadas a conteúdo, que permite aos usuários definir quem pode ter acesso aos conteúdos publicados.

4.1 Considerações iniciais

O objetivo principal do mecanismo de garantia de privacidade para aplicações em redes orientadas a conteúdo é permitir a um usuário decidir quem poderá acessar os conteúdos publicados por ele. Portanto, os conteúdos serão legíveis a outros usuários ou grupos específicos, desde que tenham sido previamente autorizados pelo publicador.

A arquitetura NDN utiliza mecanismos criptográficos para realizar a assinatura dos conteúdos publicados, permitindo que seja possível identificar se o conteúdo realmente é da fonte original, e não uma cópia alterada. Deste modo, mantém-se garantia da autenticidade dos dados.

Considerando o requisito da confidencialidade dos dados, a arquitetura NDN se mantém flexível em relação ao uso de criptografia para garantir o sigilo dos dados. Deste modo, a aplicação se torna responsável por este processo e pode fazer uso do mecanismo criptográfico mais conveniente para cada situação.

Este mecanismo explora a flexibilidade da arquitetura NDN em relação ao uso de recursos de segurança e funcionará no nível de aplicação de uma rede NDN,

criptografando conteúdos sensíveis com a utilização da criptografia baseada em atributos CP-ABE. Para garantir a revogação de privilégios, será utilizado o conceito de *proxy* proposto por Jahid e Borisov (2012).

Chaabane et al. (2013) ressaltam a utilização desse tipo de criptografia como um meio eficaz para manter a confidencialidade dos pacotes e consequentemente a privacidade dos usuários em relação aos seus dados.

O uso de ABE, ao invés dos populares algoritmos de criptografia de chave pública, se justifica principalmente por não exigir a troca de chaves entre o publicador e os assinantes a cada publicação, o que torna complexa a adoção em ambientes com grande volume de dados. Ao utilizar ABE é possível que o conteúdo seja disponibilizado mesmo para um usuário que foi registrado após a publicação e, consequentemente, a criptografia do conteúdo.

4.2 Aplicações alvo

As aplicações desenvolvidas no paradigma publicador-assinante permitem que um usuário disponibilize conteúdo que poderá ser acessado por outros usuários interessados posteriormente. Este processo ocorre de maneira assimétrica, ou seja, o publicador desconhece quem e quando exatamente alguém irá acessar o conteúdo. Nesta situação, caso não ocorra um tratamento em termos de segurança dos dados, o conteúdo poderia ser acessado indevidamente.

Devido à compatibilidade entre as ROC e o paradigma publicador-assinante, essas aplicações são beneficiadas neste contexto, pois independentemente da localização dos dados publicados, um usuário pode ter acesso ao conteúdo manifestando uma requisição de interesse.

Considerando essas características, as aplicações do tipo publicador-assinante executadas em uma ROC podem adotar este mecanismo de garantia de privacidade com o intuito de garantir que o usuário decida quem irá acessar o conteúdo publicado. Além disso, ao adotar o mecanismo, a aplicação mantém a funcionalidade do recurso de *cache* no roteador, o que é essencial para otimizar o desempenho de uma ROC.

O mecanismo proposto pode ser utilizado quando há a necessidade de

compartilhar o conteúdo somente com usuários registrados ou que paguem tarifas pelos serviços, permitindo a revogação de usuários inadimplentes.

Sistemas de compartilhamento de ativos de informação de uma empresa com seus colaboradores também podem se beneficiar do mecanismo. Neste caso, o acesso poderia estar limitado ao escopo da rede interna e os ativos seriam disponibilizados de acordo com o privilégio de acesso existente.

4.3 Características do mecanismo

As chaves utilizadas para cifrar e decifrar os conteúdos serão gerenciadas pela própria aplicação que adota o mecanismo. Cabe à aplicação que adotou o mecanismo, portanto, realizar a criação e distribuição das chaves privadas aos usuários com base em seus atributos. Essas chaves podem ser mantidas pela aplicação no *host* do usuário ou armazenadas em um *token* ou cartão inteligente, de modo que o usuário precise tê-la disponível para realizar a descriptografia e, consequentemente, acessar o conteúdo. Ou ainda, podem ser mantidas pela aplicação de modo transparente ao usuário, onde a chave seja manipulada pela aplicação após a autenticação do usuário, sem que ele tenha que informar diretamente a sua chave privada.

Os atributos fornecidos ao usuário serão utilizados em uma estrutura de acesso, atuando como requisitos definidos pelo publicador, o que irá estabelecer a política de acesso aos conteúdos.

O mecanismo exige que a aplicação verifique o privilégio do usuário em relação aos atributos que possui, pois a política de acesso ao conteúdo é dependente deste processo. Caso um usuário obtenha um atributo ao qual não deveria possuir privilégio, será capaz de acessar conteúdos de outros usuários indevidamente.

Há um pressuposto de que a aplicação é confiável para os usuários, pois como responsável pelo gerenciamento das chaves de criptografia, seria capaz de acessar os conteúdos realizando a decriptação dos mesmos.

Diante da capacidade da aplicação em verificar o conteúdo, permite-se a extensão do mecanismo para a criação de um módulo de auditoria, com o objetivo de

garantir a legalidade e a irretratabilidade. Esta característica é um requisito desejável a esta proposta, uma vez que a adoção do mecanismo de garantia de privacidade não tem como objetivo garantir a troca anônima de conteúdos.

Na Figura 4.1 é apresentada a inserção do mecanismo na arquitetura NDN. Como é possível visualizar, o mecanismo funciona no nível de aplicação.

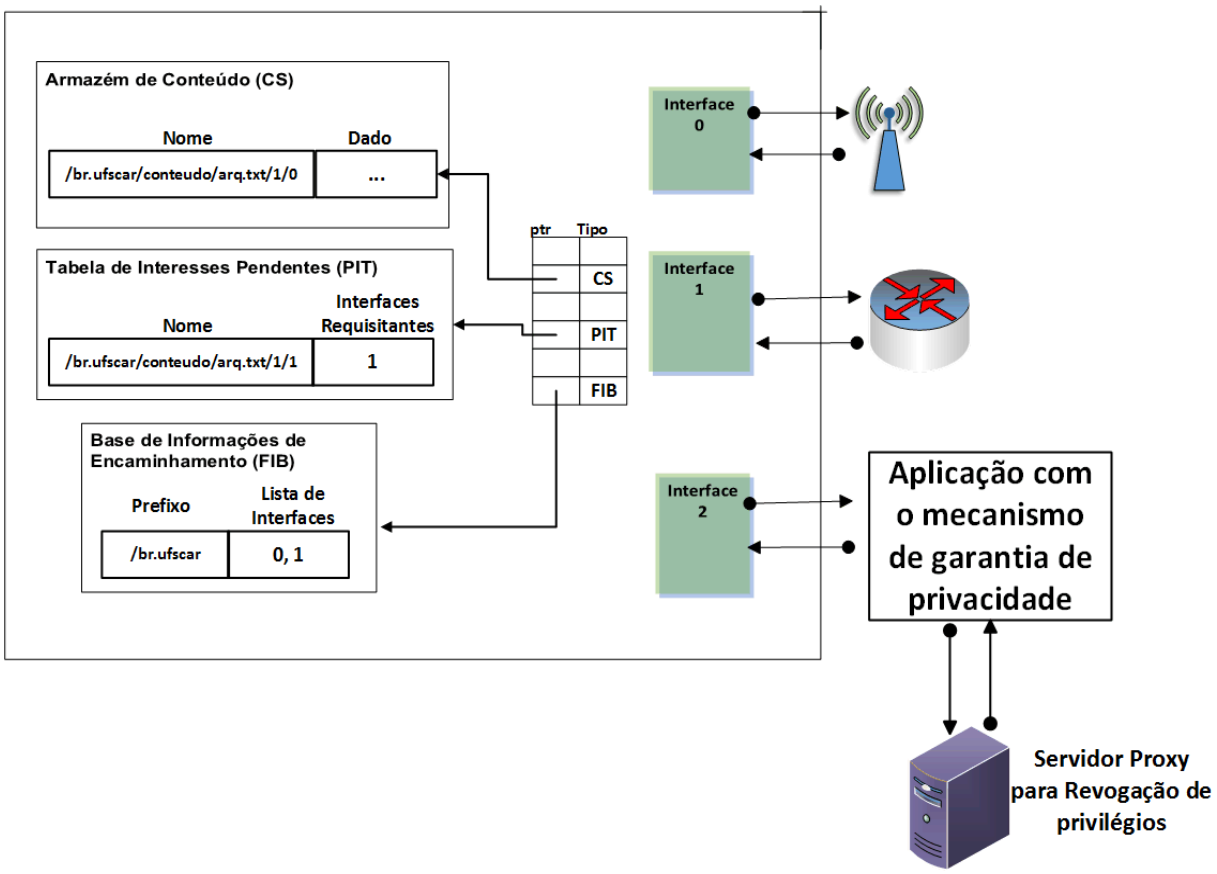

Figura 4.1 – Arquitetura NDN com o mecanismo de garantia de privacidade
Fonte: Adaptado de Zhang et al. (2010)

4.4 Funcionamento básico

Na Figura 4.2 é apresentada uma visão geral do funcionamento do mecanismo proposto com a atuação do *proxy*.

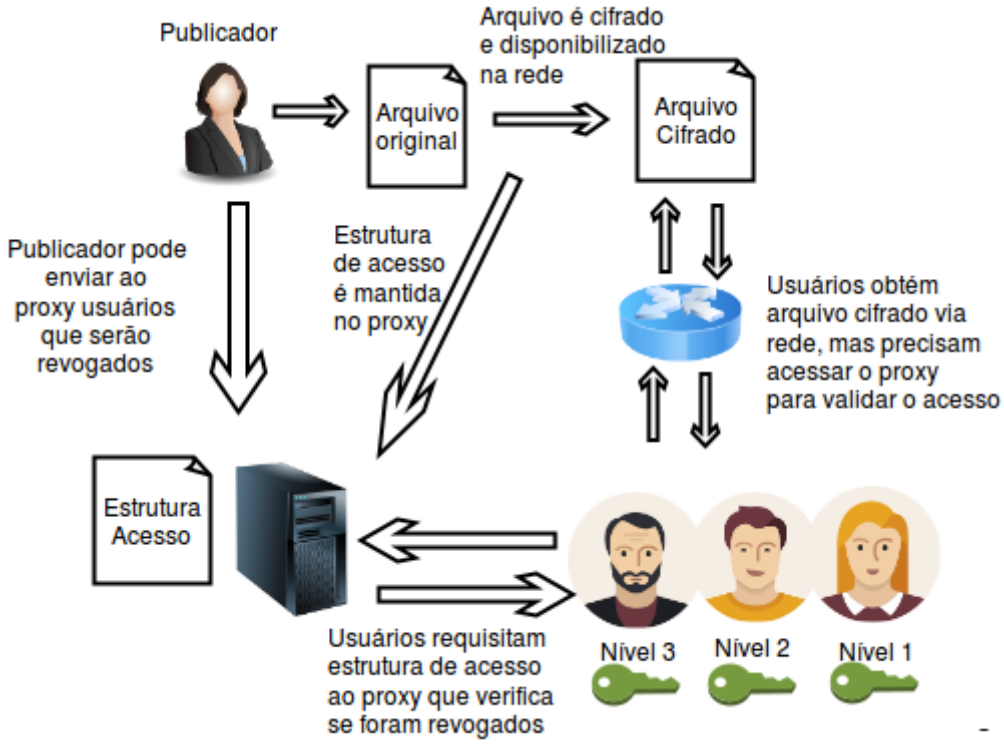

Figura 4.2 – Visão geral do mecanismo

Como ilustrado na Figura 4.2, o passo inicial será o usuário publicador informar à aplicação o arquivo que deseja compartilhar. Neste ponto, devem ser informados os atributos necessários para que outros usuários acessem o arquivo.

A aplicação irá cifrar o conteúdo e irá, utilizando os atributos fornecidos, criar a estrutura de acesso para em seguida publicá-lo. Por intermédio da arquitetura NDN, outros usuários podem manifestar interesse pelo conteúdo.

Caso o usuário publicador deseje revogar o privilégio de algum usuário, irá informar as identidades dos mesmos. Se não houver revogações, o usuário publicador não precisa contatar o *proxy.*

Ao receber o arquivo, a aplicação irá enviar ao *proxy* a estrutura de acesso. Neste momento, o *proxy* irá verificar se há restrições para o usuário e, caso não exista, devolverá a estrutura de acesso para que a aplicação realize a descriptografia.

As funções do mecanismo estão ilustradas na Figura 4.3.

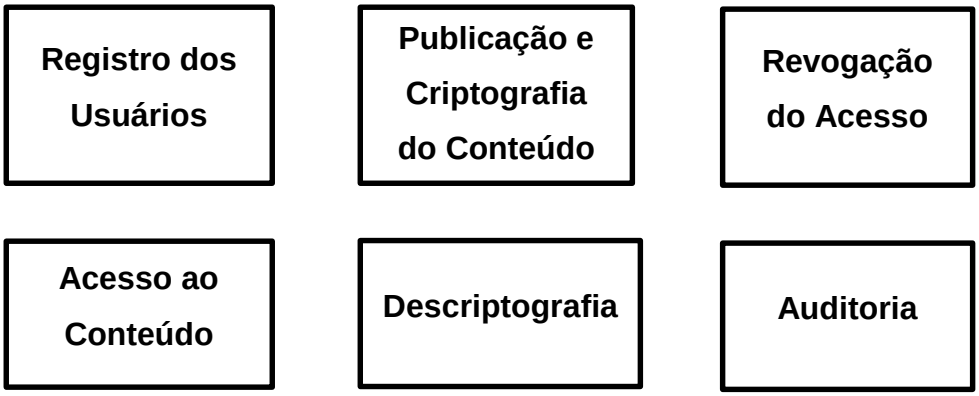

Figura 4.3 – Funções do mecanismo

Nas sub-seções a seguir cada uma das funções do mecanismo será descrita em detalhes.

4.4.1 Registro dos usuários

Cada um dos usuários que terá acesso ao conteúdo deve ser previamente registrado na aplicação. Porém, antes de realizar o registro de cada usuário é necessário cadastrar previamente os atributos que poderão ser concedidos aos usuários posteriormente.

A aplicação é totalmente livre para determinar ou limitar o tipo de atributo que poderá ser utilizado. Dependendo do contexto da aplicação pode ser útil a utilização de um sistema de validação deste processo, como por exemplo, um usuário administrador realizar a confirmação dos atributos cadastrados.

A distribuição dos atributos aos usuários é um processo fundamental no funcionamento do mecanismo, pois o acesso é liberado através destes atributos concedidos ao usuário. Caso ocorra uma atribuição incorreta, obviamente, permitirá ao usuário o acessar um conteúdo que não deveria ser autorizado.

A utilização destes atributos é totalmente transparente à rede, deste modo qualquer combinação pode ser utilizada pela aplicação

É possível criar novos atributos e atribuí-los aos usuários já existentes, porém, o processo de geração de chaves deve ser refeito.

Durante este registro, os usuários receberão um conjunto de atributos que servirá para realizar a sua identificação.

Um modo de utilizar os atributos é seguir a classificação de acordo com um processo de classificação de ativos da informação, como sugerido na norma ISO 27002. Na Tabela 4.1 é possível visualizar um exemplo de classificação de ativos por níveis de confidencialidade.

Tabela 4.1 – **Exemplo de classificação de ativos através de níveis de confidencialidade para gerar atributos**

Ativo	Nível de confidencialidade	Acessível para
Plano estratégico	1	Diretores e níveis superiores
Planejamento financeiro	2	Gerentes e níveis superiores
Dados de estoque	3	Colaboradores internos

Neste exemplo, os ativos foram classificados de acordo com o nível de confidencialidade desejado, sendo o nível 1 o mais completo e o nível 3 o que possui menos privilégios. Deste modo, o ativo “Plano estratégico” poderia ser acessível apenas por usuários que possuem o nível 1 de confidencialidade, enquanto o ativo “Planejamento financeiro”, pode ser acessado por quem tem nível 2 ou 1, e o ativo “Dados de estoque” pode ser acessado por usuários com nível 3, 2 ou 1.

Em relação ao mecanismo, o nível seria o atributo a ser relacionado com o usuário. Uma vez atribuído um nível, o usuário será capaz de acessar tanto os arquivos que já foram criptografados e disponibilizados, quanto os futuros arquivos que venham a ser publicados.

Ao concluir o registro, o usuário recebe uma chave privada (SK) e uma identificação (ID), que serão necessárias para os processos de conversão e descriptografia. Com o intuito de garantir a confidencialidade destas chaves, a transmissão da SK e ID podem utilizar um processo similar ao adotado na Internet ao utilizar Transport Layer Security (TLS).

4.4.2 Publicação e criptografia do conteúdo

Após a conclusão do processo de registro, os usuários estão de posse da chave privada (SK) e da identificação (ID), e, portanto, estão aptos a realizar a publicação de conteúdos para serem distribuídas através da rede orientada a conteúdo.

O usuário irá fornecer o arquivo com o conteúdo e definirá a política de acesso necessária para realizar a descriptografia do conteúdo. Então, utilizando a SK do

usuário e a PK, o sistema irá realizar a criptografia do conteúdo. Este processo está ilustrado na Figura 4.4.

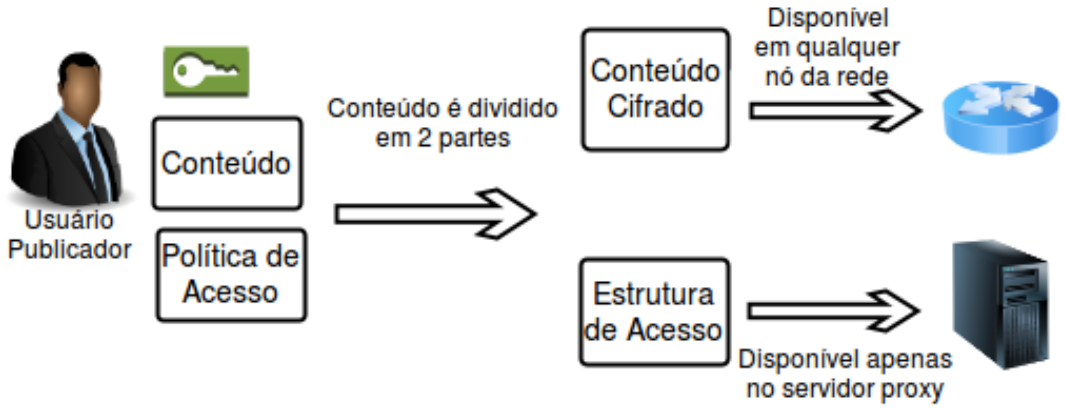

Figura 4.4 – Processo de publicação/criptografia

Como é possível perceber, o conteúdo é dividido em duas partes alocadas em dois pacotes distintos. Na primeira parte está o conteúdo criptografado em si, enquanto a segunda mantém a política de acesso armazenada também de maneira ilegível aos usuários.

Os atributos escolhidos para compor a política de acesso podem ser combinados com regras de *AND e OR.* Por exemplo, apenas o grupo de usuários classificados como “nível 1” ou “nível 2”, poderiam acessar o conteúdo.

Caso o conteúdo tenha um tamanho acima do limite máximo para um pacote da rede, ele deverá ser dividido em pedaços compatíveis com a arquitetura de rede utilizada.

Neste caso, é preferível para a aplicação criptografar todo o conteúdo e depois dividí-lo em partes adequadas ao limite da rede, sendo o conteúdo remontado no usuário final antes da realização da descriptografia.

Deste modo, é possível manter um único pacote contendo a política de acesso para vários pacotes de conteúdo, uma vez que a política de acesso sofre pouca variação de tamanho em relação ao tamanho do conteúdo.

Isso ajuda a diminuir o fluxo de pacotes na rede, pois, caso a separação do conteúdo original fosse realizada antes da criptografia, seria necessário criar uma estrutura de acesso para cada uma das partes do conteúdo.

A vantagem de separar o conteúdo da política de acesso consiste no fato de poder utilizar regras de *cache* diferentes, visando melhorar o desempenho da rede. Considerando um conteúdo que seja maior que a estrutura de acesso, é possível mantê-lo em um *cache* mais próximo aos usuários, de modo a evitar longas transferências de dados. Isso é particularmente útil para grandes conteúdos, mas também se torna útil para conteúdos acessados frequentemente, neste caso, independentemente do tamanho do arquivo.

Por outro lado, a estrutura de acesso possui um tamanho pequeno e não ficará em *cache*, sendo mantida exclusivamente no servidor *proxy*. Com isso, o *proxy* será responsável por intermediar e controlar o acesso ao conteúdo, permitindo verificar se houve a revogação do acesso. Como o tamanho do pacote contendo esta estrutura é pequeno, uma transferência de longa distância não irá causar tanto impacto, quanto seria caso fosse necessário transferir todo o conteúdo.

4.4.3 Revogação de acesso

Quando um usuário publica um determinado conteúdo, este estará liberado para acesso por qualquer outro usuário que tenha uma SK com atributos que atendam a política de acesso definida durante a publicação do conteúdo. Inclusive para usuários que tenham sido registrados depois da publicação do conteúdo.

Entretanto, pode ser necessário ao usuário publicador revogar o acesso de algum usuário especifico, independente do motivo, sem que seja necessário revogar o acesso dos demais usuários.

Nesta situação, o usuário publicador deverá informar na aplicação quais usuários devem ser revogados para um determinado conteúdo. A aplicação enviará ao *proxy* os ID’s dos usuários revogados e o *proxy* irá criar uma nova chave do *proxy*, que é utilizada para converter a estrutura de acesso de modo que esteja acessível somente aos usuários não revogados. Este procedimento será realizado toda vez que algum usuário for revogado e é apresentado na Figura 4.5.

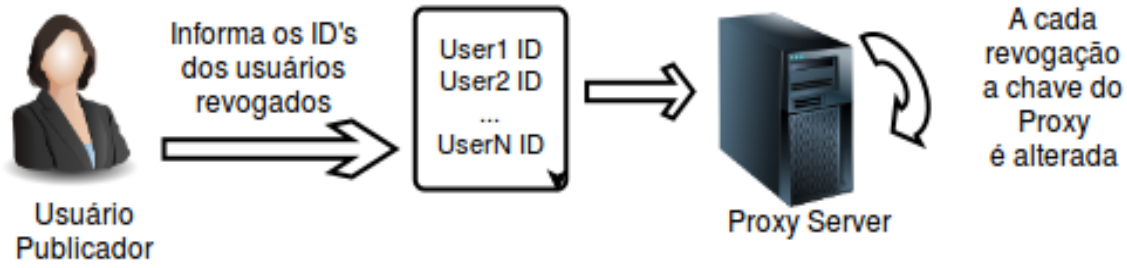

Figura 4.5 – Processo de revogação de usuários

Caso a aplicação esteja em um local diferente do *proxy*, pode ser utilizada criptografia para garantir o sigilo dos ID's que serão revogados durante a transmissão entre a aplicação e o servidor *proxy*.

4.4.4 Acesso ao conteúdo

Quando um usuário solicita acesso a algum conteúdo, a aplicação deverá requisitar o conteúdo seguindo o padrão de nomes e o processo de resolução da arquitetura de rede utilizada. O processo de resolução é ilustrado na Figura 4.6.

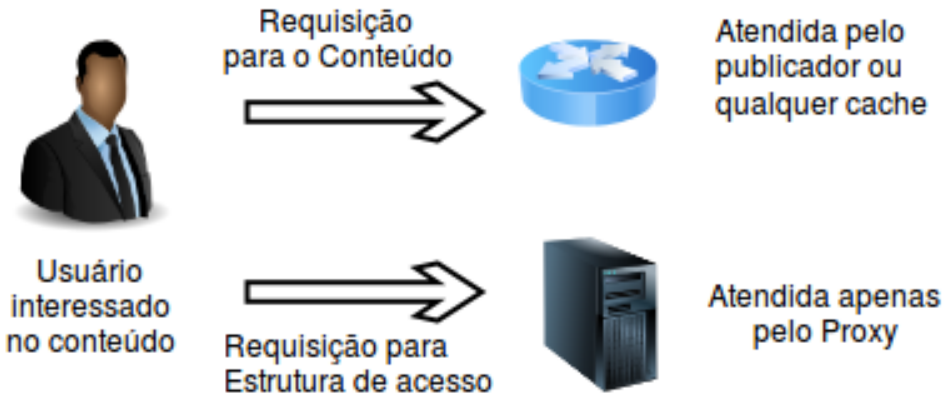

Figura 4.6 – Processo de acesso ao conteúdo

Ao menos dois pacotes serão requisitados, sendo que a aplicação pode requisitá-los em qualquer ordem e, inclusive, ao mesmo tempo. Porém, o conteúdo só estará acessível após ambos os pacotes estarem disponíveis uma vez que um dos pacotes contém o conteúdo em si, enquanto o outro armazena a estrutura de acesso necessária para descriptografar o conteúdo.

Um dos pacotes contém o conteúdo em si, o qual pode ser respondido pelo publicador original ou por qualquer dispositivo mais próximo do usuário que o tenha em *cache*. Caso o conteúdo seja maior que o tamanho máximo para um pacote da arquitetura de rede utilizada, será necessário receber vários pacotes para formar o conteúdo original, ficando a aplicação responsável por extrair o conteúdo de cada pacote e remontá-lo em um único arquivo.

A outra requisição envolve o pacote contendo a estrutura de acesso necessária para acessar o conteúdo. Na requisição da estrutura de acesso deverá ser enviada junto ao pacote de requisição a ID do usuário requisitante. Esta informação será utilizada pelo *proxy* durante a conversão da estrutura de acesso. Esse ID fornecido junto à requisição pode, opcionalmente, ser criptografado.

4.4.5 Re-encriptação por *proxy*

Após a aplicação enviar o pacote requisitando a estrutura de acesso, o mesmo será recebido pelo servidor *proxy* que atenderá ao pedido. O processo de re-encriptação por *proxy* é ilustrado na Figura 4.7.

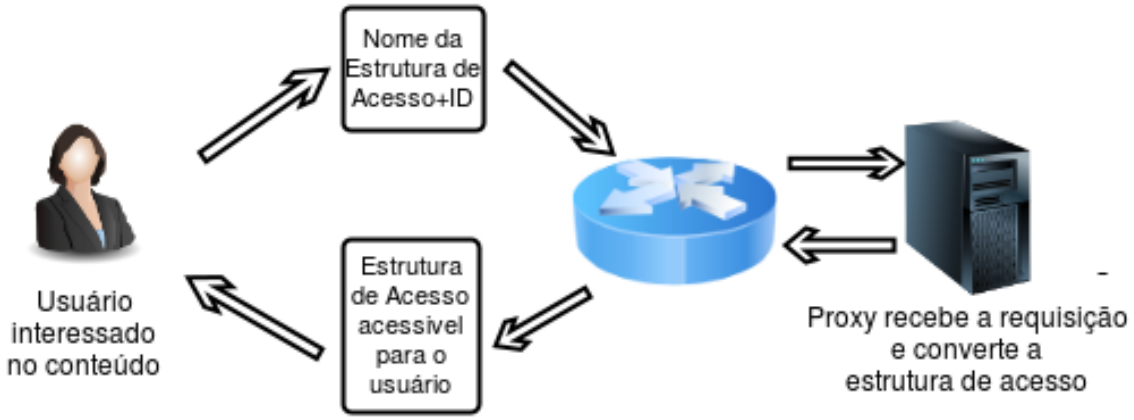

Figura 4.7 – Processo de re-encriptação por *proxy*

Caso o ID do requisitante esteja criptografado, inicialmente o servidor *proxy* deverá realizar a descriptografia desta informação.

Utilizando sua chave de *proxy*, o servidor *proxy* irá converter a estrutura de acesso, de modo que ela se torne legível ao requisitante apenas se o usuário não tiver sido revogado.

Por fim, o servidor *proxy* encaminha o pacote contendo a estrutura de acesso ao requisitante, de acordo com o esquema de encaminhamento utilizado pela arquitetura de rede.

Este pacote não poderá ser armazenado em *cache*, de modo que o usuário tenha que requisitá-lo ao *proxy*.

4.4.6 Descriptografia

Após receber todo o conteúdo e a estrutura de acesso já convertida pelo *proxy*, a aplicação pode realizar a descriptografia para acessar o conteúdo.

Para realizar este processo, a aplicação utiliza a SK do usuário, que contém os atributos concedidos durante o registro. O processo de descriptografia analisa se os atributos do usuário atendem à estrutura de acesso definida pelo publicador e que foi obtida por intermédio do servidor *proxy*.

Combinando a estrutura de acesso e a SK do usuário, a aplicação irá descriptografar o conteúdo e apresentar o conteúdo original ao usuário. Este processo está representado na Figura 4.8.

Figura 4.8 – Processos de re-encriptação e de descriptografia

4.4.7 Auditoria

A auditoria é uma função opcional do mecanismo e pode ser inserida em situações em que o requisito de legalidade exige o registro de eventos. Por exemplo,

em uma empresa que compartilha ativos de informações utilizando o mecanismo, ações como a publicação e o acesso a um conteúdo podem ser registrados.

O processo de auditoria tem o intuito de suprir a possibilidade do acesso anônimo a um conteúdo. Isso pode ocorrer em uma ROC principalmente quando o conteúdo é obtido a partir de um *cache*.

Basicamente, o registro pode ser realizado no momento da publicação, armazenando o horário e o autor proprietário do conteúdo e no momento do acesso, quando o usuário fornece o ID para a conversão da estrutura de acesso, podendo o *proxy* registrar o usuário requisitante.

4.4.8 Considerações finais

Neste capítulo foi apresentado um mecanismo capaz de prover ao usuário a possibilidade de compartilhar seus arquivos com outros usuários individuais ou grupos de usuário, de modo que possa definir políticas de acesso com alto grau de detalhamento.

O mecanismo é capaz de garantir a privacidade ao assegurar que seja mantido o sigilo imposto pelo usuário no momento da publicação. Além disso, caso o usuário publicador deseje retirar o direito de acesso de um usuário autorizado previamente, o mecanismo permite a revogação imediata do acesso sem que seja necessário a re-encriptação e redistribuição do conteúdo.

No caso da arquitetura NDN utilizada nos testes, o roteador poderia ainda ser alterado para agregar as funcionalidades do *proxy* de modo que não seja necessário acrescentar um novo elemento ao processo de acesso ao conteúdo.

Essa solução de integrar as funções do *proxy* ao roteador NDN pode ser útil em um ambiente de menor porte, no qual o número de roteadores que realizam *cache*, através do armazém de conteúdos, seja pequeno, de modo que poderia aumentar a performance ao eliminar a necessidade de um equipamento adicional.

Entretanto, em um ambiente com alto número de roteadores realizando *cache* e alta mobilidade dos usuários, a sincronização das informações a todos os roteadores é um grande desafio a ser resolvido, e poderia resultar em inviabilidade de

desempenho. Neste caso, um *proxy* centralizado, como proposto no mecanismo, se torna mais viável.

Capítulo 5

PROVA DE CONCEITO

Este capítulo apresenta como foi realizada a avaliação do mecanismo de garantia de privacidade para aplicações em redes orientadas a conteúdo e os resultados obtidos nos testes.

5.1 Considerações iniciais

A viabilidade do mecanismo proposto foi avaliada através de uma prova de conceito que inclui o desenvolvimento de uma aplicação de troca de arquivos com restrições de acesso entre os usuários de um sistema e da análise de desempenho das funções principais do mecanismo.

A aplicação, que utiliza o mecanismo proposto, foi desenvolvida de acordo com as limitações existentes no simulador *ndnSIM* versão 2.0, como a não existência de interface gráfica e de interação por meio de linha de comando, e a limitação para acesso a disco.

Deste modo, foram criados um produtor, um consumidor e um *proxy*, trocando pacotes com conteúdo criptografado e com políticas de acesso que foram passadas diretamente no código da aplicação.

Na

Figura 5.1 é apresentado uma captura de tela do simulador durante a execução.

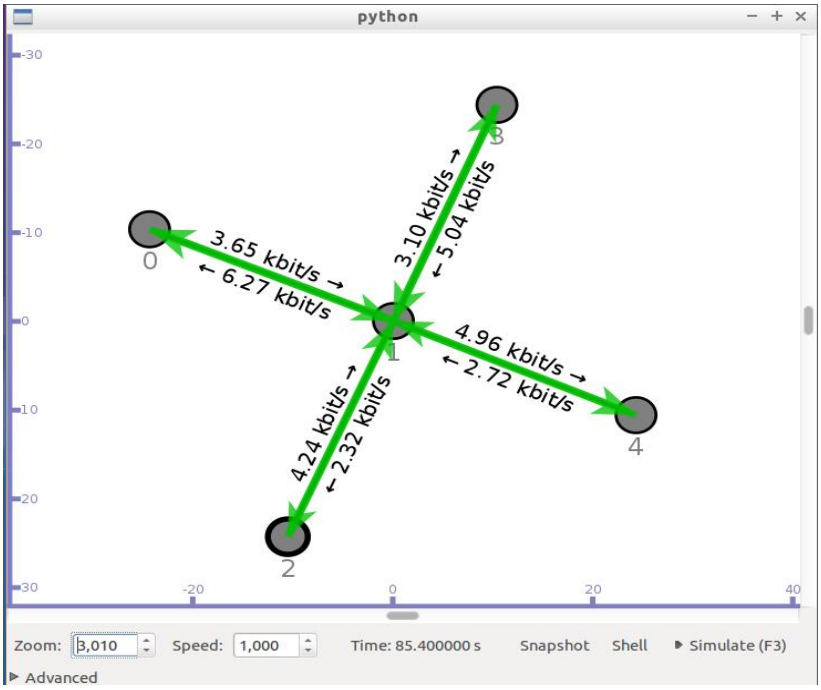

Figura 5.1 – Nós durante simulação

O nó 0 e o nó 2 possuem a aplicação consumidor instalada, e realizam requisições tanto do conteúdo quanto da estrutura de acesso. Ao receber as respostas de suas requisições, o consumidor imprime o conteúdo do pacote.

O nó 3 possui a aplicação produtor, que responde às requisições dos consumidores com um conteúdo criptografado.

O nó 4 possui a aplicação *proxy* instalada e é responsável por atender as requisições da estrutura de acesso realizadas pelos consumidores.

O nó 1 é um roteador NDN que interliga os demais nós.

Como o simulador não permite interação por linha de comando e apresenta restrições em relação ao acesso a disco, o conteúdo foi passado para o pacote diretamente nas aplicações produtor e *proxy*.

Na Figura 5.2 é apresentada a captura de tela com a saída do simulador, após o pacote ser recebido pela aplicação consumidor.

Figura 5.2 – Pacotes com conteúdo criptografado

Na Figura 5.2 é possível identificar que ambos os pacotes (conteúdo e estrutura de acesso) são entregues aos consumidores. Neste exemplo, o pacote contendo o conteúdo distribuído pelo produtor possui 2034 bytes, enquanto o pacote contendo a estrutura de acesso fornecido pelo *proxy* contém 1852 bytes.

Para que o pacote da estrutura de acesso não entre em *cache* é necessário configurar o "*freshness selector*" disponível no pacote NDN de modo que uma versão mantida em *cache* não satisfaça o requisito.

O mecanismo proposto faz uso da técnica de criptografia CP-ABE juntamente com a re-encriptação por *proxy* para garantir a confidencialidade e a privacidade do usuário ao prover políticas de acesso de alta granularidade e permitir a revogação imediata de acesso.

As funções do CP-ABE utilizadas são provenientes do *toolkit PIRATTE (Proxy-based Immediate Revocation of ATTribute-based Encryption)*[1], desenvolvido com base no trabalho de Jahid e Borisov (2012). Este toolkit, licenciado sob a GPL, possui um esquema de revogação para CP-ABE e faz uso da biblioteca PBC. O *toolkit* é dividido em duas partes, sendo a biblioteca principal a *libbswabe-piratte*, e uma interface chamada *PIRATTE*.

Para avaliar o desempenho do mecanismo, foram executados testes de desempenho em todas as funções que envolvem o mecanismo de garantia de

[1] http://hms.isi.jhu.edu/acsc/piratte/

privacidade para que fosse possível aferir o *overhead* de cada um dos processos e verificar a viabilidade de implantação.

Esses testes envolvem a análise das funções de geração de chaves, criptografia, descriptografia e conversão da estrutura de acesso, com o intuito de verificar se o mecanismo é viável na prática ou quais as suas limitações (SILVA e ZORZO, 2015).

Os testes foram realizados em uma máquina virtual com sistema GNU/Linux utilizando o kernel versão 3.6.12 sendo executado em um processador Inter Core I7-3612QM. Na configuração da máquina virtual, um único núcleo do processador foi disponibilizado para executar tanto o sistema operacional convidado quanto o a aplicação. 1GB de memória RAM, de um total de 8GB disponíveis foi alocado para a máquina virtual.

Para evitar que os testes sofressem influência de outros fatores, cada teste foi repetido cinco vezes para obtenção das médias apresentadas nos gráficos abaixo. Após cada um dos testes, os arquivos foram verificados para avaliar se a integridade do arquivo foi alterada.

Foram utilizados três diferentes tipos de arquivos, com o objetivo de avaliar se há impacto no tamanho de arquivo, uma vez que o formato não causa influência. Os arquivos são:

- Documento no formato Open Document Text (ODT) contendo 208 KB;
- Documento no formato Portable Document Format (PDF) contendo 4,1MB;
- Arquivo executável (formato EXE) com 30MB de tamanho.

Considerando que cada atributo irá compor um nó folha na estrutura de acesso em árvore, os testes utilizaram políticas de acesso contendo, 1, 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90 e 100 para cada arquivo, com o intuito de avaliar o impacto da política de acesso no desempenho.

Entretanto, cabe ressaltar que o valor de cem atributos pode ser considerado um valor alto para uma política de acesso, uma vez que diversos usuários podem ser agrupados em um único atributo. Baseado nisso, houve mais pontos de testes nas políticas contendo entre 1 e 20 atributos.

Para fins de comparação e com o intuito de obter valores de referência, os mesmos arquivos foram submetidos a testes utilizando o algoritmo RSA com uma chave de 2048 bits. A implementação do RSA utilizada é disponibilizada pelo software OPENSSL versão 1.0.1f e os testes foram realizados utilizando o mesmo hardware e sistema descrito anteriormente.

5.2 Resultados

Os resultados foram organizados em tamanho do arquivo, consumo de processamento e consumo de memória, os quais são apresentados nas seções a seguir. Além disso, testes adicionais foram realizados contendo uma política de acesso de 5 atributos e um número maior de arquivos para avaliar o desempenho do processo de conversão.

5.2.1 Tamanho do arquivo

Os testes envolvendo o tamanho dos arquivos após a criptografia têm como objetivo avaliar o impacto no armazenamento de dados, bem como no *overhead* para realização da transferência dos arquivos.

Caso o tamanho do arquivo após a aplicação da criptografia seja muito grande em relação ao arquivo original, o mecanismo pode ser inviável para aplicações que exigem um grande volume de dados.

Nos testes realizados foi detectado que o tamanho do arquivo sofre influência de acordo com o número de atributos utilizados na estrutura de acesso. Os resultados podem ser observados na Figura 5.3. Cabe ressaltar que o valor apresentado no gráfico inclui a soma das duas partes.

No gráfico apresentado é demonstrada a variação do arquivo criptografado em relação ao tamanho original do arquivo.

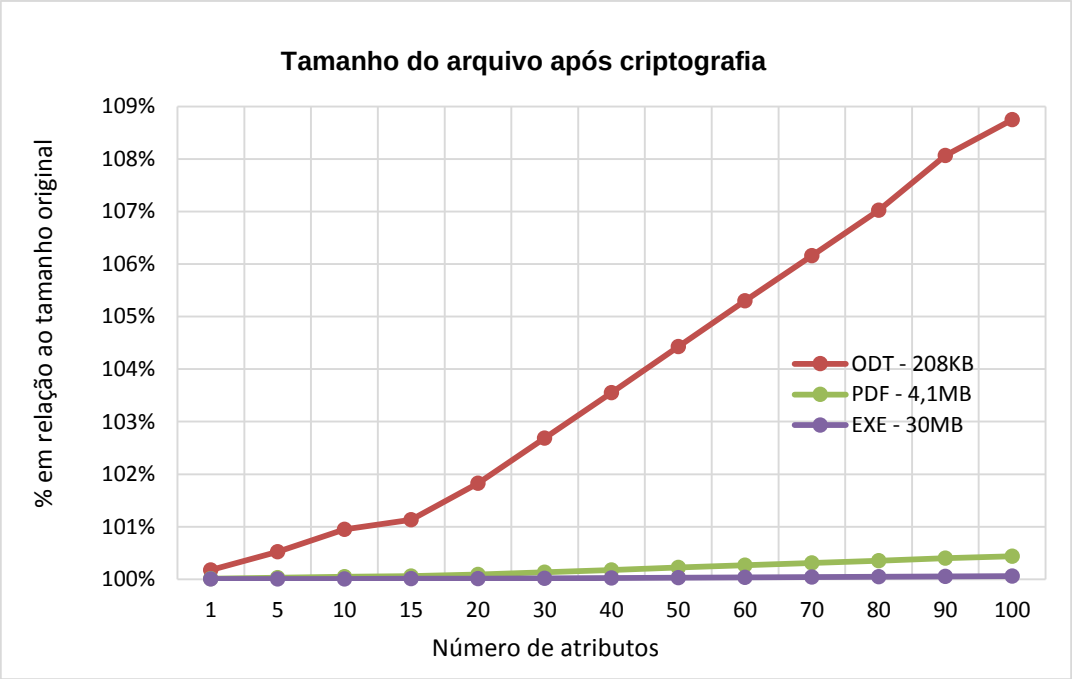

Figura 5.3 – Relação entre o tamanho do arquivo e a quantidade de atributos da politica

Considerando as partes separadas (conteúdo e estrutura de acesso), a parte referente ao conteúdo se mantém fixa, independentemente da quantidade de atributos utilizada. No caso do arquivo ODT, o valor foi 208,05 KB para o arquivo criptografado ante 208,03 KB no arquivo original; para o arquivo PDF, 4.179,39 KB no arquivo criptografado contra 4.178,38 KB no arquivo original; e para o arquivo EXE, 31.484,29 KB no arquivo criptografado versus 31484,27 KB no arquivo original.

Na parte da estrutura de acesso, o tamanho varia em relação ao número de atributos, sem sofrer variação por conta do tamanho do arquivo. Deste modo, esta parte do arquivo variou de 351 bytes, quando utilizando apenas 1 atributo, chegando a 18,18 KB quando utilizando 100 atributos.

Deste modo, fica claro o motivo da variação percentual no menor arquivo (208 KB) ser maior em relação aos demais arquivos, uma vez que a variação é influenciada mais pelo número de atributos, do que pelo tamanho original do arquivo. Como os outros dois arquivos possuem um tamanho maior irão sofrer uma diferença mínima em termos percentuais, mesmo utilizando um número alto de atributos na política de acesso.

Para fins de comparação, os mesmos arquivos foram testados utilizando o algoritmo RSA com uma chave de 2048 bits e sem compactação dos arquivos. Como

não há número de atributos no algoritmo RSA, os resultados são fixos para os três tipos de arquivos. Nestas condições, a variação foi de 136,04% para o arquivo de 208K, 135,81% para o arquivo de 4,1MB, e 132,13% para o arquivo de 30MB.

Diante dos resultados, verificou-se que o *overhead* resultante no tamanho dos arquivos é aceitável, inclusive em relação à outros mecanismos de criptografia, pois a variação de tamanhos não comprometeria a transmissão ou armazenamento.

5.2.2 Tempo de processamento

Os testes em relação ao processamento têm como objetivo analisar o tempo dispendido para o processador executar os principais processos do mecanismo proposto. Estes testes foram realizados com a intenção de verificar as limitações do mecanismo proposto em termos de desempenho.

Inicialmente foi analisado o tempo gasto para executar a geração das chaves, realizada durante o registro do usuário, quando ele recebe a sua SK baseada nos atributos que lhe foram concedidos.

Na Figura 5.4 foram apresentados os tempos gastos em segundos para gerar as chaves.

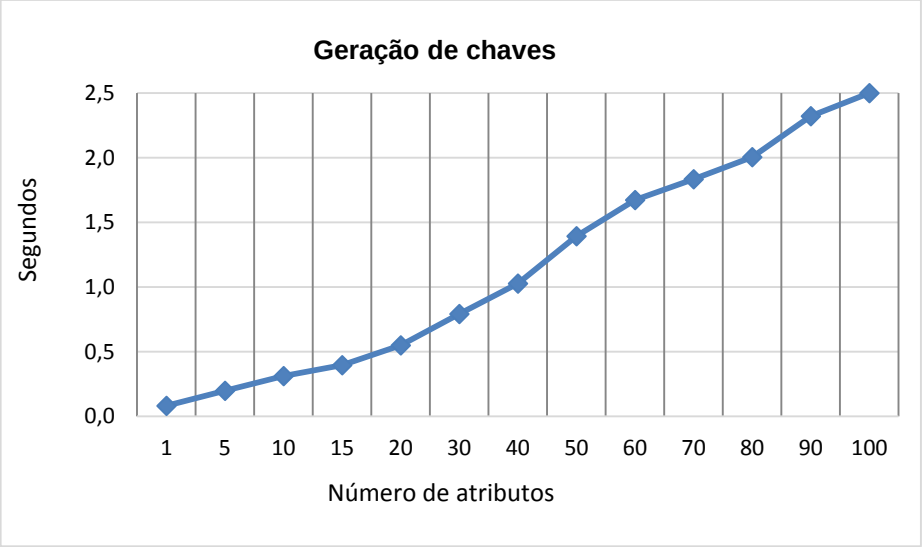

Figura 5.4 – Tempo gasto para gerar a chave do usuário

Analisando os dados coletados, foi identificado que para gerar uma chave com um único atributo, foram gastos 80 milissegundos, sendo necessário utilizar 20 atributos para ultrapassar a marca de 500 milissegundos. Com um tempo gasto abaixo

de 1 segundo é possível gerar uma chave com cerca de 40 atributos, o que já consiste em um número grande de atributos. Para uma chave contendo 100 atributos, foram gastos 2,5 segundos.

Para fins de comparação, uma chave RSA de 2048 bits gerada no mesmo sistema, levou em média 160 milissegundos para ser criada, o que demonstra um valor próximo às chaves com menos de 5 atributos geradas pelo algoritmo de criptografia CP-ABE adotado pelo mecanismo.

Uma vez que este processo de geração de chaves não é realizado frequentemente para o mesmo usuário e só seria refeito em caso de alteração dos atributos, os valores obtidos nos testes foram considerados aceitáveis.

Durante a publicação do conteúdo, ocorre a criptografia do mesmo, sendo que este processo não precisa ser refeito para a posterior revogação de acesso de um usuário.

Dependendo da implantação da aplicação adotada, este processo pode ser executado no *host* do usuário publicador, distribuindo assim o processamento e evitando a sobrecarga no *proxy*, o que afetaria todos os usuários. Na Figura 5.5 estão os resultados obtidos com o processo de criptografia.

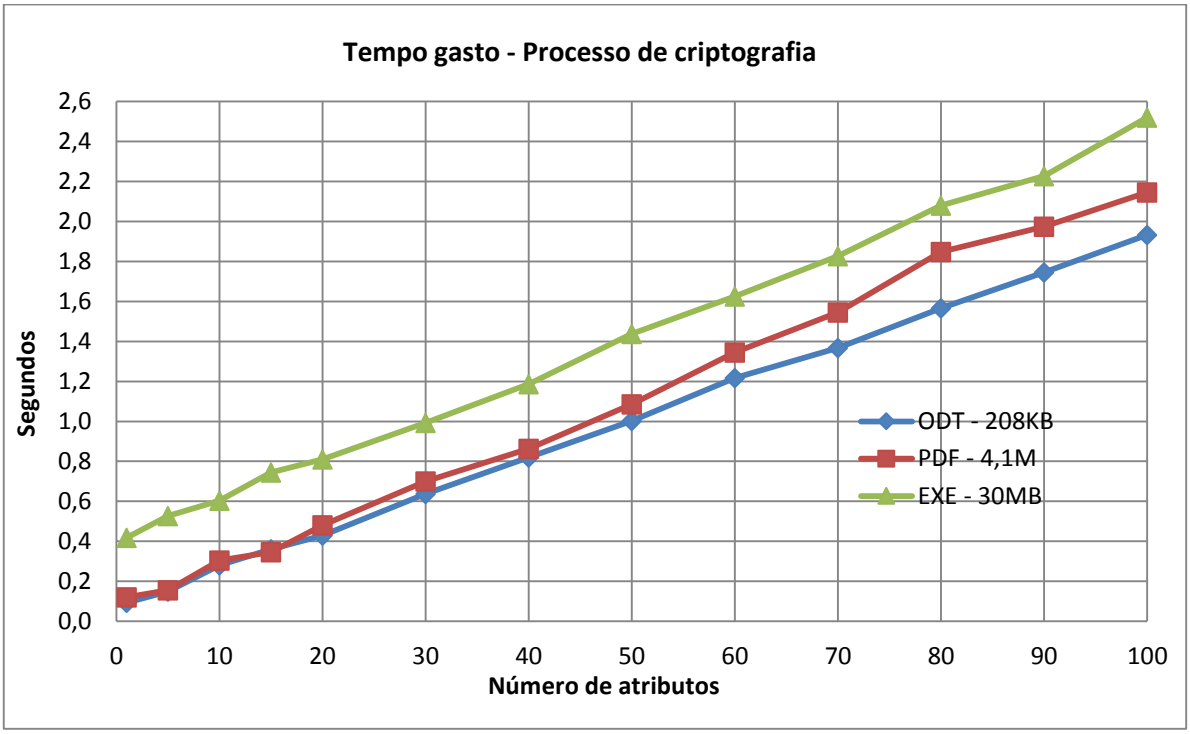

Figura 5.5 – Tempo gasto para realizar o processo de criptografia

Analisando o gráfico é possível observar que o processo de criptografia sofre influência em menor nível do tamanho do arquivo e em maior escala do número de atributos existentes na política de acesso.

Em média, a diferença entre o tempo gasto para criptografar o menor arquivo, que possui 208 KB, e o maior arquivo, que possui 30MB, foi inferior a 0,5 segundo, mesmo o maior arquivo tendo cerca de 150 vezes o tamanho do menor arquivo.

Para uma política de acesso contendo até 30 atributos, que representa um número substancial de atributos na árvore da estrutura de acesso, o tempo gasto foi menor que 1 segundo para todos os arquivos. Para o maior arquivo, contendo uma política de acesso bastante extensa com 100 atributos, ficou em torno de 2,5 segundos.

Na comparação com os testes realizados com o algoritmo RSA, a criptografia utilizando a chave de 2048 bits demorou em média valores inferiores que o algoritmo adotado pelo mecanismo, obtendo em média 10 milissegundos para o arquivo de 208KB, 160 milissegundos para o arquivo de 4,1MB e 1,14 segundos para o arquivo de 30MB.

Quando em posse do conteúdo e da estrutura de acesso, a aplicação pode realizar a descriptografia do conteúdo. Este processo, pode ser realizado no *host* do usuário requisitante.

O resultado dos testes em relação ao processo de descriptografia são apresentados na Figura 5.6.

Como é possível observar no gráfico apresentado, o processo de descriptografia é executado de maneira mais rápida em relação à criptografia do conteúdo apresentada anteriormente.

Este processo, assim como a criptografia, sofre uma pequena variação em relação ao tamanho do arquivo. Em todos os testes os valores ficaram abaixo de 700 milissegundos, pico de tempo que foi ocasionado na descriptografia do arquivo de 30MB. Para este mesmo arquivo, a média ficou em 600 milissegundos. Para o arquivo de 4,1MB havendo picos de 260 milissegundos e ficando em 20 milissegundos na média. Para o menor arquivo, de 208KB, a média foi de 150 milissegundos, atingindo picos de 230 milissegundos.

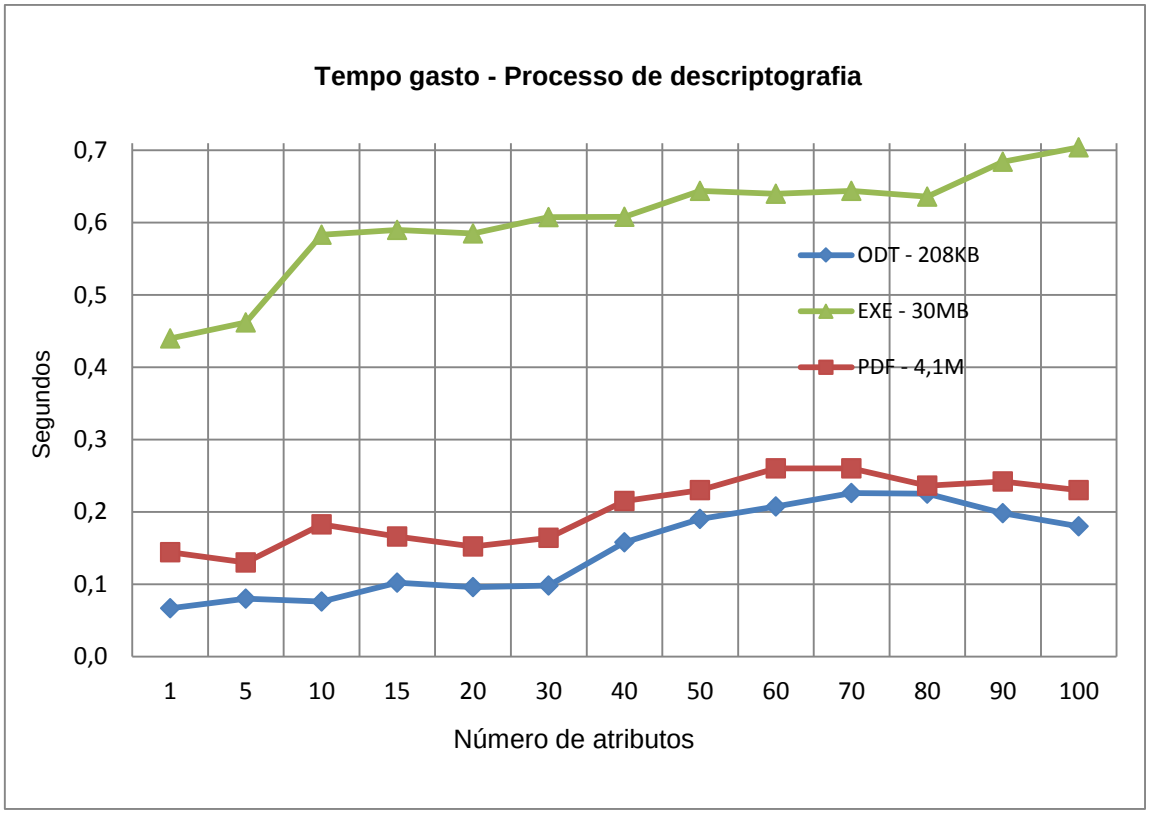

Figura 5.6 – Tempo gasto para realizar o processo de descriptografia

Em comparação aos testes realizados com o algoritmo RSA, foram obtidos em média 10 milissegundos para o arquivo contendo 208KB, 990 milissegundos para o arquivo contendo 4,1MB e 1,16 segundo para o arquivo que possui 30 MB de conteúdo.

O processo de conversão é realizado sobre a estrutura de acesso pelo servidor *proxy*. Este processo é o que gera um *overhead* adicional em relação a outros sistemas de criptografia que não utilizem re-encriptação por *proxy*, portanto, neste caso não cabe comparação em relação ao algoritmo RSA.

Como esperado, este processo não possui influência em relação ao tamanho do arquivo de conteúdo, uma vez que é aplicado somente sobre a estrutura de acesso. Deste modo, sofre influência do número de atributos utilizado pela política de acesso ao conteúdo.

Na Figura 5.7Figura 5.7 os resultados dos testes aplicados ao processo de conversão do *proxy* são apresentados.

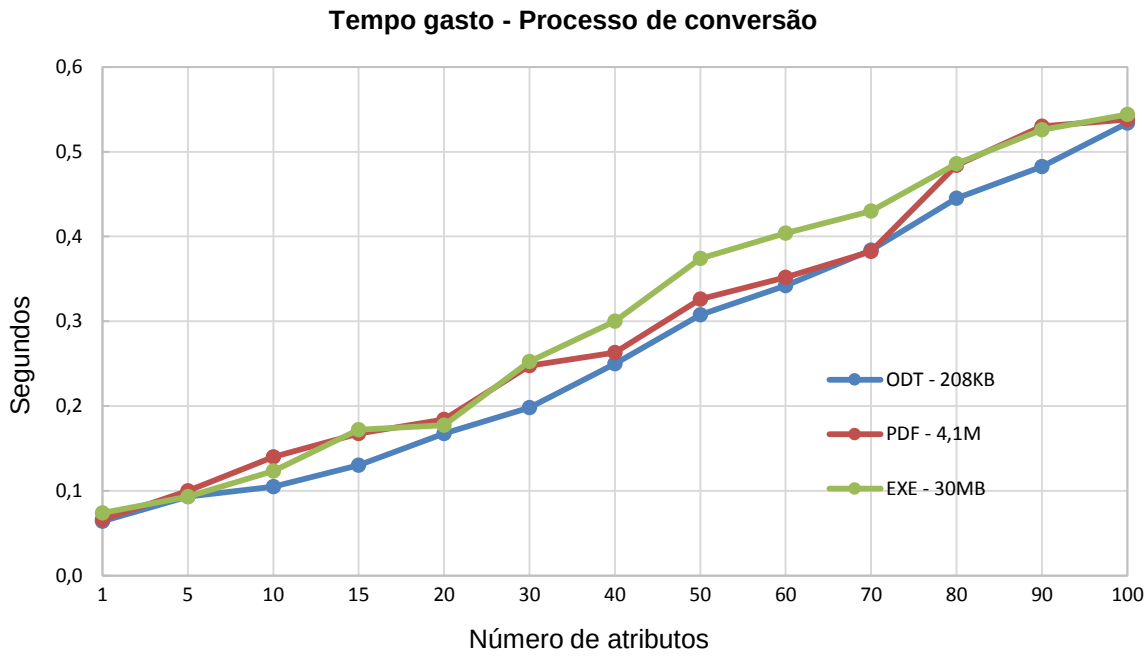

Figura 5.7 – Tempo gasto para realizar o processo de conversão

Como é possível identificar no gráfico apresentado na Figura 5.7, nestes testes a variação do tempo gasto entre as estruturas dos diferentes arquivos é pequena. Entretanto, há uma variação considerável em relação ao número de atributos existente na política.

Em estruturas de acesso contendo até 5 atributos o processo de conversão pelo *proxy* foi realizado dentro de 100 milissegundos em todas as situações. A marca de 20 milissegundos só foi atingida quando a estrutura de acesso utilizava mais de 20 atributos. Com a estrutura de acesso contendo 100 atributos o tempo gasto foi de 540 milissegundos.

5.2.3 Consumo de memória

Assim como os testes em relação ao tempo de processamento apresentados anteriormente, os testes de consumo de memória visam avaliar a quantidade de memória RAM utilizada para realizar as principais funções do mecanismo proposto, com o objetivo de identificar sua viabilidade.

Os resultados obtidos nos testes de consumo de memória pelo processo de criptografia são apresentados na Figura 5.8Figura 5.8.

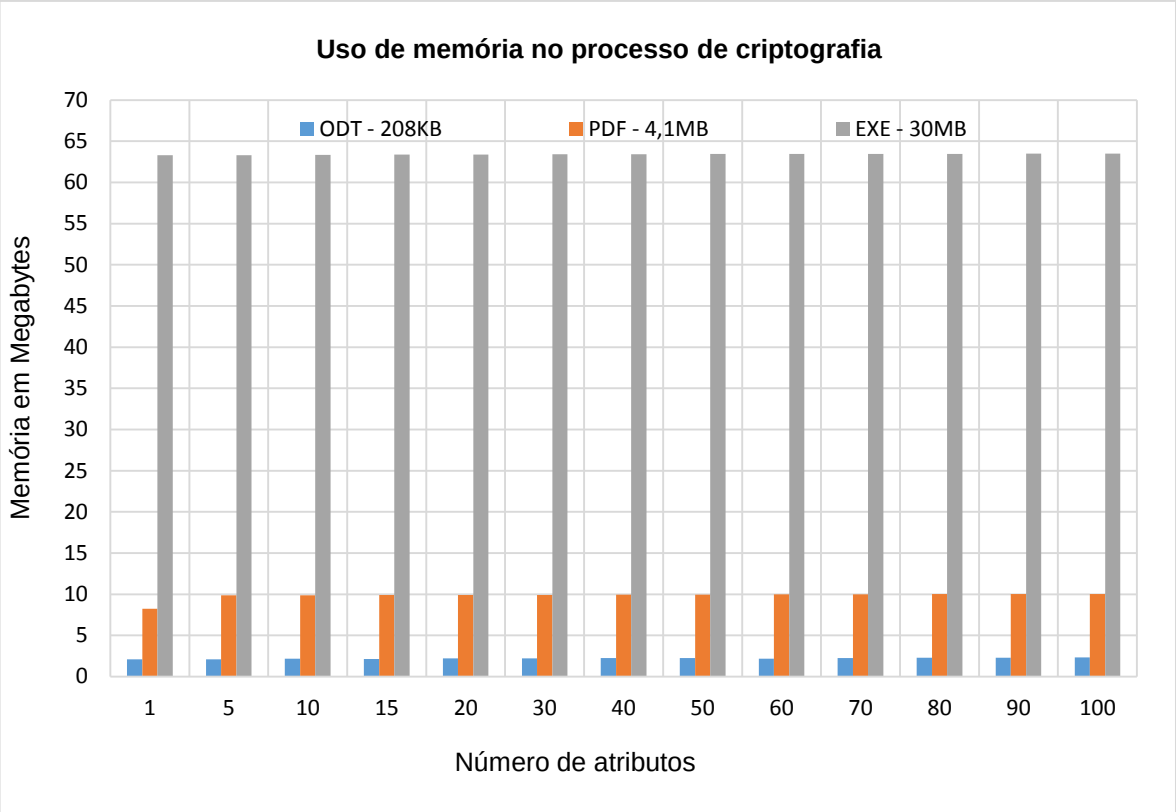

Figura 5.8 – Consumo de memória durante o processo de criptografia

Como é possível identificar no gráfico acima, o consumo de memória está diretamente relacionado ao tamanho do arquivo, tendo a variação entre o número de atributos da política de acesso uma influência menor sobre o resultado dos testes realizados.

Para o menor arquivo, de 208 KB, foi necessário entre 2,08 MB para uma estrutura com apenas 1 atributo e 2,31 MB para a política com 100 atributos, tendo uma média de 2,21 MB. Para o arquivo de 4MB, o consumo de memória RAM variou entre 8,25 MB para a política contendo apenas 1 atributo e 10,02 MB para a política com 100 atributos, obtendo 9,81 MB de média. O maior arquivo utilizado no teste, contendo 30 MB de conteúdo, consumiu em média 63,41 MB durante o processo de criptografia, tendo um mínimo de 63,30 MB na menor política de 1 atributo, e 63,51 MB na política de 100 atributos.

Considerando que para criptografar os dois arquivos maiores foi necessário, em média, utilizar um pouco mais que o dobro de memória em relação ao tamanho do arquivo, pode ser necessário que arquivos muito grandes sejam divididos em partes

menores caso o *host* do usuário publicador possua limitação para o consumo de memória.

Em relação aos testes realizados com o algoritmo RSA, o consumo de memória utilizado pelo na criptografia do mecanismo proposto foi ligeiramente menor. No processo de criptografia do algoritmo RSA foram obtidos de 3,55 MB de consumo de memória RAM para o arquivo de 208 KB, 11,55 MB utilizados para cifrar o arquivo de 4,1 MB e 65,09 MB para o arquivo de 30 MB.

O processo de descriptografia, realizado pelo *host* do requisitante do conteúdo possui um consumo de memória bem próximo ao apresentado pelo processo de criptografia. Na Figura 5.9 estão os resultados dos testes realizados.

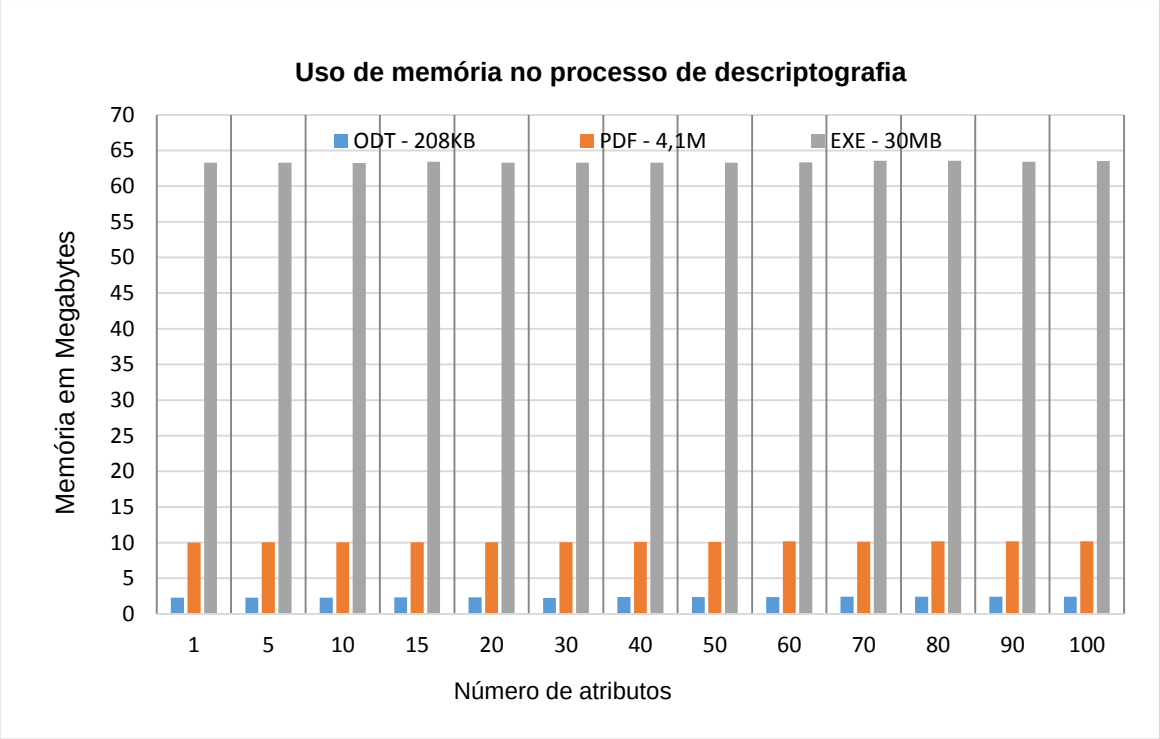

Figura 5.9 – Consumo de memória durante o processo de descriptografia

Analisando o gráfico da Figura 5.9 é possível observar que o número de atributos utilizado na política de acesso possui apenas uma pequena influência em relação à quantidade de memória RAM utilizada.

Para o menor arquivo, o consumo variou de 2,08 MB para a política contendo 1 único atributo até 2,31 MB para a estrutura com o número de 100 atributos, atingindo uma média de 2,21 MB de consumo de memória. Para o arquivo de 4,1 MB, foram necessários 8,25 MB para a menor estrutura de acesso utilizada, até 10,02 MB para

a maior estrutura, tendo uma média de 9,81 MB. Para o maior arquivo, contendo 30 MB, o consumo variou entre 63,31 MB para a estrutura de acesso contendo apenas 1 atributo, até 63,50 MB na estrutura com 100 atributos, possuindo uma média de 63,41 MB.

Em relação aos testes realizados com o algoritmo RSA, os resultados obtidos foram: 3,46MB para o arquivo de 208 KB, 12,60 MB gastos com a decifragem do arquivo de 4,1MB e 73,17 MB dispendidos na decifragem do arquivo de 30 MB. Deste modo, é possível notar que em média, o consumo de memória do algoritmo CP-ABE adotado pelo mecanismo apresentou menor consumo de memória RAM.

O processo de conversão é realizado pelo *proxy* e seu consumo memória ocorre de acordo com o número de atributos utilizado na política. Os resultados dos testes realizados são demonstrados na Figura 5.10.

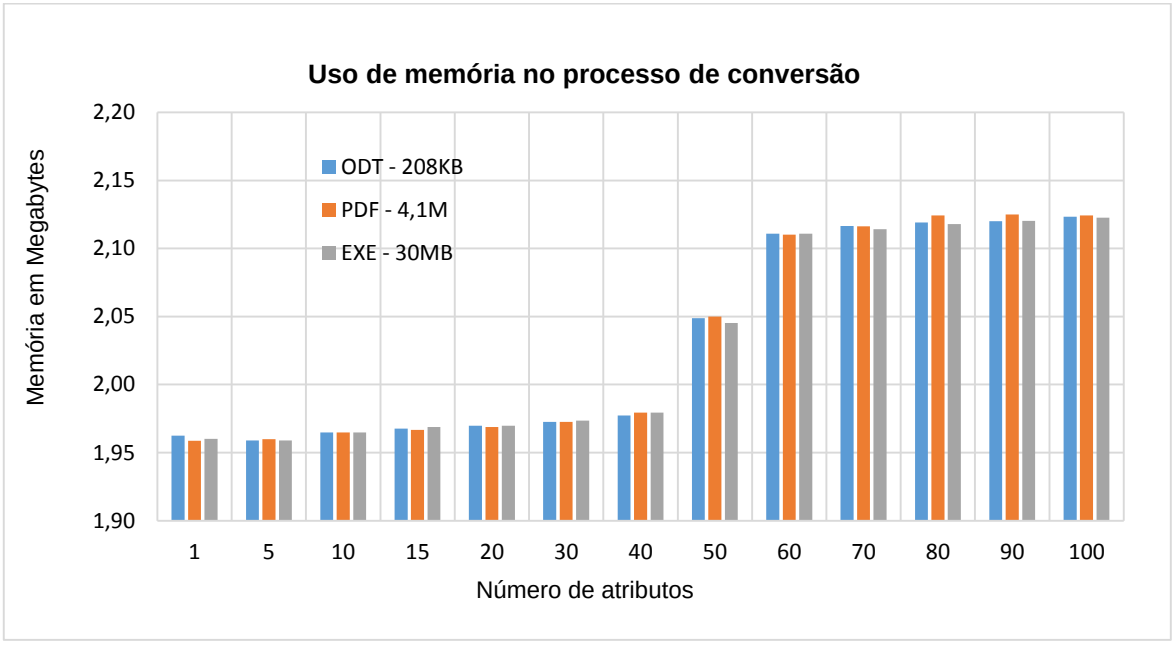

Figura 5.10 – Consumo de memória durante o processo de conversão

Pelos resultados apresentados neste gráfico da Figura 5.10 é possível notar que a variação na utilização de memória RAM pelo processo de conversão praticamente não sofre influência do tamanho do arquivo, uma vez que é aplicada somente sobre a estrutura.

As conversões das estruturas de acesso contendo entre 1 e 40 atributos consumiram menos que 2 MB de memória RAM cada, só ultrapassando esta marca

no teste com 50 atributos. O pico de consumo foi atingido ao converter a estrutura contendo 100 atributos, quando houve um consumo de 2,12 MB de RAM.

5.2.4 Testes complementares

Com o objetivo de avaliar melhor o desempenho em relação ao *proxy* e também analisar o impacto de arquivos maiores na execução do processo de conversão, foram realizados testes complementares.

Nestes testes uma quantidade maior de arquivos foi avaliada, variando entre 1MB e 250MB. Para compor a política de acesso, foi utilizada uma estrutura de acesso contendo 5 atributos diferentes combinados (SILVA E ZORZO, 2016).

O resultado deste teste em relação à criptografia e descriptografia é apresentado na Figura 5.11.

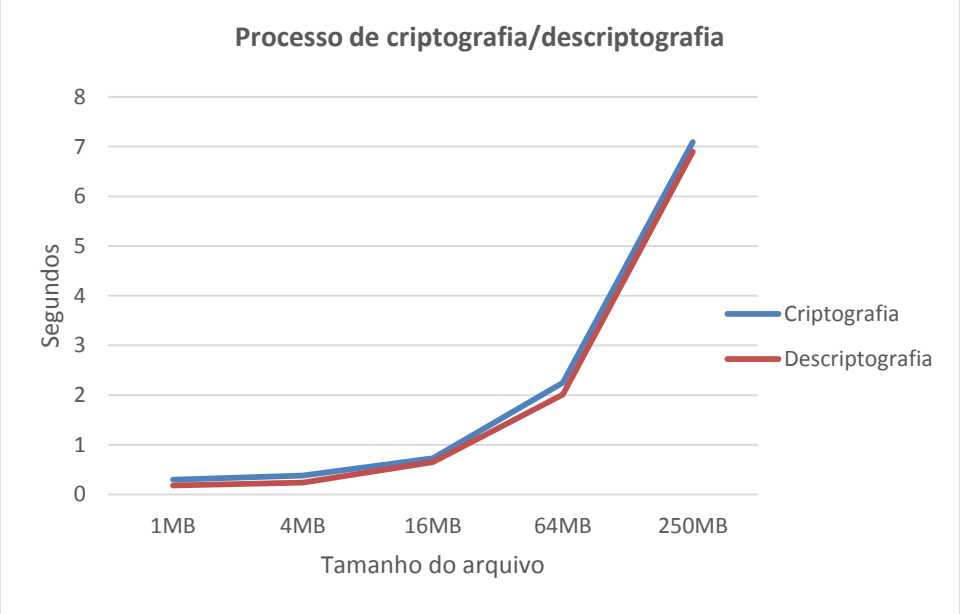

Figura 5.11 – Tempo gasto para realizar criptografia e descriptografia

Nestes testes o tempo gasto variou entre 300 milissegundos para realizar a criptografia e cerca de 200 milissegundos para realizar a descriptografia do arquivo de 1MB, atingindo 7,09 segundos para a criptografia e 6,9 segundos para a descriptografia do arquivo de 250 MB.

Considerando que estes processos podem ser realizados no *host* dos usuários, os resultados são considerados como aceitáveis.

Analisando o processo de conversão, objeto alvo destes testes por representar o *overhead* adicional em relação a outros esquemas de criptografia, foram obtidos os resultados demonstrados na Figura 5.12 para os mesmos 5 arquivos anteriormente descritos.

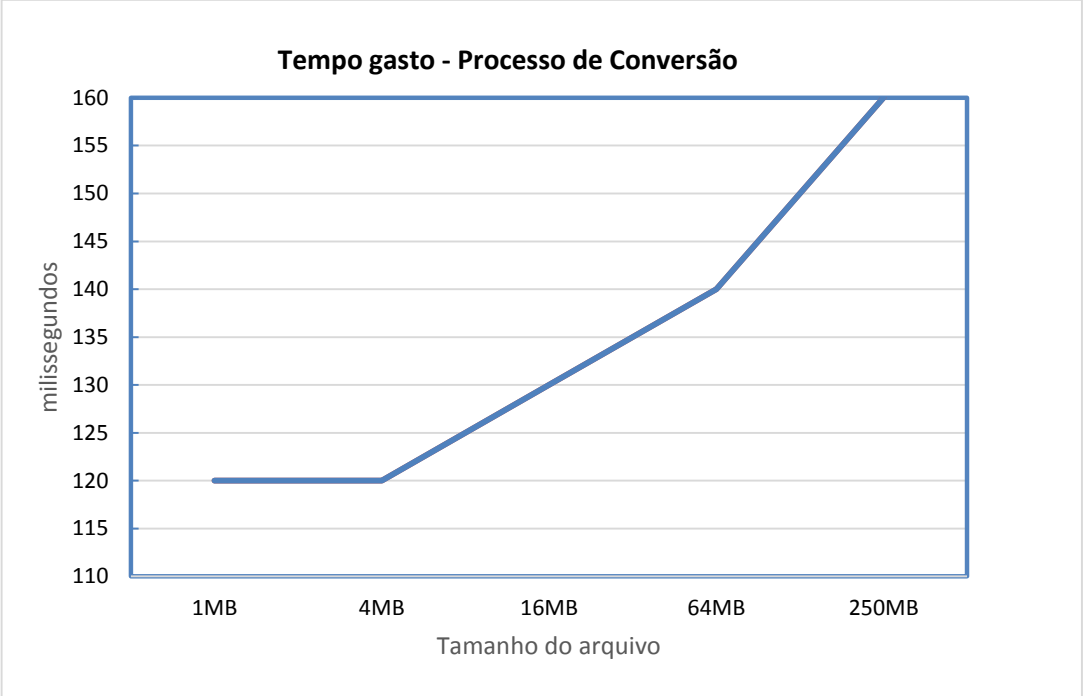

Figura 5.12 – Tempo gasto para realizar o processo de conversão

Como é possível notar no gráfico apresentado, para arquivos menores que 4 MB a conversão foi realizada em cerca de 120 milissegundos e atingiu um pico de 160 milissegundos para converter o arquivo de 250 MB.

Considerando uma média de 150 milissegundos seria possível converter cerca de 428 arquivos em um segundo. Isso utilizando um único núcleo de processamento e 1GB de memória RAM, que nos testes estavam compartilhados com a execução do sistema operacional.

Já em relação ao consumo de memória o processo de conversão utilizou menos de 2 MB para converter cada um dos 5 arquivos, variando entre 1,969 MB para o arquivo de 1 MB e 1,973 MB para o arquivo de 250 MB. Estes resultados são apresentados na Figura 5.13.

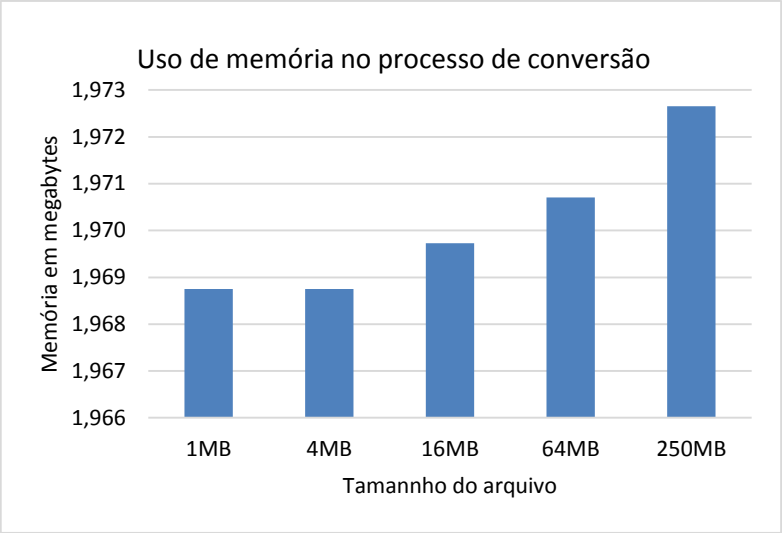

Figura 5.13– Consumo de memória durante o processo de conversão

5.3 Considerações finais

O mecanismo foi submetido a testes de desempenho que comprovam sua viabilidade de operação para uma grande quantidade de aplicações em termos de tempo de processamento, consumo de memória e tamanho dos arquivos criptografados.

Este resultado positivo foi encontrado até mesmo em condições nas quais os equipamentos contam com pouca capacidade de processamento e memória, como nos testes realizados.

Para fins de comparação foram realizados testes com o algoritmo RSA no mesmo hardware em que o mecanismo foi avaliado, de modo a obter um valor de referência comparativo.

Para um fluxo ainda maior de transferência de arquivos, uma infraestrutura de maior capacidade pode ser adotada, com equipamentos providos de múltiplos processadores, o que permitiria o processamento de arquivos em paralelo.

Capítulo 6

CONCLUSÃO E TRABALHOS FUTUROS

Este capítulo apresenta as conclusões deste trabalho enfatizando as contribuições e os trabalhos futuros.

6.1 Conclusão

As arquiteturas ROC têm sido propostas com a intenção de resolver problemas relacionados à segurança, mobilidade e manipulação de grandes volumes de dados existentes na Internet atual, especialmente considerando desempenho na transmissão de grandes quantidades de dados, mobilidade e problemas de segurança.

Os mecanismos de segurança adotados pelas principais arquiteturas ROC mitigam alguns problemas de segurança. Entretanto, quando se trata de dados sensíveis é necessário adotar recursos adicionais de segurança, para garantir a confidencialidade dos conteúdos e a privacidade dos usuários em relação a garantir quem pode acessar seus arquivos.

O mecanismo de garantia de privacidade proposto permite que o usuário compartilhe conteúdos utilizando uma ROC de modo que seja capaz de definir quais usuários vão poder acessar o conteúdo posteriormente.

A restrição de acesso é realizada através de uma política de acesso com alto nível de detalhamento e flexível para atender quaisquer tipos de rótulos desejados, podendo combinar atributos e operadores para determinar quem irá acessar o

conteúdo. O conteúdo será criptografado e só poderá ser acessado por um usuário que atenda a política determinada no momento da publicação.

Além disso, permite que o acesso a um conteúdo seja revogado para um usuário autorizado previamente sem que seja necessário republicar o conteúdo ou redistribuir chaves para os usuários.

Para garantir o controle do acesso, o mecanismo insere um *proxy* para intermediar o acesso do usuário ao conteúdo. Este *proxy* é parcialmente confiável, pois é incapaz de acessar o conteúdo utilizando sua própria chave. Entretanto, ele pode registrar os acessos a cada conteúdo uma vez que recebe a identificação do usuário requisitante. Isso permite estender o mecanismo para incluir a função de auditoria.

O mecanismo proposto se adequa ao perfil de uma ROC, pois permite que o conteúdo seja mantido em *cache* próximo ao usuário, sendo que a estrutura de acesso só estará acessível por intermédio do *proxy*.

O recurso de auditoria do mecanismo pode ser aplicado em ambientes onde o acesso a um conteúdo seja obrigatoriamente registrado, o que poderia ser inviável em uma ROC com *cache* de arquivos.

O paradigma ponto-a-ponto de uma ROC difere do esquema baseado em localização do *host* em uso na Internet. A inserção de um *proxy* envolve a existência de um processo extra e pode prejudicar o desempenho se o volume de tráfego for muito grande. Nesta situação, uma infraestrutura com alta capacidade de processamento de arquivos deve ser adotada.

Entretanto, diante dos testes realizados, o mecanismo se demonstrou viável considerando o tamanho dos arquivos criptografados, o que envolve o armazenamento e transmissão, tempo de processamento e consumo de memória, especialmente para uma aplicação que utilize um número de atributos baixo em suas políticas.

Obviamente, a adoção do mecanismo implica em um *overhead* ao acesso dos conteúdos. Entretanto, essa é uma condição inerente a qualquer mecanismo de segurança que seja aplicado.

6.2 Limitações e trabalhos futuros

Este trabalho propõe um mecanismo que garante a confidencialidade dos dados e a negociação de privacidade do usuário em relação aos dados que publica, uma vez que lhe confere o direito de escolher quais usuários ou grupos possuem privilégio para suas publicações.

Entretanto, outros aspectos de privacidade e segurança, como o sigilo dos eventos e do conteúdo perante a aplicação, não são abordados neste trabalho. Isso se deve ao fato do mecanismo ter como premissa a possibilidade de se realizar auditoria, como por exemplo, em um caso de violação legal que exija uma investigação.

Portanto, há um pressuposto de que os usuários confiam na aplicação e que não pretendem utilizá-la de maneira anônima, mas sim para que sejam compartilhados conteúdos restringindo o acesso de maneira controlada.

O mecanismo não aborda a questão de republicação do conteúdo por um usuário autorizado em outros meios, o que se torna uma sugestão de trabalho futuro.

REFERÊNCIAS

ACS, G.; CONTI, M.; GASTI, P.; GHALI, C.; TSUDIK, G. Cache Privacy in Named-Data Networking. In: ***Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*.** IEEE, 2013. p. *41-51.*

AFANASYEV, A.; MOISEENKO, I.; ZHANG , Z. ***ndnSIM: NDN simulator for NS-3***. Named Data Networking (NDN) Project, Tech. Rep. NDN-0005, Rev, v. 2, 2012.

AHLGREN, B.; DANNEWITZ, C.; IMBRENDA, C.; KUTSCHER, D.; OHLMAN, B. A Survey of Information-Centric Networking. ***IEEE Communications Magazine***, July 2012. Pag 26-36.

ARIANFAR, S.; KOPONEN, T. RAGHAVAN, B.; SHENKER, S. On Preserving Privacy in Content-Oriented Networks. In: ***Proceedings of the ACM SIGCOMM workshop on Information-centric networking ICN'11***. ACM Tenth International Conference on, 2011. P. 19-24.

BETHENCOURT, J.; SAHAI, A; WATERS, B. Ciphertext-policy attribute-based encryption. In: ***Security and Privacy, 2007. SP'07. IEEE Symposium on.*** IEEE, 2007. p. 321-334.

BRITO, G. M.; VELLOSO, P. B; MORAES, I. M. Redes Orientadas a Conteúdo: Um Novo Paradigma para a Internet. ***Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC***, v. 2012, p. 211-264, 2012.

CHAABANE, A.; CRISTOFARO, E., KAAFAR, M.A., UZUN, E. Privacy in content-oriented networking: Threats and countermeasures. In: ***ACM SIGCOMM Computer Communication Review***, v. 43, n. 3, p. 25-33, 2013.

CHEN, D.; ZHAO, H. Data security and privacy protection issues in cloud computing. In: ***Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on.*** IEEE, 2012. p. 647-651.

CISCO (2012). ***Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update-2013-2018***. White Paper. Disponível em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf>. Acesso em 02/03/2014.

DIBENEDETTO, S.; GASTI, P; TSUDIK, G.; UZUN, E. ***ANDaNA: Anonymous named data networking application****.* arXiv preprint arXiv:1112.2205, 2011.

GOYAL, V.; PANDEY, O.; SAHAI, A.; WATERS, B. Attribute-based encryption for fine-grained access control of encrypted data. In**: *Proceedings of the 13th ACM***

***conference on Computer and communications security*, CCS '06**, pages 89–98, New York, NY,USA, 2006.

GHODSI, A.; KOPONEN, T.; RAJAHALME, J.; SAROLAHTI, P.; SHENKER, S. Naming in content-oriented architectures. In: ***Proceedings of the ACM SIGCOMM workshop on Information-centric networking.*** ACM, 2011. p. 1-6.

HAMDANE, B.; SERHROUCHNI, A.; FADLALLAH, A.; EL FATMI, S. G. Named-Data Security Scheme for Named Data Networking. In: ***Proceedings of the third International Conference on the Network of the Future (NoF 2012)***. Tunis, Tunisia. 2012.

HUGHES, E. A Cypherpunk's manifesto. In: ***The electronic privacy papers***. John Wiley & Sons, Inc., 1997. p. 285-287.

ION, M; RUSSELO, G.; CRISPO, B. Design and implementation of a confidentiality and access control solution for publish/subscribe systems. ***Computer Networks***, v. 56, n. 7, p. 2014-2037, 2012.

ION, M.; ZHANG, J.; SCHOOLER, E. M. Toward content-centric privacy in ICN: attribute-based encryption and routing. In: ***Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM***. ACM SIGCOMM'13. Hong Kong, China: 2013. p. 513-514.

JACOBSON, V.; SMETTERS, D. K.; BRIGGS, N. H.; THORNTON , J. D.; MICHAEL, F. P. ; BRAYNARD, R. L. Networking named content. In*:* ***Proceedings of the 5th international conference on Emerging networking experiments and technologies***. ACM CoNEXT'09. Rome, Italy: 2009. p. 1-12.

JAHID, S.; BORISOV, N. *PIRATTE:* ***Proxy-based Immediate Revocation of ATTribute-based Encryption***. Tech Report. arXiv preprint arXiv:1208.4877, 2012

KUROSE, J. F.; ROSS, K. W. ***Redes de Computadores e a Internet: Uma abordagem Top Down***. 5 ed. São Paulo: Pearson, 2010.

KIM, E.; KIM, D.; HUH, M.; LEE, B. J.. Privacy protected content sharing in extended home environment over Content-Centric Networking. In: ***Consumer Electronics (ICCE)***, 2012 IEEE International Conference on. IEEE, 2012. p. 648-649.

LAUINGER, T.; LAOUTARIS, N.; RODRIGUEZ, P.; STRUFE, T.; BIERSACK, E.; KIRDA, E. Privacy risks in named data networking: what is the cost of performance?. ***ACM SIGCOMM Computer Communication Review***, Vol. 42, n. 5, p. 54-57, 2012.

MASSAWE, E. A.; DU, S.; ZHU, H. A Scalable and Privacy-Preserving Named Data Networking Architecture Based on Bloom Filters. In: ***Distributed Computing Systems Workshops (ICDCSW)***, 2013 IEEE 33rd International Conference on. IEEE, 2013. p. 22-26.

MOHAISEN, A; MEKKY, H.; ZHANG, X.; XIE, H.; KIM, Y. ***Timing attacks on access privacy in information centric networks and countermeasures.*** Dependable and Secure Computing, IEEE Transactions on, v. 12, n. 6, p. 675-687, 2015.

MOHAISEN, A.; ZHANG, X.; SCHUCHARD , M.; XIE, H.; KIM, Y. Protecting access privacy of cached contents in information centric networks. In*:* ***Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security***. ACM ASIA CCS '13, New York, NY, USA, 173-178.

MOISEENKO, I.; ZHANG, L.. ***Consumer-Producer API for Named Data Networking***. Technical Report NDN-0017, 2014. <Disponível em: http://named-data.net/techreports.html> Acesso em 18/02/2014.

NAOR, M.; PINKAS, B. Efficient trace and revoke schemes. In: ***Financial cryptography***. Springer Berlin/Heidelberg, 2001. p. 1-20.

O'REILLY, Tim. What is Web 2.0: Design patterns and business models for the next generation of software. ***Communications & strategies***, n. 65, 2007.

PAPANIS, J. P.; PAPAPANAGIOTOU, S. I.; MOUSAS, A. S.; LIOUDAKIS, G. V.; KAKLAMANI, D. I.; VENIERIS, I. S., On the use of Attribute-Based Encryption for multimedia content protection over Information-Centric Networks. ***Transactions on Emerging Telecommunications Technologies***. Wiley Online Library, 2013. doi: 10.1002/ett.2722

PASSARELA, A. A survey on content-centric technologies for the current Internet: CDN and P2P solutions. ***Journal of Computer Communications***, Volume 35, Issue 1, 1 January 2012, Pages 1-32, ISSN 0140-3664, Disponível em: <http://dx.doi.org/10.1016/j.comcom.2011.10.005> Acesso em 10/02/2014.

PAUL, S.; PAN, J.; JAIN, R. Architectures for the future networks and the next generation Internet: A survey. ***Computer Communications***, v. 34, n. 1, p. 2-42, 2011.

SAHAI, A.; WATERS, B. Fuzzy identity-based encryption. In: ***Ronald Cramer, editor, Advances in Cryptology* – EUROCRYPT 2005**, volume 3494 of Lecture Notes in Computer Science, p. 457–473. Springer Berlin Heidelberg, 2005.

SHAMIR, A. Identity-based cryptosystems and signature schemes. *In:* ***Advances in cryptology.*** Springer Berlin Heidelberg, 1985. p. 47-53.

SILVA, R. S.; ZORZO, S. D., "An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges," in: ***Consumer Communications and Networking Conference (CCNC)***, 2015 12th Annual IEEE, pp.128-133, 9-12 Jan. 2015.

SILVA, R. S.; ZORZO, S. D., "On the Use of Proxy Re-Encryption to Control Access to Sensitive Data on Information Centric Networking," in: ***International Conference on Information Networking (ICOIN)****,* 2016 30th Annual, 13-15 Jan. 2016.

STALLINGS, W. ***Criptografia e Segurança de Redes***. 4ª ed. São Paulo, SP, Brasil: Pearson, 2008. ISBN 9788576051190.

TOURANI, R; MICK, T; MISRA, S.; PANWAR, G. ***Security, Privacy, and Access Control in Information-Centric Networking: A Survey***. arXiv preprint arXiv:1603.03409, 2016.

WANG, L.; HOQUE, A. K. M. M.; YIY, C.; ALYYAN, A.; ZHANG, B. *OSPFN:* ***An OSPF based routing protocol for Named Data Networking***. University of Memphis and University of Arizona, Technical Report NDN-003, 2012. Disponível em <http://named-data.net/publications/techreports>. Acesso em 18/12/2013.

WETHERALL, DAVID J.; TANENBAUM, ANDREW S. ***Redes de Computadores*** - 5ª ed. São Paulo, SP, Brasil: Pearson: 2011. ISBN: 9788576059240.

ZHANG, L.; ESTRIN, D.; BURKE, J.; JACOBSON, V.; THORNTON, J. D.; SMETTERS, D. K.; ZHANG, B.; TSUDIK, G.; CLAFFY, K.; KRIOUKOV, D.; MASSEY, D.; PAPADOPOULOS, C.; ABDELZAHER, T.; WANG, L.; CROWLEY, P.; YEH, E. ***Named Data Networking (NDN) Project***. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 2010. Disponível em <https://www.parc.com/content/attachments/named-data-networking.pdf> Acesso em 04/01/2014.

ZHANG, L.; AFANASYEV, A. BURKE; JACOBSON, V.; CLAFFY, K.C.; CROWLEY, P.; PAPADOPOULOS, C.; WANG, L.; ZHANG, B. ***Named Data Networking (NDN) Project***. Relatório Técnico NDN-0019, Xerox Palo Alto Research Center-PARC, 2014. Disponível em <http://named-data.net/techreports.html> Acesso em 16/04/2014.

ZHANG, X.; CHANG, K.; XIONG, H.; WEN, Y.; SHI, G.; WANG, G. Towards name-based trust and security for content-centric network. In: ***Network Protocols*** *(ICNP),* 2011 19th IEEE International Conference on. IEEE, 2011. p. 1-6.