UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

RAFAEL HANSEN DA SILVA

Uma Abordagem Escalável para Controle de Acesso Muitos para Muitos em Redes Centradas em Informação

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Luciano Paschoal Gaspary

Porto Alegre
2016

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Hansen da Silva, Rafael

Uma Abordagem Escalável para Controle de Acesso Muitos para Muitos em Redes Centradas em Informação / Rafael Hansen da Silva. – Porto Alegre: PPGC da UFRGS, 2016.

67 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2016. Orientador: Luciano Paschoal Gaspary.

1. Redes centradas em informação. 2. Controle de acesso. 3. Publicação e recuperação muitos para muitos. 4. Criptografia baseada em atributos. I. Paschoal Gaspary, Luciano. II. Título.

“If I have seen farther than others,
it is because I stood on the shoulders of giants.”
— Sir Isaac Newton

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por ter me guiado durante toda a minha jornada e continuar me guiando.

Agradeço a minha família por todo o apoio e suporte para concluir mais uma etapa na minha vida. Meus pais, Achylles e Flávia, obrigado pelo carinho, compreensão, incentivo e, acima de tudo, pelos princípios que vocês me transmitiram. Imagino o quanto deve ser difícil compreender todos os momentos que dediquei aos estudos ao invés de estar com a família durante esse período de três anos. Agradeço, também, todos os meus familiares, que se preocuparam e, de alguma forma, estimularam-me a seguir em frente.

Agradeço, ainda, ao meu orientador professor Luciano Gaspary por ter me guiado com a sua experiência e paciência. Obrigado pela confiança depositada no meu trabalho, pelos conselhos e pelo exemplo. Agradeço, também, aos professores Marinho Barcellos, Lisandro Granville e Liane Tarouco pelos ensinamentos transmitidos durante o meu mestrado. Agradeço ao meu antigo orientador de bolsa de Iniciação Científica, professor Renato Ribas, o qual incentivou o meu interesse pela pesquisa científica.

Agradeço à universidade por todo o suporte necessário para realizar esta pesquisa e por minha formação na graduação. Em especial, à Beatriz, à Elisiane, à Valéria, à Silvania, à Eliane, à Sulamar, ao Luis Otávio e ao Leandro. Ademais, agradeço à CAPES e à CNPq pelo suporte financeiro durante o período de mestrado.

Agradeço a todos os amigos, sem exceção. Posso dizer que os integrantes do grupo de redes me ensinaram, cada qual a seu modo, o significado de parceria e amizade. Para aqueles que tive a oportunidade de conviver diariamente, como o Weverton (pelos debates ocasionados durante a escrita do artigo para o SBRC 2016), Rodolfo (por estar sempre à disposição), Marcelo L., Leonardo B., Guilherme, Glederson, Vinícius, Roberto, Jéferson, Carlos R., Rafael E., Walas, Júlio, Eduardo, Gabriel, Lucas M., Daniel, Rodrigo O., Rodrigo M., Matheus L., Pedro, Juliano, Lucas B., Miguel, Tobias e Bruna. Obrigado a todos pelas conversas e momentos de descontração. Por fim, agradeço a todos os amigos que não estão diretamente relacionados ao mundo da computação. Meu muito obrigado!

RESUMO

Um dos principais desafios em Redes Centradas em Informação (ICN) é como prover controle de acesso à publicação e recuperação de conteúdos. Apesar das potencialidades, as soluções existentes, geralmente, consideram um único usuário agindo como publicador. Ao lidar com múltiplos publicadores, elas podem levar a uma explosão combinatória de chaves criptográficas. As soluções projetadas visando a múltiplos publicadores, por sua vez, dependem de arquiteturas de redes específicas e/ou de mudanças nessas para operar. Nesta dissertação é proposta uma solução, apoiada em criptografia baseada em atributos, para controle de acesso a conteúdos. Nessa solução, o modelo de segurança é voltado a grupos de compartilhamento seguro, nos quais todos os usuários membros podem publicar e consumir conteúdos. Diferente de trabalhos anteriores, a solução proposta mantém o número de chaves proporcional ao de membros nos grupos e pode ser empregada em qualquer arquitetura ICN de forma gradual. A proposta é avaliada quanto ao custo de operação, à quantidade de chaves necessárias e à eficiência na disseminação de conteúdos. Em comparação às soluções existentes, ela oferece maior flexibilidade no controle de acesso, sem aumentar a complexidade do gerenciamento de chaves e sem causar sobrecustos significativos à rede.

Palavras-chave: Redes centradas em informação. Controle de acesso. Publicação e recuperação muitos para muitos. Criptografia baseada em atributos.

A Scalable Approach for Many-to-Many Access Control in Information-Centric Networks

ABSTRACT

One of the main challenges in Information-Centric Networking (ICN) is providing access control to content publication and retrieval. In spite of the potentialities, existing solutions often consider a single user acting as publisher. When dealing with multiple publishers, they may lead to a combinatorial explosion of cryptographic keys. Those solutions that focus on multiple publishers, on the other hand, rely on specific network architectures and/or changes to operate. In this dissertation, it is proposed a solution, supported by attribute-based encryption, for content access control. In this solution, the security model is focused on secure content distribution groups, in which any member user can publish to and retrieve from. Unlike previous work, the proposed solution keeps the number of cryptographic keys proportional to the number of group members, and may even be adopted gradually in any ICN architecture. The proposed solution is evaluated with respect to the overhead it imposes, number of required keys, and efficiency in the content dissemination. In contrast to existing solutions, it offers higher access control flexibility, without increasing key management process complexity and without causing significant network overhead.

Keywords: Information-centric networking. Access control. Many-to-many publication and retrival. Attribute-based encryption.

LISTA DE ABREVIATURAS E SIGLAS

ABE	*Attribute-Based Encryption*
AES	*Advanced Encryption Standard*
ACM	*Association for Computing Machinery*
CCN	*Content-Centric Networking*
CCN-AC	*Content-Centric Networking Access Control*
CP-ABE	*Ciphertext-Policy Attribute-Based Encryption*
DONA	*Data-Oriented Network Architecture*
DoS	*Denial-of-Service*
FIB	*Forwarding Information Base*
FIBE	*Fuzzy Identity-Based Encryption*
HABE	*Hierarchical Attribute-Based Encryption*
ICN	*Information-Centric Networking*
IP	*Internet Protocol*
KP-ABE	*Key-Policy Attribute-Based Encryption*
NetInf	*Network of Information*
NRS	*Name Resolution Service*
ONL	*Open Network Laboratory*
OSPF	*Open Shortest Path First*
OSPFN	*OSPF for Named-Data*
PARC	*Palo Alto Research Center*
PKI	*Public Key Infrastructure*
PURSUIT	*Publish-Subscribe Internet Technology*
QoE	Qualidade de Experiência
PIT	*Pending Interest Table*

RSA	*Rivest-Shamir-Adleman*
SA	Sistema Autônomo
SDSI	*Simple Distributed Security Infrastructure*
SPKI	*Simple Public Key Infrastructure*
SSH	*Secure Shell*
TCP	*Transmission Control Protocol*
UGC	*User-Generated Content*
UNIX	*Uniplexed Information Computing System*
URI	*Unique Resource Identifier*
URL	*Uniform Resource Locator*
VM	*Virtual Machine*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O paradigma de Redes Centradas em Informação (*Information-Centric Networking*, ICN) emergiu como uma direção promissora para a Internet do Futuro (XYLOMENOS et al., 2014). Apesar de suas potencialidades – por exemplo, de tornar a distribuição de conteúdos escalável e eficiente, e diminuir o tráfego no núcleo da Internet (AHLGREN et al., 2012) – há, ainda, diversos desafios que precisam ser abordados. Um dos mais importantes, e decisivo para o sucesso desse paradigma, está relacionado ao controle de acesso (BRITO; VELLOSO; MORAES, 2012; XYLOMENOS et al., 2014). Uma vez que os conteúdos passam a ser recuperados a partir de *caches* distribuídas na rede, os mecanismos de segurança precisam garantir que conteúdos publicados de forma protegida (isto é, com restrições de acesso) sejam consumidos apenas por usuários devidamente autorizados.

As soluções existentes, geralmente, focam em cenários em que grupos de compartilhamento seguro de conteúdo são formados por um publicador e vários consumidores (MISRA; TOURANI; MAJD, 2013; PAPANIS et al., 2014), sendo interessantes para uso por provedores como YouTube, Google Play, iTunes Store e NetFlix. No entanto, elas podem levar a um problema de explosão combinatória de chaves criptográficas, caso adotadas em cenários em que grupos formados por múltiplos publicadores e consumidores são a norma. As soluções focadas em múltiplos publicadores, por outro lado, introduzem entidades extras na rede para realizar reencriptação de conteúdos e/ou controle de acesso (FOTIOU; MARIAS; POLYZOS, 2012; SINGH et al., 2012). Embora efetivas, elas são intrusivas e pouco flexíveis para adoção de forma gradual, além de serem vulneráveis ao comportamento malicioso dessas entidades e, em alguns casos, dependentes de arquitetura.

Para lidar com esses problemas, nesta dissertação, propõe-se um modelo de segurança, apoiado por criptografia baseada em atributos, para controle de acesso a conteúdos em ICN. O modelo utiliza o conceito de participação em grupos de compartilhamento seguro, nos quais apenas usuários membros podem recuperar conteúdos. A publicação pode ser refinada pelo uso de atributos de usuários, de modo a restringir a recuperação a subconjuntos específicos de membros. O modelo proposto concilia suporte a múltiplos publicadores e controle de acesso agnóstico de arquitetura, mantendo o número de chaves proporcional ao de membros nos grupos e sem depender de entidades centrais. Imagina-se que a solução seja aplicável em um cenário em que grupos de usuários estão interessados

em trocar conteúdos (*User-Generated Content* ou UGC) formados por arquivos de diversas naturezas, especialmente vídeos (CISCO, 2015). A proposta é avaliada quanto ao suporte a múltiplos publicadores e ao custo de operação. Os resultados alcançados, por meio de avaliações em ambiente experimental controlado, confirmam a efetividade do modelo, o qual introduz um custo marginal para a publicação e a recuperação de conteúdos (em comparação às soluções existentes), ao passo que torna mais robusto e escalável o controle de acesso. De forma resumida, destaca-se como principais contribuições desta dissertação:

1. proposta (e respectiva modelagem) de emprego de criptografia baseada em atributos, mais especificamente CP-ABE (*Ciphertext-Policy Attribute-based Encryption*) (BETHENCOURT; SAHAI; WATERS, 2007), com o intuito de permitir a publicação/recuperação de conteúdos protegidos por múltiplos usuários;
2. implementação prototípica da solução proposta em um arcabouço de *software* real para uma arquitetura ICN, no caso CCN (*Content-Centric Networking*) (JACOBSON et al., 2012);
3. avaliação extensiva desses mecanismos comparando o seu desempenho a outros métodos de controle de acesso existentes.

O restante da dissertação está organizado como segue. O Capítulo 2 discute a fundamentação teórica e estado da arte. O Capítulo 3 descreve, em detalhes, a solução proposta para controle de acesso a conteúdos em ICN. Por sua vez, o Capítulo 4 apresenta a prototipação da solução proposta. O Capítulo 5 discute o ambiente experimental utilizado para avaliação da solução e os principais resultados alcançados. Por fim, o Capítulo 6 conclui a dissertação com considerações finais e perspectivas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E ESTADO DA ARTE

Neste capítulo são apresentados os principais conceitos relacionados a Redes Centradas em Informação e ao problema de publicação/recuperação de múltiplos conteúdos. Mais especificamente, a Seção 2.1 consiste em uma compilação de dois *surveys* (AHLGREN et al., 2012; XYLOMENOS et al., 2014) sobre ICN. Por sua vez, a Seção 2.2 introduz o conceito de criptografia baseada em atributos e aborda a técnica empregada pela solução proposta, CP-ABE. Por fim, expõe-se, na Seção 2.3, os trabalhos relacionados ao problema de controle de acesso em ICN.

2.1 Redes Centradas em Informação

Atualmente, a maior parte do tráfego da Internet é utilizada para distribuição de conteúdos. Esse fato tem levado a comunidade acadêmica e a indústria a repensarem, no próprio contexto de investigação da Internet do Futuro, arquiteturas de rede que venham a coexistir (ou substituir) a existente (reconhecidamente “ossificada"). Nesse contexto, tem-se investido em uma arquitetura de rede voltada para a distribuição eficiente de conteúdos na Internet, conhecida como Redes Centradas em Informação (*Information-Centric Networks* ou ICN) (AHLGREN et al., 2012; JACOBSON et al., 2012; XYLOMENOS et al., 2014). Essencialmente, essa arquitetura apresenta como principal característica o foco na informação e na sua distribuição, lançando mão de uma estratégia *publish-subscribe*.

A principal característica de ICN é centrar a rede no conteúdo disponível e na busca do conteúdo desejado pelo seu nome, independente de onde esse esteja localizado na rede. Para tal, arquiteturas ICN implementam diferentes mecanismos, como, por exemplo, *caching*, para reduzir o consumo de recursos de rede durante a distribuição de conteúdos. Nesta seção, discute-se o cenário atual sobre ICN e seus fundamentos. Primeiramente (Subseção 2.1.1), são revisados os mecanismos fundamentais comuns às arquiteturas ICNs propostas. Na sequência (Subseção 2.1.2), aborda-se brevemente a arquitetura CCN, por ser uma das mais representativas e ter sido empregada na prototipação da solução proposta.

2.1.1 Mecanismos Fundamentais

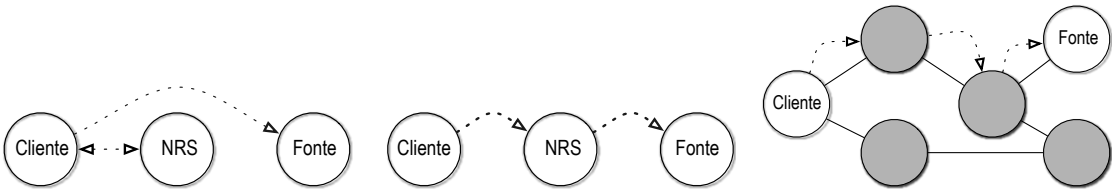
Como recém mencionado, ICN representa um paradigma no qual o foco da comunicação passa a ser o conteúdo ou a informação, ao invés de seu endereço ou localização na rede, como ocorre nas redes atuais. Nesse paradigma, aplicações precisam apenas determinar qual informação desejam recuperar. Encontrar a localização de um equipamento que a armazena passa a ser uma função da rede. Isso é viabilizado por meio de um conjunto de abstrações que interagem entre si, como explicado a seguir.

Publicadores podem disponibilizar conteúdos publicando-os. Um **conteúdo**[1] pode ser, por exemplo, um documento, livro ou vídeo. A comunicação é dirigidida pelos **requisitantes**, que solicitam conteúdos publicados na rede. A rede pode atender a requisição entregando dados provenientes de qualquer **fonte** que possua uma **cópia** do conteúdo (por exemplo, o próprio publicador ou uma unidade de *cache*). Para viabilizar esse desacoplamento em termos de tempo e espaço entre publicadores e requisitantes, são adicionados, a cada conteúdo, metadados que permitem verificar sua integridade e autenticidade. Essas abstrações são concretizadas por meio de três mecanismos fundamentais (AHLGREN et al., 2012; XYLOMENOS et al., 2014): nomeação de conteúdos, roteamento e *caching*.

A **nomeação de conteúdos** corresponde ao mecanismo responsável por associar um identificador à informação que se deseja publicar ou obter, a qual passa a ser representado por um conteúdo nomeado. O esquema de nomeação desses conteúdos desempenha um papel importante no conceito de ICN. O nome é dado à própria informação, de forma independente de sua localização. Ou seja, um conteúdo mantém seu nome desassociado do método como ele é transmitido ou armazenado. Existem dois esquemas principais de nomeação adotados por arquiteturas ICN: plano e hierárquico. A nomeação plana utiliza um conjunto de *bytes* de tamanho definido para identificar o conteúdo. Tal sequência pode ser gerada, por exemplo, por meio de uma função *hash* aplicada sobre o conteúdo. O esquema hierárquico, por sua vez, utiliza uma estrutura similar àquela das *Uniform Resource Locators* (URLs). Alguns esquemas de nomeação plana apresentam a vantagem de serem autocertificáveis, ou seja, o próprio nome serve de garantia de que o conteúdo representado equivale ao que se está procurando. Por sua vez, a nomeação hierárquica apresenta como vantagem a inteligibilidade e a possibilidade de agregação de conjuntos

[1] Nesta dissertação refere-se a conteúdo como uma peça indivisível que se materializa em uma rede ICN na forma de um único objeto. Um conteúdo se materializa na rede ICN na forma de um único arquivo. Objetos como vídeo que potencialmente precisariam ser materializados por meio de múltiplos arquivos, em uma rede ICN, por conta do seu tamanho, são tratados como um conteúdo distinto.

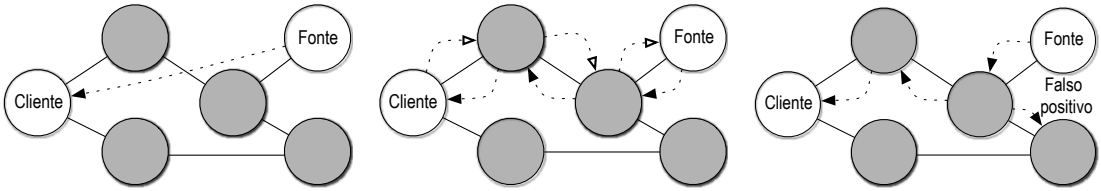
Figura 2.1: Modelos de roteamento de requisições.



(a) Requisição direta. (b) Requisição por serviço. (c) Requisição por nome.

Fonte: do autor (2016).

Figura 2.2: Modelos de encaminhamento de conteúdos.



(a) Conexão direta. (b) Caminho reverso. (c) Filtro *Bloom*.

Fonte: do autor (2016).

de nomes para fins de roteamento.

O segundo mecanismo fundamental de ICN, **roteamento**, divide-se em duas etapas. A primeira, chamada roteamento de requisições, refere-se ao caminho percorrido entre um requisitante e uma fonte que contenha uma cópia do conteúdo requisitado. A segunda etapa, encaminhamento de conteúdos, refere-se ao caminho de volta, ou seja, da fonte de uma cópia do conteúdo até o requisitante. As opções existentes para cada etapa são discutidas em maior nível de detalhes a seguir.

As soluções de roteamento de requisições podem ser classificadas em três modelos, os quais são ilustrados na Figura 2.1. A solução de roteamento de requisição inclui um *Name Resolution Service* (NRS) quando a nomeação é plana. O primeiro modelo utiliza um NRS para transformar o interesse do usuário (como um filme) em localizador (por exemplo, endereço IP) e em nome plano. O localizador e o nome são retornados para o cliente, que os utiliza para enviar a requisição diretamente à (uma) fonte do conteúdo. No segundo modelo, também, necessita-se de um NRS, mas o próprio serviço envia a requisição para a fonte do conteúdo. No terceiro modelo, por fim, o próprio nome do conteúdo é suficiente para o processo de roteamento, similarmente ao protocolo IP.

A segunda etapa de roteamento refere-se à entrega de conteúdos. Basicamente, as soluções podem ser classificadas em três modelos, ilustrados na Figura 2.2: conexão direta, caminho reverso e utilizando filtro *Bloom*. Algumas arquiteturas empregam conexão direta para viabilizar encaminhamento eficiente um para um, o que pode ser útil

em determinadas aplicações, em particular, nas que transmitem dados sensíveis. A segunda opção consiste em manter um rastro durante a etapa de roteamento das requisições e utilizá-lo para entregar o conteúdo por meio do caminho reverso. A terceira opção prevê o uso de filtros *Bloom* para encaminhar o conteúdo. Esse filtro diminui a sobrecarga com estruturas auxiliares, porém gera uma sobrecarga, na forma de mensagens transmitidas, em decorrência de falsos positivos.

O terceiro e último mecanismo fundamental em arquiteturas ICN refere-se ao uso de ***caching*** no núcleo da rede. Isto é, enquanto conteúdos são encaminhados de fontes a clientes, os dados podem ser armazenados por qualquer elemento intermediário. Caso uma nova requisição com a mesma informação seja realizada, pode-se respondê-la com uma cópia válida do conteúdo armazenado em um elemento de *caching*. O uso de *caching* representa um dos principais fatores que tornam arquiteturas ICN mais eficientes para a distribuição de conteúdo. Por sua relevância, o tema de *caching* no núcleo da rede é um dos mais discutidos em trabalhos recentes, os quais tratam de questões fundamentais como o posicionamento de cópias na rede (ELAYOUBI; ROBERTS, 2015) e métodos para rotear requisições para a cópia do conteúdo mais próximo do cliente (HEMMATI; GARCIA-LUNA-ACEVES, 2015).

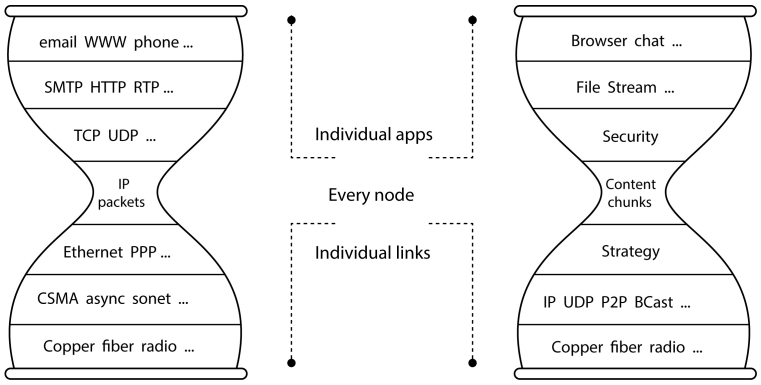
Nos últimos dez anos, diversas arquiteturas foram propostas e prototipadas com o intuito de concretizar o conceito de Redes Centradas em Informação. Entre elas, destacam-se CCN (JACOBSON et al., 2012), DONA (KOPONEN et al., 2007), NetInf (University of Paderborn, 2013) e PURSUIT (PURSUIT, 2009). Como a materialização da proposta deste trabalho foi feita sobre CCN, a seguir, descreve-se brevemente essa arquitetura.

2.1.2 Arquitetura CCN

A arquitetura CCN foi originalmente desenvolvida no âmbito do projeto CCNx (Palo Alto Research Center, 2015; JACOBSON et al., 2012). A arquitetura propõe-se a resolver os pontos negativos da Internet atual, projetada nas décadas de 60 e 70, e maximizar o desempenho de aplicações amplamente utilizadas, tais como de disseminação de dados. A arquitetura CCN propõe substituir a solicitação e execução de um comando a/em um servidor remoto pela requisição, à rede, de um conteúdo desejado. Realiza-se essa modificação sobre o argumento de que conteúdos nomeados correspondem a uma melhor abstração, no contexto atual, em comparação a identificação de servidores remotos.

A Figura 2.3 ilustra que apesar de CCN conservar as estruturas adotadas por TCP/IP, possui algumas diferenças cruciais. A primeira modificação consiste em substituir o protocolo IP pela requisição e transmissão de *chunks*, que equivalem a pedaços de conteúdos. A segunda característica fundamental de CCN consiste na possiblidade de ser executada sobre o protocolo IP, permitindo a sua adoção de modo incremental. Além dessas alterações, CCN introduz duas novas camadas relacionadas à segurança e à estratégia. A camada de segurança possibilita que conteúdos sejam recuperados de qualquer nodo da rede, com isso, garantindo a sua autenticidade e integridade. A camada de estratégia incrementa a eficiência da rede ao possibilitar que as requisições de conteúdos sejam propagadas por múltiplas interfaces e que sejam determinadas quais delas devem ser utilizadas analisando o contexto da rede.

Figura 2.3: Diferenças entre a pilha de protocolos TCP/IP e CCN.

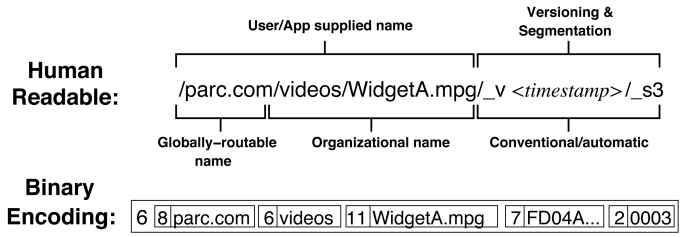


Fonte: (JACOBSON et al., 2012).

Mecanismos empregados. Nesse contexto, a arquitetura CCN tem como ideia geral que requisições sejam enviadas em direção à fonte em que determinado conteúdo foi publicado. Ao longo do caminho, caso uma cópia do conteúdo seja encontrada em *cache* ou se a requisição chegar à fonte, transfere-se o conteúdo pelo caminho inverso da requisição, desse modo, povoando as *caches* dos nodos no caminho com uma cópia do conteúdo. A seguir, são descritos os aspectos de nomeação de conteúdos, roteamento e *caching* propostos pela arquitetura.

O mecanismo de nomeação da arquitetura CCN lança mão de uma estrutura hierárquica e legível por humanos, similar àquela atualmente utilizada em *Uniform Resource Identifiers* (URIs). Embora os nomes sejam hierárquicos, não há uma semântica definida para eles. A Figura 2.4 ilustra um exemplo de como pode ser criado um nome hierárquico legível por humanos e de como a implementação da arquitetura CCN o codifica. O roteamento é feito utilizando *longest prefix matching* de forma a encaminhar requisições em

Figura 2.4: Exemplo de nome de conteúdo na arquitetura CCN.



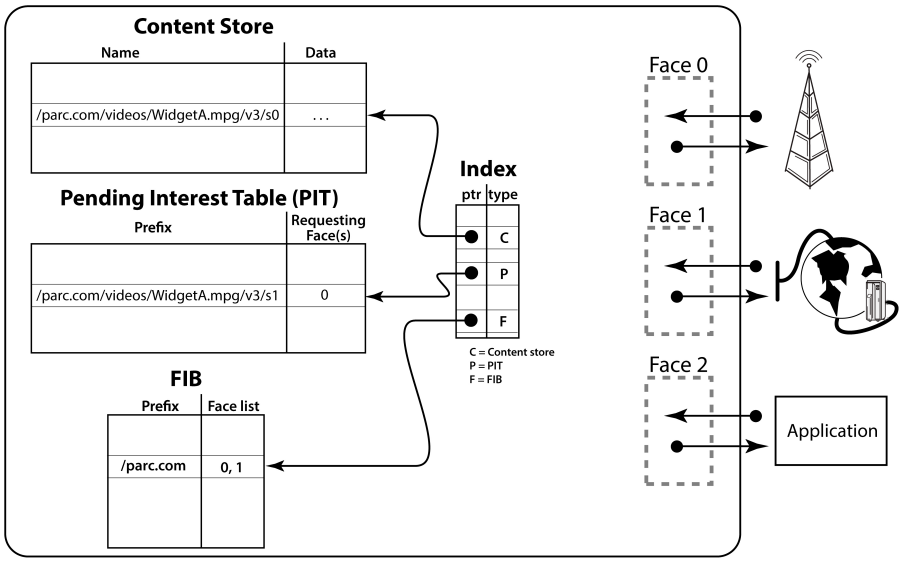
Fonte: (JACOBSON et al., 2012).

direção à fonte. Na implementação da arquitetura CCN, aplica-se uma extensão do *Open Shortest Path First* (OSPF) (COLTUN et al., 2008), denominado *OSPF for Named-Data* (OSPFN) (WANG et al., 2012), que utiliza as informações do OSPF para montar uma árvore de menores caminhos e uma estrutura auxiliar para mapear conteúdos à topologia gerada. Dessa forma, cada nodo sabe o melhor caminho para encaminhar requisições. O terceiro e último aspecto, *caching*, é realizado pelos nodos à medida que o conteúdo é transferido da fonte ao usuário que o requisitou. A abordagem mais simples corresponde a que todos os nodos armazenem uma cópia em sua *cache*. Outras alternativas mais sofisticadas podem ser usadas para coordenação de *cache*, tais como armazenar apenas uma cópia em cada sistema autônomo (SA) ou fazer *cache* probabilístico.

Uma vez elucidado como os mecanismos fundamentais empregados funcionam, é oportuno explicar o processo de encaminhamento de um conteúdo na arquitetura CCN. A Figura 2.5 ilustra os componentes que possibilitam a execução do encaminhamento de conteúdos por um roteador na arquitetura. Esse procedimento depende das estruturas *Content Store* (memória *cache*), *Forwarding Information Base* (FIB, tabela de roteamento) e *Pending Interest Table* (PIT, tabela de requisições pendentes) (JACOBSON et al., 2009). Quando uma solicitação por um conteúdo é recebida por um determinado roteador CCN, encaminha-se a requisição diretamente para a *Content Store*. A *Content Store* verifica em sua tabela se a *cache* do roteador possui o conteúdo para retorná-lo imediatamente. Caso não o possua, registra-se a solicitação na PIT. Isso propicia que, quando o roteador receber o conteúdo (vindo da fonte ou de outro nodo no caminho), o mesmo seja retornado pelo caminho inverso (de onde partiu a requisição). Por fim, a PIT consulta a FIB, com o objetivo de determinar para quais das interfaces do roteador (ou *faces*, como nomeado pela arquitetura) deve ser encaminhada a solicitação.

Estágio de desenvolvimento. O protótipo da arquitetura CCN, denominado CCNx teve seu desenvolvimento iniciado em 2008 (Palo Alto Research Center, 2015) e, atualmente, encontra-se na versão 1.0, disponibilizada em março de 2015. O projeto é liderado

Figura 2.5: Componentes de um roteador CCN.



Fonte: (JACOBSON et al., 2009).

pela PARC, que controla o desenvolvimento da versão oficial com a colaboração da comunidade por meio de comentários sobre *bugs* ou funcionalidades desejadas. CCNx possui documentação de seu código fonte e tutoriais sobre o funcionamento, permitindo a sua implantação facilmente em qualquer ambiente e que novas aplicações sejam desenvolvidas. Um dos seus usos mais significativos ocorre no *Open Network Laboratory* (ONL), um *testbed*, localizado nos Estados Unidos, apoiado pela *National Science Foundation*, que permite seu teste em maior escala e cenários mais realísticos. Além do protótipo funcional, um simulador, denominado ccnSIM (CHIOCCHETTI; ROSSI; ROSSINI, 2013), também, foi desenvolvido para auxiliar em potenciais pesquisas futuras. Especificamente sobre CCNx, existe um evento anual direcionado à comunidade que utiliza e pesquisa a plataforma, denominado CCNxCon (PARC, Inc., 2015). O CCNxCon reúne os principais trabalhos, sendo desenvolvidos no contexto, servindo como um fórum de discussão para futuras diretrizes e decisões de projeto. Vale ressaltar que no contexto deste trabalho, iniciado em 2013, utilizou-se a versão 0.8.2, pois, na sua versão mais recente, 1.0, existem vários elementos de implementação e mecanismos fundamentais que foram revisitados e reformulados.

Programas disponibilizados. A versão 0.8.2 do protótipo CCNx oferece algumas aplicações. Dentre elas, destacam-se: *ccnd*, *ccnr*, *ccnputfile*, *ccngetfile*, *ccndstatus*, *ccnputmeta* e *ccngetmeta*. *Ccnd* corresponde ao *software* responsável por encaminhamento e roteamento de interesses (roteador ICN). *Ccnr* permite que conteúdos publicados sejam armazenados em um repositório local. Além disso, responde a requisições feitas pela rede

a conteúdos que possui. *Ccnputfile* possibilita que seja inserido no repositório local um novo conteúdo associado a uma URI. *Ccngetfile* recupera um conteúdo disponibilizado na rede. *Ccndstatus* retorna um conjunto de informações sobre o roteador ICN em execução na máquina local. *Ccnputmeta* permite a associação de um arquivo de metadados (na forma de um outro objeto CCNx) a conteúdos previamente publicados. Finalmente, *ccngetmeta* possibilita a recuperação de um arquivo de metadados de um determinado conteúdo.

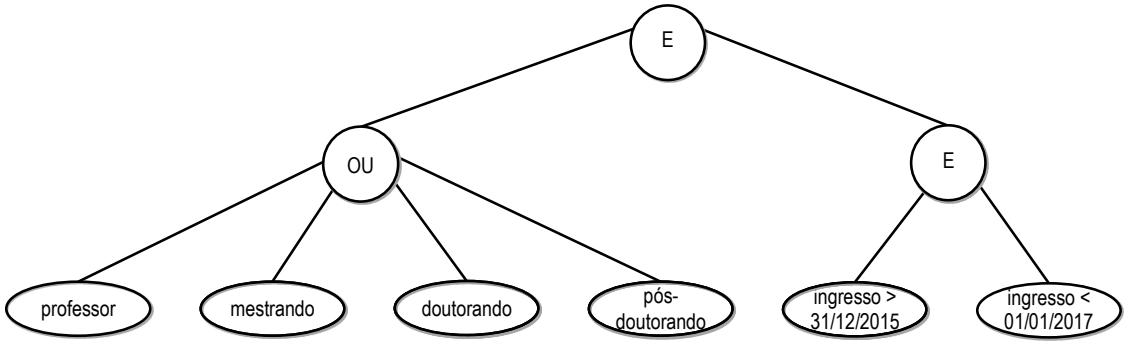
Após apresentar um dos componentes centrais da solução, que são as Redes Centradas em Informação, aborda-se o segundo conceito-chave empregado, denominado criptografia baseada em atributos.

2.2 Criptografia Baseada em Atributos

Tradicionalmente, restringe-se o acesso a determinados dados ou arquivos implementando algum mecanismo de controle de acesso em um servidor. Tal mecanismo, equivale a um controle que possibilita o acesso aos dados do servidor apenas se o requisitante possuir as permissões apropriadas para acessar os dados solicitados. Como explicado em Boneh *et al.* (BONEH; SAHAI; WATERS, 2012) e Bethencourt *et al.* (BETHENCOURT; SAHAI; WATERS, 2007), os serviços de armazenamento estão sendo cada vez mais utilizados com o intuito de aumentar a eficiência de acesso a dados e garantir uma maior disponibilidade por meio de replicação. A desvantagem introduzida por esse cenário corresponde à dificuldade em preservar a segurança usando métodos tradicionais de criptografia de chave pública. Isso ocorre, pois quando os dados estão armazenados em vários locais, a probabilidade de que um deles seja comprometido aumenta drasticamente. Por esses motivos, surgiu a necessidade que dados sensíveis sejam armazenados de um modo que se mantenham protegidos mesmo que um servidor seja comprometido.

A maioria dos métodos de criptografia de chave pública existentes possibilita que dados sejam criptografados apenas para um usuário em particular. Essa natureza de solução é eficaz para controle de acesso com poucos usuários, não sendo escalável no contexto em que um mesmo conjunto de dados deva ser acessado por muitos usuários. Para resolver esse problema, passou-se a focar em uma outra forma de realizar controle de acesso, empregada diretamente sobre conteúdos (e não mais sobre servidores). Nesse contexto, um novo tipo de criptografia de chave pública conhecido como Criptografia Baseada em Atributos (*Attribute-Based Encryption* ou ABE) ganhou destaque.

Figura 2.6: Representação de uma política na forma de uma árvore de acesso em ABE.



Fonte: do autor (2016).

A Criptografia Baseada em Atributos é vista como uma técnica com o potencial de solucionar o problema de armazenamento de dados em várias localidades e compartilhamento de dados para muitos usuários. Introduzida por Sahai e Waters no artigo Fuzzy Identity-Based Encryption (FIBE) (SAHAI; WATERS, 2005) e aprimorada por Goyal *et al.* (GOYAL et al., 2006), ABE consiste em uma criptografia de chave pública que permite o ciframento de um conjunto de dados (ou arquivos) para muitos usuários ao invés de apenas um. Com o intuito de permitir a funcionalidade proposta, a Criptografia Baseada em Atributos introduz dois novos conceitos. O primeiro consiste em descrever e identificar os usuários por um conjunto de atributos. O segundo conceito, denominado política de acesso, representa um conjunto de restrições que limita o acesso aos arquivos.

Atributos. Os atributos consistem em características relevantes que são associadas aos usuários, sendo eles definidos pelo administrador de um grupo. Um exemplo seria um conjunto de membros associados a uma universidade. Esses usuários poderiam ser definidos por um conjunto de atributos como “visitante”, “graduando”, “mestrando”, “doutorando”, “pós-doutorando”, “técnico”, “professor” e “ingresso”. Como demonstrado pelo exemplo e definido por Sahai e Waters (SAHAI; WATERS, 2005), os atributos são descritos por um conjunto de caracteres arbitrários.

Política de acesso. A política de acesso consiste em um conjunto de restrições que devem ser atendidas, com o intuito de descriptografar um determinado arquivo. Implementa-se a política na forma de uma árvore de acesso (vide Figura 2.6). Para compor uma política ou uma árvore de acesso, são empregados, normalmente, operadores relacionais ($>$, $<$ e $=$) e lógicos (**E** e **OU**). Os nodos compostos por operadores lógicos necessariamente possuem nodos filhos. Os nodos folhas obrigatoriamente correspondem a atributos. A política de acesso exibida pela figura corresponde à expressão (“professor ” OU “mestrando” OU “doutorando” OU “pós-doutorando”) E (ingresso $>$ 31/12/2015 E

ingresso $<$ 01/01/2017). A expressão restringe o acesso ao arquivo apenas a usuários associados a um dos atributos "professor", "mestrando", "doutorando" ou "pós-doutorando" e que tiveram o seu ingresso à universidade no ano de 2016. Usuários descritos como "visitante", "graduando" e/ou "técnico" não poderão acessar o arquivo mesmo pertencendo ao grupo, uma vez que não satisfazem a política empregada.

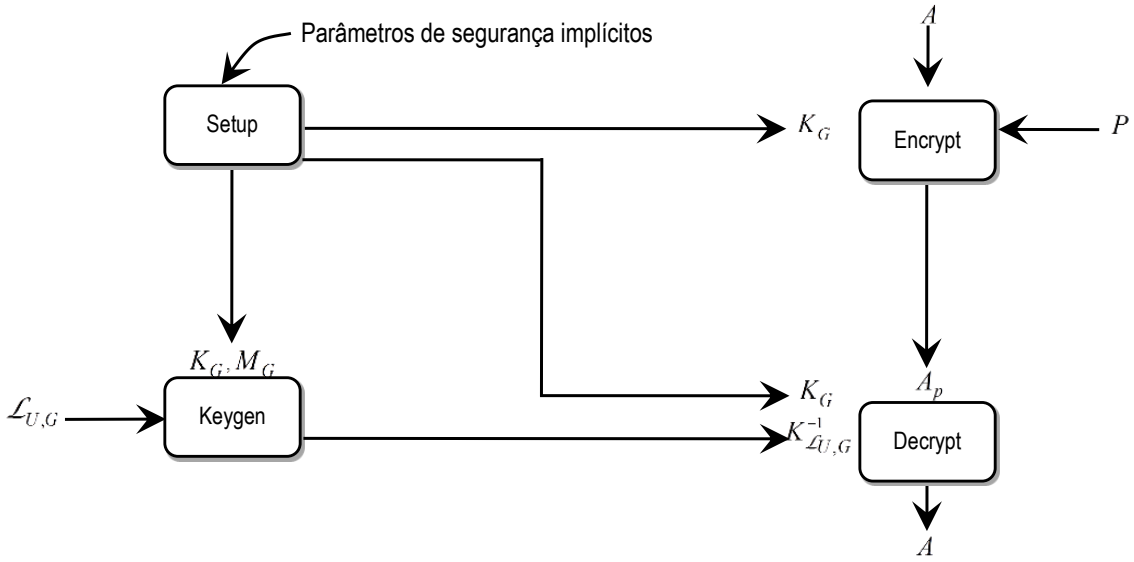
Além do trabalho original FIBE (SAHAI; WATERS, 2005), outras soluções foram propostas com o objetivo de melhorar os mecanismos de Criptografia Baseada em Atributos. Dentre elas, conforme afirmado por Qiao *et al.* (QIAO et al., 2014), destacam-se os trabalhos *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE) (BETHENCOURT; SAHAI; WATERS, 2007), *Attribute-based Encryption for Fine-grained Access Control of Encrypted Data* (KP-ABE) (GOYAL et al., 2006) e *Hierarchical Attribute-based Encryption for Fine-grained Access Control in Cloud Storage Services* (HABE) (WANG; LIU; WU, 2010). Dentre os trabalhos mencionados, foi escolhido CP-ABE para a materialização da solução proposta, pois permite que a política de acesso seja utilizada diretamente sobre arquivos e é mais simples quando comparada, por exemplo, com HABE que oferece um suporte muito mais rico em termos de hierarquia de atributos, dispensável no contexto deste trabalho. A técnica CP-ABE é descrita, com maiores detalhes, a seguir.

Ciphertext-Policy Attribute-Based Encryption. A criptografia CP-ABE, proposta, por Bethencourt *et al.* (BETHENCOURT; SAHAI; WATERS, 2007), associa a chave privada de um usuário a um conjunto de atributos. Quando determinado usuário criptografa um conteúdo empregando essa técnica, ele específica uma política sobre um dado conjunto de atributos e a emprega para gerar um conteúdo criptografado. Um usuário só será capaz de decifrá-lo se os atributos do mesmo atenderem à política de acesso definida.

CP-ABE tem similaridades ao trabalho de Goyal *et al.* (GOYAL et al., 2006). No entanto, embasa-se em um conjunto de técnicas diferentes. No CP-ABE, o usuário que criptografa o conteúdo possui a responsabilidade de montar uma política de acesso, de modo que somente os usuários habilitados possam descriptografá-lo. A seguir, são descritas as quatro operações principais oferecidas por CP-ABE. Também elas são ilustradas de forma resumida na figura Figura 2.7.

Operações fundamentais de CP-ABE. Inicializa-se o compartilhamento seguro de arquivos em um grupo G quando um administrador executa a operação de *Setup*. *Setup* corresponde à criação de um grupo, sendo esta inicializada por valores pseudoaleatórios (também conhecidos como parâmetros de segurança implícitos) fornecidos por uma biblioteca auxiliar. Concluída a operação, retorna-se um par de chaves pública (ou K_G,

Figura 2.7: Operações fundamentais do CP-ABE.



Fonte: (CODIO, S, 2011).

empregada para criptografia, descriptografia de arquivos e criação de chaves privadas) e mestra (ou M_G, responsável por auxiliar na criação de chaves privadas de usuários pertencentes ao grupo). Ressalta-se que a chave mestra M_G deve permanecer apenas com o administrador, enquanto a chave pública K_G deve ser concedida para todos que desejem criptografar um arquivo para o grupo. Em posse do par de chaves K_G e M_G, o administrador pode criar uma chave privada $K^{-1}_{\mathcal{L}_{U,G}}$ para um usuário U utilizando *Keygen*. A operação *Keygen* permite a geração de uma chave privada $K^{-1}_{\mathcal{L}_{U,G}}$ de um determinado usuário U a partir do par K_G, M_G e de um conjunto de atributos $\mathcal{L}_{U,G}$, que representa informações relevantes sobre um usuário U no grupo G. Ao receber a chave pública K_G, um determinado usuário está apto para realizar a operação *Encrypt*. *Encrypt* possibilita que um usuário criptografe um arquivo A usando a chave pública K_G e uma política P, retornando o arquivo criptografado A_P. Por fim, quando um usuário do grupo desejar descriptografar um arquivo criptografado A_P, ele deve realizar a operação *Decrypt*. *Decrypt* exige que o usuário U interessado informe a chave pública K_G, a sua chave privada $K^{-1}_{\mathcal{L}_{U,G}}$ e o arquivo criptografado A_P. Caso o usuário possua os atributos $\mathcal{L}_{U,G}$ (incorporados em sua chave privada $K^{-1}_{\mathcal{L}_{U,G}}$) que atendam à política P empregada, o arquivo A_P será descriptografado e o original A será retornado ao usuário U.

Programas disponibilizados. O *kit* de ferramentas desenvolvido por Bethencourt *et al.* (BETHENCOURT; SAHAI; WATERS, 2015) provém um conjunto de programas que implementam o esquema de *Ciphertext-Policy Attribute-Based Encryption* proposto. A implementação baseia-se na biblioteca *Pairing Based Cryptography* (PBC) (LYNN,

2007). O *kit* de ferramentas corresponde a quatro programas responsáveis por executar as operações previamente abordadas: *cpabe-setup*, responsável por gerar a chave pública e a chave mestra de um grupo; *cpabe-keygen*, que possibilita a criação de chaves privadas para um determinado conjunto de atributos; *cpabe-enc*, que possui a função de criptografar um dado conteúdo; e *cpabe-dec*, que realiza a operação de descriptografia.

Encerrada a exposição dos fundamentos empregados nessa dissertação, apresenta-se, na próxima seção, os trabalhos relacionados que possibilitam a publicação de conteúdos de modo protegido em Redes Centradas em Informação.

2.3 Trabalhos Relacionados

Criptografia é o mecanismo mais fundamental para se implementar a publicação segura e privativa de conteúdos (JACOBSON et al., 2012). No entanto, os mecanismos de criptografia simétrica e assimétrica não são suficientes se empregados isoladamente em ICN: enquanto a primeira requer algum recurso externo (*ex.:* telefone ou *e-mail*) para o processo de distribuição de chaves, a segunda torna inócua as facilidades de *cache* na rede, uma vez que o conteúdo precisa ser cifrado para cada usuário.

De forma geral, as soluções propostas possuem como objetivos em comum o maior aproveitamento possível do mecanismo de *cache* na rede e a redução da complexidade associada ao controle de acesso. Apesar disso, elas diferem quanto à cardinalidade (na publicação de conteúdos), à intrusividade (isto é, introdução ou modificação de componentes na rede) e à forma como o conteúdo em si é protegido. A Tabela 2.1 apresenta uma visão geral das soluções propostas, organizadas segundo esses critérios.

Misra *et al.* (MISRA; TOURANI; MAJD, 2013) e Papanis *et al.* (PAPANIS et al., 2014) propõem que os provedores protejam o conteúdo empregando criptografia simétrica. Enquanto Misra *et al.* utilizam o conceito de criptografia em *broadcast* para a distribuição das chaves de acesso ao conteúdo, por sua vez, Papanis *et al.* aplicam o mecanismo CP-ABE. O *framework Content Centric Networking Access Control* (CCN-AC) (KURIHARAY; UZUN; WOOD, 2015) possibilita a instanciação de ambos os trabalhos, usando um arcabouço de controle de acesso que assegura a aplicação das políticas de acesso. Essas soluções tiram bastante proveito do mecanismo de *cache* na rede, por utilizarem criptografia simétrica para proteger o conteúdo. No entanto, elas permitem apenas um publicador no grupo de compartilhamento seguro. Uma vez que cada publicador precisa criar um par de chaves para cada usuário que deve ter acesso aos conteúdos,

Tabela 2.1: Propostas para o controle de acesso a conteúdos, organizadas por critérios de cardinalidade, de intrusividade e de proteção ao conteúdo.

Propostas	Cardinalidade		Intrusividade		Proteção do Conteúdo	
	um publicador	vários publicadores	não intrusivas	intrusivas	criptografia simétrica	criptografia assimétrica
Misra *et al.* (MISRA; TOURANI; MAJD, 2013)	x		x		x	
Papanis *et al.* (PAPANIS et al., 2014)	x		x		x	
Wood e Uzun (WOOD; UZUN, 2014)	x			x	x	
Mannes *et al.* (MANNES et al., 2014)	x		x			x
Singh *et al.* (SINGH et al., 2012)		x		x	nenhum	
Fotiou *et al.* (FOTIOU; MARIAS; POLYZOS, 2012)		x		x	x	
Hamdane *et al.* (HAMDANE et al., 2013)		x		x	x	
Ghali *et al.* (GHALI et al., 2015)		x		x	nenhum	

Fonte: do autor (2016).

no cenário em que todos são publicadores, o número de pares de chaves necessárias é proporcional à $\binom{n}{2}$.

As propostas de Wood e Uzun (WOOD; UZUN, 2014), e Mannes *et al.* (MANNES et al., 2014) seguem o mesmo modelo baseado em apenas um publicador. No entanto, elas empregam a técnica de reencriptação por *proxy*, proposta, orginalmente, por Ateniese *et al.* (ATENIESE et al., 2006). Essa técnica utiliza entidades de *proxy* na rede para transformar um conteúdo criptografado, com a chave pública do provedor, em outro conteúdo criptografado, agora com a chave pública do usuário. A principal diferença entre essas soluções reside no critério de intrusividade: enquanto na de Wood e Uzun, depende-se de nodos intermediários para atuar como redistribuidores de chaves de reencriptação, na proposta de Mannes *et al.*, possibilita-se que os próprios publicadores implementem esse papel. Embora menos intrusiva, essa última requer que o publicador do conteúdo esteja sempre disponível para criar e distribuir as chaves de reencriptação sempre que o conteúdo for acessado.

Outro conjunto de soluções propostas avança no critério de cardinalidade, permitindo vários publicadores no mesmo grupo de compartilhamento seguro, evitando, assim, o problema de explosão combinatorial de chaves. Fotiou *et al.* e Hamdane *et al.* (FOTIOU; MARIAS; POLYZOS, 2012; HAMDANE et al., 2013) sugerem o emprego de *third-parties* encarregados de armazenar os conteúdos e/ou de realizar o controle de acesso. Contudo, essa técnica requer o uso de ICNs específicas para o seu funcionamento. Além disso, caso o componente responsável por autenticar os usuários esteja indisponível,

não será permitido o acesso de qualquer conteúdo, mesmo que ele esteja em *cache*.

As demais soluções dependem, porém, de entidades adicionais para armazenar conteúdos e/ou realizar controle de acesso, o que implica em modificações na arquitetura ICN ou na dependência da disponibilidade dessas entidades. As soluções de Singh *et al.* (SINGH et al., 2012) e Ghali *et al.* (GHALI et al., 2015) são, particularmente, dependentes do comportamento honesto dessas entidades; caso sejam subvertidas, a privacidade dos conteúdos controlados por essas poderá ser comprometida.

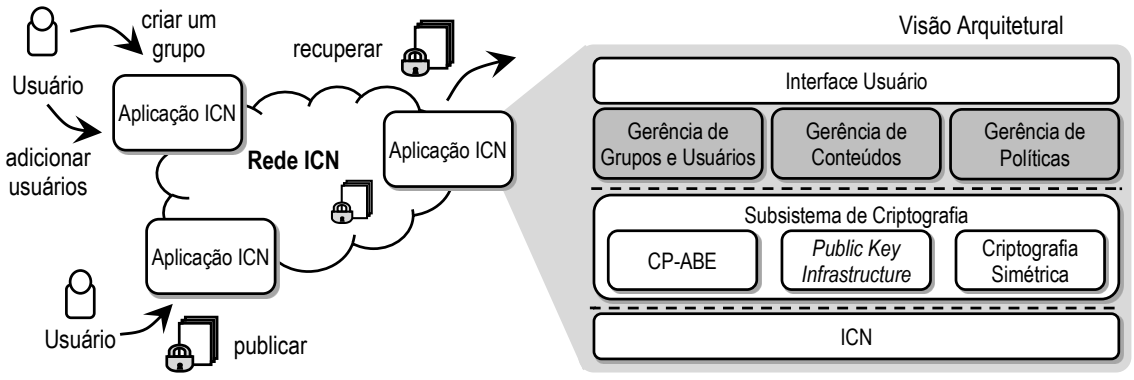
Um aspecto importante observado na Tabela 2.1 é que nenhuma das propostas existentes reúne as características de não intrusividade e de suporte a múltiplos publicadores. No capítulo a seguir, apresenta-se um modelo de segurança que satisfaz esses requisitos, sem dependência de componentes específicos na arquitetura subjacente, e sem aumentar a complexidade do processo de gerenciamento de chaves.

3 SOLUÇÃO PROPOSTA

Para solucionar o problema exposto no capítulo anterior, formalizou-se uma proposta com o intuito de permitir as seguintes características, já previamente discutidas: (i) possibilitar que usuários participem de grupos de compartilhamento seguro; (ii) restringir o acesso de conteúdo a um subconjunto específico de usuários; (iii) suportar múltiplos publicadores; (iv) ser agnóstico de arquitetura ICN; (v) manter o número de chaves criptográficas proporcional ao número de membros; e (vi) manter o custo marginal para publicação e recuperação de conteúdos.

A partir dos requisitos descritos, apresenta-se na Figura 3.1 uma visão geral do modelo e da arquitetura que o apoia, destacando ainda os atores e componentes envolvidos. O compartilhamento seguro de conteúdos inicia-se quando um usuário interage com uma instância da *Aplicação ICN*, executando na sua própria estação local, para *criar um grupo*. A aplicação ICN corresponde a um *software* (o equivalente a um navegador *web*) que permite o compartilhamento de conteúdos via paradigma ICN, estendido para suportar o modelo de segurança proposto.

Figura 3.1: Visão arquitetural da solução proposta.



Fonte: do autor (2016).

Apenas o usuário que criou o grupo (denominado de *administrador* no restante da dissertação) poderá *adicionar usuários* ao mesmo. Conforme será discutido na próxima seção, a adição de um usuário consiste na criação de uma credencial de membro (uma chave privada) e na entrega dessa credencial para o usuário. Essa entrega deve ocorrer via compartilhamento seguro, por exemplo, com o uso de criptografia assimétrica. Uma vez habilitado como membro do grupo, o usuário poderá compartilhar conteúdos (isto é, *publicar* e *recuperar* conteúdos) de forma segura com os demais membros.

A visão arquitetural apresentada na parte direita da Figura 3.1 destaca (em cinza)

Tabela 3.1: Glossário de notações relacionadas ao modelo de segurança proposto.

Notação	Definição formal	Descrição
Entidades e conjuntos		
C		Conteúdo original
G		Grupo de compartilhamento seguro
U		Usuário (membro do grupo)
$\mathbb{P}$		Política de acesso
$\mathcal{L}_G$		Conjunto de atributos existentes no grupo G
$\mathcal{L}_{U,G} \subseteq \mathcal{L}_G$		Conjunto de atributos do usuário U no grupo G
Chaves criptográficas		
K_s		Chave de criptografia simétrica
K_U		Chave pública do usuário U
K_U^{-1}		Chave privada do usuário U
K_G		Chave pública do grupo de compartilhamento seguro G
M_G		Chave mestra do grupo de compartilhamento seguro G
$K_{\mathcal{L}_{U,G}}^{-1}$		Chave privada do usuário U no grupo G
Funções criptográficas		
$\{X\}_{K_x}$		Elemento X cifrado usando a chave K_x (simétrica ou assimétrica)
$\{X\}_{(K_G,P)}$		Elemento X cifrado usando a chave do grupo K_G e a política P
Modelo de segurança		
$\widehat{X}$		Identificador do elemento X
C_P	$C_P = \langle \{C\}_{K_s}, \widehat{H_C} \rangle$	Conteúdo C protegido
H_C	$H_C = \langle \{K_s\}_{(K_G,\mathbb{P})}, \widehat{K_G} \rangle$	Bloco habilitador de um conteúdo protegido C_P

Fonte: do autor (2016).

os componentes que fazem parte da proposta. O componente *Gerência de Grupos e Usuários* reúne as funcionalidades para criação de grupos e adição de membros. O componente *Gerência de Conteúdos* está relacionado com a publicação e recuperação segura de conteúdos. Por fim, o componente *Gerência de Políticas* possibilita o controle fino de acesso aos conteúdos, apoiando, por exemplo, a concessão e a revogação de acesso. Esses componentes são apoiados por um subsistema de criptografia, composto por um mecanismo de criptografia simétrica, uma solução de infraestrutura de chaves públicas (*Public Key Infrastructure, PKI*) e um mecanismo de criptografia baseada em atributos (CP-ABE) (BETHENCOURT; SAHAI; WATERS, 2007) (doravante referido como *componente CP-ABE*). Observe que os componentes da solução proposta acomodam-se exclusivamente entre as camadas de interface com o usuário e com a ICN, sendo restritos, portanto, ao *software* que executa na estação do usuário.

As Seções 3.1, 3.2 e 3.3, a seguir, descrevem em detalhes as funcionalidades proporcionadas por cada um dos componentes destacados na visão arquitetural da Figura 3.1. A Seção 3.4 encerra a apresentação da proposta, discutindo possíveis estratégias de ataque contra a solução. Para a explicação que segue, adota-se um conjunto de notações e convenções sumarizado na Tabela 3.1.

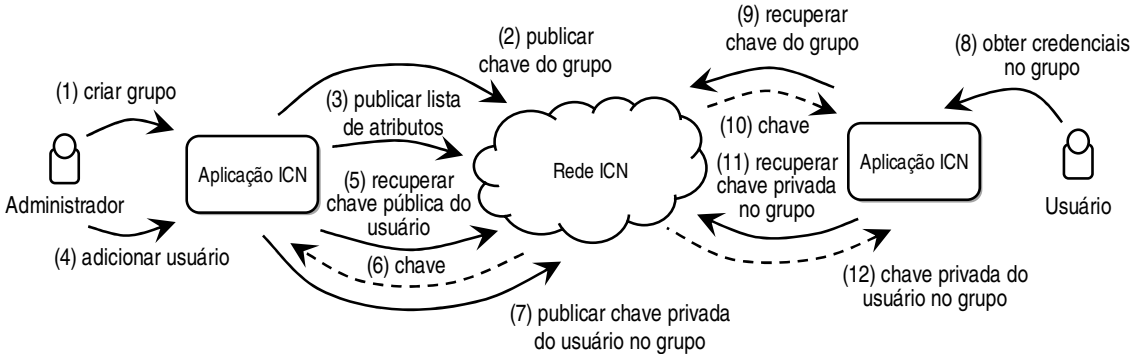
3.1 Gerência de Grupos e Usuários

A Figura 3.2 ilustra a dinâmica do processo de manutenção de grupos de compartilhamento seguro, nesse sentido, destacando as atividades de criação de grupos e de adição de usuários ao mesmo.

Criação de grupo. Como brevemente mencionado anteriormente, o administrador inicia esse processo ao interagir com a aplicação ICN (fluxo 1 na Figura 3.2). Esse processo compreende basicamente a criação de um par de chaves pública K_G e mestra M_G para o grupo, o que é feito com o apoio do componente CP-ABE. O grupo passa a existir na rede a partir do momento que se dissemina a chave pública K_G na rede, na forma de um objeto, utilizando como identificador o nome do grupo (fluxo 2). A chave mestra M_G é mantida em segredo pelo administrador.

Cada grupo possui um conjunto de atributos $\mathcal{L}_G$, que são utilizados para descrever os usuários membros. Os atributos são cadeias de caracteres de tamanho livre, definidos pelo administrador, e existentes somente no escopo daquele grupo. Por exemplo, suponha um grupo composto por integrantes de uma comunidade acadêmica. Alguns atributos possíveis são "professor", "graduando", "mestrando", "doutorando" e "pós-doutorando". Note que não há uma regra geral para a formação dos atributos. Da mesma forma, a semântica dos atributos é dada pelo contexto do grupo. A lista de atributos, também, deve ser publicada como um objeto na rede (fluxo 3).

Figura 3.2: Manutenção de grupos de compartilhamento seguro.



Fonte: do autor (2016).

Adição de usuário a um grupo. O administrador inicia esse processo, por intermédio da aplicação ICN (fluxo 4), ao informar os atributos que o novo membro possuirá. Esse processo se desdobra em três passos: (*i*) criar a chave privada do usuário no grupo $K^{-1}_{\mathcal{L}_{U,G}}$; (*ii*) publicar a chave $K^{-1}_{\mathcal{L}_{U,G}}$ de forma segura na rede, de modo que apenas o usuário adicionado possa recuperá-la; e (*iii*) recuperar a chave $K^{-1}_{\mathcal{L}_{U,G}}$ da rede (esse último passo

feito pelo usuário adicionado). Esses passos são descritos em detalhes a seguir.

A chave privada $K^{-1}_{\mathcal{L}_{U,G}}$ é criada com o apoio do componente CP-ABE. Para tal, o administrador deve especificar um conjunto de atributos $\mathcal{L}_{U,G} \subseteq \mathcal{L}_G$ para o usuário. Destaca-se que, por simplicidade, o administrador conhece os atributos $\mathcal{L}_{U,G}$ designados para descrever um usuário U; no mundo real ele potencialmente precisaria da ajuda de alguma pessoa e/ou mecanismo para validar esses atributos. A criação de $K^{-1}_{\mathcal{L}_{U,G}}$ requer, também, a chave mestra do grupo M_G. Após criada, disponibiliza-se a chave $K^{-1}_{\mathcal{L}_{U,G}}$ (com os atributos $\mathcal{L}_{U,G}$ incorporados) na rede para que o usuário alvo possa recuperá-la. A entrega deve ocorrer de forma privativa, visto que a posse de $K^{-1}_{\mathcal{L}_{U,G}}$ materializa a participação no grupo. Em outras palavras, $K^{-1}_{\mathcal{L}_{U,G}}$ será empregada para recuperar conteúdos protegidos no grupo (conforme discutido na próxima seção). Para realizar essa entrega, o administrador precisa recuperar da rede a chave pública K_U do usuário e verificá-la via mecanismo de infraestrutura de chave pública (fluxos 5 e 6). A chave $K^{-1}_{\mathcal{L}_{U,G}}$ é criptografada usando K_U, assim, gerando a chave criptografada $\{K^{-1}_{\mathcal{L}_{U,G}}\}_{K_U}$, a qual é publicada como um objeto na rede (fluxo 7). Por fim, o usuário alvo precisa recuperar as chaves $K^{-1}_{\mathcal{L}_{U,G}}$ e K_G da rede para poder utilizá-las na publicação e recuperação de conteúdos no grupo. O usuário inicia esse procedimento (fluxo 8) especificando o nome do grupo para recuperar esses objetos. A aplicação recupera, então, a chave pública do grupo K_G (fluxos 9 e 10) e a chave privada do usuário no grupo, criptografada $\{K^{-1}_{\mathcal{L}_{U,G}}\}_{K_U}$ (fluxos 11 e 12). A chave privada no grupo é descriptografada usando a própria chave privada do usuário K^{-1}_U. A partir de então, ele está habilitado para publicar e recuperar conteúdos no grupo.

Há três observações importantes sobre o modelo de segurança proposto. Primeiro, cada usuário deve possuir uma chave privada $K^{-1}_{\mathcal{L}_{U,G}}$ no grupo para recuperar conteúdos do mesmo. Portanto, o administrador precisa se adicionar ao grupo (ou seja, criar a sua própria chave $K^{-1}_{\mathcal{L}_{U,G}}$) para poder publicar e recuperar conteúdos. A segunda observação está relacionada aos atributos de usuários. Caso o administrador deseje descrever o usuário com um atributo não contido em $\mathcal{L}_G$, esse atributo deve ser adicionado à lista, e a mesma deve ser atualizada na rede. Essa atualização é essencial para que os usuários, ao publicarem conteúdos, saibam quais são os atributos vigentes no grupo. Terceiro, após a criação do grupo e a adição de usuários, o compartilhamento seguro de conteúdos entre eles independe da disponibilidade do administrador.

3.2 Gerência de Conteúdos

A gerência de conteúdos reúne todos os procedimentos necessários para a publicação e a recuperação segura de conteúdos na rede. Esses procedimentos se apoiam em dois elementos importantes, ou seja, o conteúdo protegido e o bloco habilitador.

Conteúdo protegido e bloco habilitador. O conteúdo protegido corresponde a um conteúdo cifrado usando criptografia simétrica. O bloco habilitador, por sua vez, contém a chave necessária para decifrar um dado conteúdo. No modelo proposto, um conteúdo protegido possui um (e apenas um) bloco habilitador correspondente. Formalmente, um conteúdo protegido é uma tupla $C_P = \langle \{C\}_{K_s}, \widehat{H_C} \rangle$, onde $\{C\}_{K_s}$ corresponde ao conteúdo original C, cifrado usando uma chave simétrica K_s, e $\widehat{H_C}$ é o identificador do bloco habilitador desse conteúdo protegido. Um bloco habilitador é uma tupla $H_C = \langle \{K_s\}_{(K_G, \mathbb{P})}, \widehat{K_G} \rangle$, onde $\{K_s\}_{(K_G, \mathbb{P})}$ corresponde à chave K_s (usada para cifrar C) cifrada usando (*i*) a chave pública K_G do grupo e (*ii*) uma política de acesso $\mathbb{P}$, e $\widehat{K_G}$ é o identificador da chave pública do grupo. Observe que esse projeto possibilita que vários conteúdos sejam protegidos por um mesmo bloco habilitador. Essa característica pode ser conveniente quando se desejar publicar múltiplos conteúdos usando uma política de acesso única.

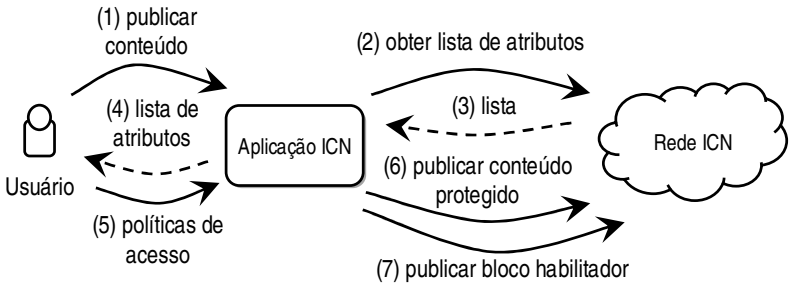
O processo para construir um bloco habilitador compreende (*i*) a definição da chave simétrica que será usada para cifrar o conteúdo e (*ii*) a especificação (pelo usuário) da *política de controle de acesso* $\mathbb{P}$. Em relação à chave simétrica, ela pode ser gerada automaticamente pela aplicação ICN ou informada pelo usuário. Sobre a política $\mathbb{P}$, ela determinará quais usuários do grupo estarão autorizados a decifrar o conteúdo e é executada envolvendo elementos da lista de atributos do grupo $\mathcal{L}_G$.

Para ilustrar o conceito de políticas de acesso, suponha o grupo de compartilhamento de conteúdos acadêmicos nos quais os usuários possuem um (ou mais) dos seguintes atributos: $\mathcal{L}_G = \{\text{professor, graduando, mestrando, doutorando, pos-doutorando}\}$. O usuário pode, por exemplo, especificar uma política $\mathbb{P} = \{\text{professor } \textbf{ou} \text{ graduando}\}$. Nesse caso, a descriptografia baseada em atributos (usando o componente CP-ABE) poderá ser realizada apenas pelos usuários detentores do atributo "professor" ou "graduando" (ou ambos). A metodologia para a formação dessas políticas será discutida mais detalhadamente na Seção 3.3. Uma vez determinada $\mathbb{P}$, emprega-se o componente CP-ABE para cifrar K_s. Essa cifragem é feita usando a chave pública K_G do grupo e a política $\mathbb{P}$. A chave cifrada $\{K_s\}_{(K_G, \mathbb{P})}$ é então encapsulada em H_C.

Após construído o bloco habilitador, o conteúdo protegido pode, então, ser formado. A sua construção compreende a cifragem do conteúdo original C a ser disseminado na rede, para isso, utilizando-se a chave simétrica K_s encapsulada no bloco habilitador que será associado a esse conteúdo.

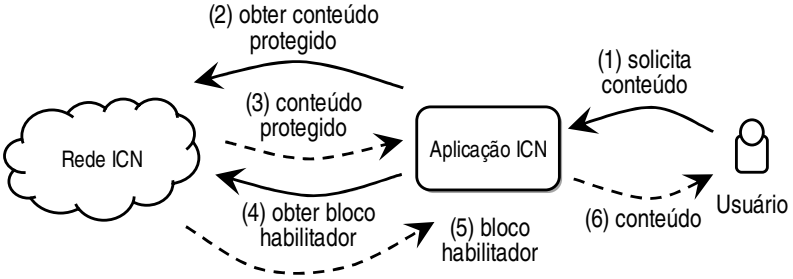
Publicação de conteúdos. A Figura 3.3 ilustra a dinâmica do processo de publicação de um conteúdo no grupo. O usuário inicia esse processo ao interagir com a aplicação ICN (fluxo 1 na Figura 3.3), informando o conteúdo C a ser publicado. Nesse momento, seis passos são executados. Primeiro, a aplicação recorre à rede para obter a lista atualizada dos atributos do grupo $\mathcal{L}_G$ (fluxos 2 e 3) e disponibiliza-a ao usuário. Em seguida, o usuário elabora a política de acesso $\mathbb{P}$, conforme as restrições de acesso desejadas (fluxos 4 e 5). O terceiro passo, realizado pela aplicação, consiste em criptografar o conteúdo C usando uma chave simétrica K_s. O quarto passo corresponde à construção do bloco habilitador H_C do conteúdo, conforme discutido anteriormente. No quinto passo, o conteúdo protegido C_P é construído encapsulando o conteúdo cifrado $\{C\}_{K_s}$ e o identificador para o bloco habilitador $\widehat{H_C}$. Por fim, ambos, conteúdo protegido C_P e bloco habilitador H_C (com o identificador da chave pública do grupo $\widehat{K_G}$), são publicados na rede (fluxos 6 e 7).

Figura 3.3: Etapas para publicação dos conteúdos.



Fonte: do autor (2016).

Figura 3.4: Etapas para recuperação dos conteúdos.



Fonte: do autor (2016).

Recuperação de conteúdos. O processo de recuperação, ilustrado na Figura 3.4, inicia-se quando o usuário solicita um conteúdo (fluxo 1). A aplicação solicita à rede o

Tabela 3.2: Regras para a definição de políticas de acesso a conteúdos.

⟨política⟩ ::= ⟨atributo⟩ | '(' ⟨política⟩ ')'
| ⟨atributo⟩ e ⟨política⟩ | ⟨atributo⟩ ou ⟨política⟩
| ⟨atributo⟩ = ⟨inteiro⟩ | ⟨atributo⟩ < ⟨inteiro⟩ | ⟨atributo⟩ > ⟨inteiro⟩
| ⟨inteiro⟩ de '(' ⟨coleção⟩ ')'
⟨coleção⟩ ::= ⟨política⟩ ',' ⟨política⟩ | ⟨coleção⟩

Fonte: do autor (2016).

conteúdo protegido C_P correspondente (fluxos 2 e 3), o qual é obtido da fonte mais próxima. Ao abrir C_P, a aplicação identifica qual bloco habilitador H_C está relacionado a esse conteúdo (por meio do identificador $\widehat{H_C}$ presente na tupla). A aplicação solicita, então, H_C à rede (fluxos 4 e 5) para recuperar a chave simétrica criptografada $\{K_s\}_{(K_G,\mathbb{P})}$. A chave simétrica $\{K_s\}_{(K_G,\mathbb{P})}$ recuperada é submetida ao componente CP-ABE para decifragem. Para isso, o usuário utiliza a sua chave privada no grupo $K^{-1}_{\mathcal{L}_{U,G}}$, que possui os atributos $\mathcal{L}_{U,G}$ incorporados. A chave $\{K_s\}_{(K_G,\mathbb{P})}$ é descriptografada *se e somente se* a política de acesso $\mathbb{P}$ usada para cifrá-la for compatível com os atributos utilizados pelo administrador na criação de $K^{-1}_{\mathcal{L}_{U,G}}$ (ao adicionar o usuário no grupo). Caso seja descriptografado com sucesso, o conteúdo é entregue ao usuário (fluxo 6).

3.3 Gerência de Políticas

O modelo de segurança, por meio do componente *Gerência de Políticas*, permite determinar quando e quais usuários podem ter acesso aos conteúdos publicados. Em outras palavras, o modelo reúne mecanismos que permitem conceder, limitar e revogar autorizações de acesso a conteúdos, com base em políticas.

A composição de uma política envolve operadores relacionais ($>$, $<$ e $=$) e lógicos (**e** e **ou**). Com o apoio do componente CP-ABE (BETHENCOURT; SAHAI; WATERS, 2007), esses operadores permitem determinar quando e quais usuários têm acesso ao conteúdo. A Tabela 3.2 apresenta o conjunto de regras que regem o processo de construção de políticas de acesso a conteúdo. Nesse conjunto, $<$*atributo*$> \in \mathcal{L}_G$ e $<$*inteiro*$> \in \mathbb{N}$.

Concessão de acesso. Ela consiste basicamente em definir uma política que um determinado conjunto de usuários do grupo deve satisfazer para decifrar um conteúdo. Observe que uma política pode ser formada por apenas um atributo. Nesse caso, para ter acesso ao conteúdo, a chave privada do usuário no grupo $K^{-1}_{\mathcal{L}_{U,G}}$ deve atender à restrição descrita pela política. Por exemplo, suponha dois usuários: *joão* com

os atributos $\mathcal{L}_{\text{joão},G} = \{\text{professor}, \text{pesquisador}\}$, e *josé* com os atributos $\mathcal{L}_{\text{josé},G} = \{\text{aluno}, \text{pesquisador}\}$. Uma política de acesso $\mathbb{P}_1 = \{\text{professor}\}$ permite acesso ao conteúdo apenas para *joão*. A política de acesso $\mathbb{P}_2 = \{\text{pesquisador}\}$, por sua vez, permite o acesso a ambos. As regras não permitem a formação de políticas "coringa", isto é, para acesso universal. Uma forma de alcançar todos os usuários é citar um a um os atributos dos mesmos na política de acesso. Alternativamente, o administrador pode definir um atributo comum aos usuários (*ex.:* "todos"); assim, cada publicador poderá usá-lo em políticas que visem ao acesso universal.

Os atributos (dos usuários e de políticas) podem ainda ser valorados. Eles podem ser criados para indicar, por exemplo, o nível do usuário na hierarquia de uma corporação. Suponha o usuário *joão* com o atributo "nível = 5" e *josé* com o atributo "nível = 2". Caso deseje-se publicar um conteúdo somente para os usuários com nível 3 ou superior (no caso, *joão*), basta definir a política $\mathbb{P} = \{\text{nível} > 2\}$ (assumindo que os níveis de hierarquia são dados por números discretos).

Revogação de acesso. A revogação baseia-se na possibilidade de publicar uma versão mais recente de um bloco habilitador (por exemplo, quando a versão existente expirar na *cache* dos roteadores). Assim, o usuário pode reformular a política daquele bloco para restringir o acesso de algum usuário particular. Há duas estratégias que podem ser empregadas. A primeira é definir um atributo único para cada usuário. Nesse caso, a revogação compreenderia selecionar todos os usuários exceto aquele(s) cujo acesso deve ser revogado. Supondo os usuários *joão*, *josé*, *maria* e *fátima*, e um conteúdo publicado com $\mathbb{P} = \{\text{todos}\}$, revogar o acesso de *joão* a esse conteúdo requer que a política seja reformulada para $\mathbb{P}' = \{1 \text{ de } (\text{jose}, \text{maria}, \text{fatima})\}$. A complexidade de definir essa restrição para grupos com dezenas de usuários ou mais pode ser trivialmente resolvida na interface com o usuário, não sendo necessário transportá-la para o modelo.

A segunda forma de implementar revogação é por meio de expiração, usando o mecanismo de comparação de valores de atributos. Para ilustrar, suponha que *joão* possui o atributo "criado = 1435708800" (2015-07-01 00:00:00) e *maria*, "criado = 1446336000" (2015-11-01 00:00:00). A semântica desses atributos corresponde à data e hora (no formato de *timestamp*) que cada um foi adicionado ao grupo. Agora, suponha um conteúdo com a política $\mathbb{P} = \{\text{criado} > 1420070400\}$. Esta garante acesso ao conteúdo apenas aos usuários adicionados ao grupo após 1º de janeiro, o que se aplica a *joão* e *maria*. O acesso de *joão* pode ser revogado por expiração, nesse caso, ao publicar um novo bloco habilitador com $\mathbb{P}' = \{\text{criado} > 1443657600\}$ (1º de outubro). Destaca-se que ambas estratégias

poderiam ser usadas em conjunto permitindo revogação a curto e a longo prazo.

3.4 Modelos de Ataque

Após ter-se apresentado uma visão geral do modelo proposto, passa-se agora a analisar sua robustez perante possíveis estratégias que um atacante possa lançar mão para violar a privacidade dos conteúdos protegidos publicados. Primeiramente, cabe destacar que o modelo foi projetado considerando três premissas básicas: (*i*) o administrador do grupo é confiável; (*ii*) os membros do grupo mantém em segredo suas respectivas chaves privadas no grupo; e (*iii*) o membro com acesso a um conteúdo protegido não divulga a chave simétrica usada para cifrá-lo.

As implicações das premissas enumeradas acima são descritas a seguir. A primeira estabelece que a concessão de atributos aos usuários membros do grupo é feita de forma confiável. Ou seja, o administrador não deturpará o uso dos atributos ao atribuí-los aos membros, por exemplo, conferindo atributo(s) a um adversário ou mesmo a usuários que não sejam compatíveis com aquele(s) atributo(s). Essa premissa é similar à confiabilidade depositada no gerente de projetos estratégicos e sensíveis, por exemplo, de um *software open-source* (no qual a admissão de um adversário ao time de desenvolvedores pode levar à inclusão de código malicioso no *software* desenvolvido). A segunda premissa está relacionada à segurança dos conteúdos que são acessíveis por determinados usuários. Essa premissa é equivalente à guarda das credenciais de acesso que um usuário possui em um sistema (por exemplo, a chave para um servidor *Secure Shell* (SSH) ou a senha para um portal de conteúdos pagos). Finalmente, a terceira premissa implica na segurança dos conteúdos protegidos que foram acessados. Nesse caso, o vazamento da chave simétrica seria equivalente ao ato de vazar o próprio conteúdo.

De acordo com essas premissas, um atacante pode burlar o modelo de segurança proposto apenas se comprometer (via acesso físico ou remoto) a própria estação do administrador/usuário para subtrair a chave privada do grupo ou as chaves privadas dos usuários. Observe que esse ataque foge ao escopo desta dissertação, uma vez que a proteção contra o mesmo requer mecanismos que garantam a segurança da própria estação do administrador/usuário. Assumindo a segurança desta, o modelo mantém-se resiliente mesmo que o atacante possua acesso privilegiado a quaisquer elementos da rede ou nela disponíveis (roteadores, blocos habilitadores, chaves públicas, listas de atributos, etc.).

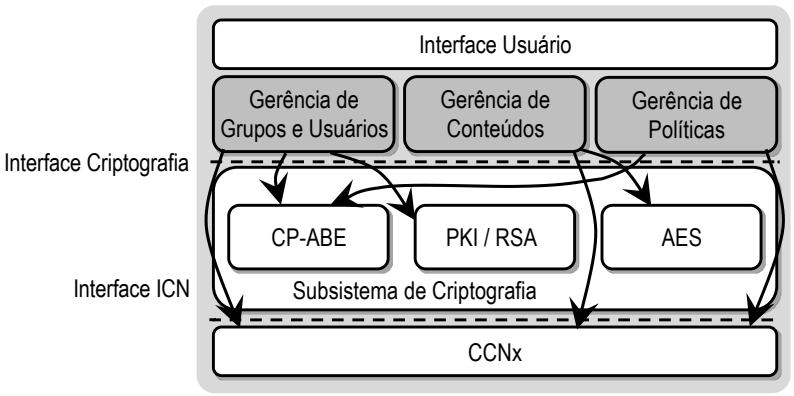
O modelo proposto é robusto a ataques em conluio. Por exemplo, suponha um

usuário com o atributo professor e outro com o atributo aluno. Mesmo que esses usuários atuem em conjunto, eles não podem decifrar um conteúdo protegido com a política $\mathbb{P} = \{\text{professor } \mathbf{e} \text{ aluno}\}$, visto que a política requer que o mesmo membro possua ambos os atributos, simultaneamente. Conforme discutido anteriormente, apenas usuários pertencentes ao grupo e que satisfaçam integralmente as políticas de acesso definidas podem acessar conteúdos publicados. O modelo não impede que usuários não pertencentes ao grupo publiquem conteúdos protegidos no mesmo. Isso é possível visto que a publicação requer apenas a chave pública e a lista de atributos do grupo, ambos disponíveis em claro na rede. Os membros do grupo podem evitar o acesso a conteúdos indesejados verificando a origem dos mesmos usando, por exemplo, mecanismos de autocertificação de conteúdos da própria rede (JACOBSON et al., 2012). Por fim, um outro tipo de ataque consiste em um publicador malicioso disponibilizar vários conteúdos usando o mesmo identificador de um conteúdo do grupo, que poderia culminar em um ataque de negação de serviço (DoS). O ataque de negação de serviço seria bem sucedido, pois mesmo o usuário identificando que os conteúdos recuperados não correspondem ao solicitado e requisitasse outro, o tempo necessário para localizar o verdadeiro tornaria a operação inviável. Contudo, destaca-se que esse tipo de ataque representa um problema em aberto em ICN, estando fora do escopo da solução proposta.

4 PROTÓTIPO IMPLEMENTADO

Neste capítulo são apresentadas as tecnologias empregadas no desenvolvimento de um protótipo da solução e como elas se relacionam para implementar os componentes *Gerência de Grupos e Usuários*, *Gerência de Conteúdos* e *Gerência de Políticas*. Implementou-se a proposta para o ambiente UNIX (TANENBAUM; BOS, 2014) utilizando as linguagens de programação Python 2.7 (Python Software Foundation, 2015b), Java SE 8 (Oracle, 2015b) e a de comando *bash shell* (*bash*) (TANENBAUM; BOS, 2014).

Figura 4.1: Visão da implementação da solução proposta.



Fonte: do autor (2016).

A Figura 4.1 retorna à visão arquitetural detalhada, enfatizando, agora, as tecnologias adotadas em cada componente e suas relações. O componente *Gerência de Grupos e Usuários* foi implementado por meio de mecanismos que invocam primitivas oferecidas pelo Subsistema de Criptografia e por CCNx a fim de executar os algoritmos de criação de grupo e adição de um usuário a um grupo. Mais especificamente, usa-se os componentes de criptografia CP-ABE, PKI/RSA e CCNx. O componente CP-ABE permite a criação das chaves relacionadas aos grupos administrados pelo usuário. Os componentes PKI/RSA auxiliam o módulo na distribuição segura das chaves via CCNx.

O componente *Gerência de Conteúdos* foi desenvolvido por meio de mecanismos que invocam primitivas oferecidas pelo Subsistema de Criptografia e por CCNx a fim de executar os algoritmos de publicação de conteúdos e recuperação de conteúdos em um grupo. De modo mais preciso, o módulo utiliza os componentes de criptografia AES e o CCNx. O componente AES possibilita ao módulo a criptografia/descriptografia de conteúdos disponibilizados por usuários do grupo. O componente CCNx assiste o módulo na publicação/recuperação de conteúdos.

O componente *Gerência de Políticas* foi implementado por meio de mecanismos que empregam primitivas disponibilizadas pela Interface de Criptografia e pela Interface ICN a fim de executar os algoritmos de concessão de acesso e revogação de acesso a um conjunto de usuários. Especificamente, *Gerência de Políticas* invoca os componentes de criptografia CP-ABE e CCNx. O componente CP-ABE restringe o acesso à chave simétrica, associada a um conteúdo, conforme a política utilizada. O componente apoia a distribuição segura das chaves simétricas criptografadas via a CCNx.

Na próxima seção (Seção 4.1), apresenta-se, em detalhes, o conjunto comum de métodos expostos pelo Subsistema de Criptografia (AES, PKI/RSA e CP-ABE) e por CCNx. Subsequentemente (Seção 4.2), exibe-se como esses métodos são utilizados para materializar os diferentes algoritmos previstos pela solução proposta para permitir a publicação e a recuperação de um conteúdo protegido.

4.1 Métodos Empregados

Nesta seção, aborda-se os diferentes métodos expostos pela implementação dos componentes *Subsistema de Criptografia* e *CCNx* por meio das *Interfaces Criptografia* e *ICN*. A primeira apresenta três conjuntos de métodos: de criptografia simétrica, de criptografia assimétrica e CP-ABE, enquanto a segunda disponibiliza acesso ao componente CCNx.

Métodos de criptografia simétrica. O componente de criptografia simétrica foi implementado em Python 2.7 (Python Software Foundation, 2015b) usando a biblioteca *Crypto.Cipher.AES* (Python Software Foundation, 2015a) e a criptografia AES-CBC (BISHOP, 2004). É materializado pelos métodos *generateAESKey*, *encryptFileAES* e *decryptFileAES*, como enumerado na Tabela 4.1. O método *generateAESKey* gera uma chave simétrica, dados o tamanho da chave simétrica e a localização onde a chave deve ser armazenada. Após gerada uma chave simétrica, ela pode ser utilizada para criptografar um conteúdo. O método *encryptFileAES* retorna um conteúdo criptografado, dados um conteúdo, uma chave simétrica e a localização onde o conteúdo criptografado será salvo. Por fim, o método *decryptFileAES* retorna um conteúdo descriptografado, informados um conteúdo protegido, a chave simétrica e a localização de onde o conteúdo será armazenado.

Métodos de criptografia assimétrica. O componente de criptografia assimétrica foi implementado em Java SE 8 (Oracle, 2015b) empregando a biblioteca RSA *criptou-*

Tabela 4.1: Métodos AES empregados.

Nome	Descrição	Entrada	Saída
generateAESKey	Gera uma chave simétrica	Tamanho da e localização onde a chave simétrica será salva na máquina local	Chave simétrica
encryptFileAES	Criptografa um conteúdo	Localizações onde o conteúdo e a chave simétrica estão, e onde o conteúdo criptografado será salvo na máquina local	Conteúdo criptografado
decryptFileAES	Descriptografa um conteúdo criptografado	Localizações onde o conteúdo criptografado e a chave simétrica estão, e onde o conteúdo será salvo na máquina local	Conteúdo plano

Fonte: do autor (2016).

Tabela 4.2: Métodos RSA empregados.

Nome	Descrição	Entrada	Saída
generateRSAKey	Gera um par de chaves	Tamanho das e localizações onde as chaves pública e privada serão salvas na máquina local	Chaves pública e privada do usuário
encryptFileRSA	Criptografa um conteúdo	Localizações onde a chave pública e o conteúdo estão, e onde o conteúdo criptografado será salvo	Conteúdo criptografado
decryptFileRSA	Descriptografa um conteúdo criptografado	Localizações onde a chave privada e o conteúdo criptografado estão, e onde o conteúdo será salvo	Conteúdo plano

Fonte: do autor (2016).

til.RSA (Oracle, 2015a). É materializado em três métodos: *generateRSAKey*, *encryptFileRSA* e *decryptFileRSA*, exibidos na Tabela 4.2. O método *generateRSAKey* cria um par de chaves, pública e privada, específico de um usuário, dados o tamanho das e as localizações onde as chaves serão salvas. Uma vez criado um par de chaves, pode-se utilizá-lo para cifrar um conteúdo. O método *encryptFileRSA* retorna um conteúdo criptografado, informados a chave pública de um usúario, um conteúdo e a localização onde será salvo o conteúdo criptografado. Finalmente, o método *decryptFileRSA* retorna o conteúdo plano, dados um conteúdo criptografado, a chave privada de um usuário e a localização onde o conteúdo descriptografado será salvo.

Métodos CP-ABE. O componente CP-ABE corresponde aos programas desenvolvidos por Bethencourt *et al.* (BETHENCOURT; SAHAI; WATERS, 2015). A implementação CP-ABE, conforme previamente abordado na Seção 2.2, possui quatro métodos: *cpabe-setup*, *cpabe-keygen*, *cpabe-enc* e *cpabe-dec*, listados na Tabela 4.3. O método *cpabe-setup* cria as chaves pública e mestra de um grupo, informadas as localizações onde as chaves serão salvas. Uma vez criado o par de chaves de um grupo, o administrador pode executar o método *cpabe-keygen* para criar chaves para usuários do grupo. O método *cpabe-keygen* permite a geração de uma chave privada para um determinado usuário, dados os atributos que descrevem o usuário, as chaves pública e mestra, e onde será salva a chave privada do usuário no grupo. Após o usuário receber a sua chave privada no grupo, por um modo seguro, o mesmo passa a poder criptografar um conteúdo para o

grupo. O método *cpabe-enc* permite que um conteúdo seja criptografado, informadas a chave pública do grupo, a chave privada de um usuário no grupo e a localização de onde será armazenado o conteúdo criptografado. O último método, *cpabe-enc* descriptografa um conteúdo cifrado, dados as chaves pública e privada do usuário no grupo, o conteúdo criptografado e a localização onde o conteúdo plano será salvo.

Tabela 4.3: Métodos CP-ABE empregados.

Nome	Descrição	Entrada	Saída
cpabe-setup	Gera par de chaves pública e mestra do grupo	Localizações onde as chaves pública e mestra serão salvas na máquina local	Chaves pública e mestra do grupo
cpabe-keygen	Gera a chave privada de um usuário no grupo baseado em um conjunto de atributos	Atributos relativos a um usuário, localizações onde as chaves pública e mestra estão, e onde a chave privada de um usuário no grupo será salva na máquina local	Chave privada de um usuário no grupo
cpabe-enc	Criptografa um conteúdo usando uma determinada política	Localizações onde a chave pública está e onde o conteúdo criptografado será salvo na máquina local, e política a ser empregada	Conteúdo criptografado
cpabe-dec	Descriptografa um conteúdo criptografado	Localizações onde as chaves pública do grupo, privada do usuário no grupo e o conteúdo criptografado estão, e onde o conteúdo decriptografado será salvo	Conteúdo plano

Fonte: do autor (2016).

Métodos CCNx. O componente CCNx corresponde aos programas desenvolvidos pela PARC (Palo Alto Research Center, 2015), já abordado na Subseção 2.1.2. A implementação do protótipo CCNx 0.8.2 disponibiliza um conjunto de programas, dos quais emprega-se, na solução proposta, os seguintes: *ccnputfile*, *ccngetfile*, *ccnputmeta* e *ccngetmeta*, listados na Tabela 4.4. *Ccnputfile* permite a publicação de um conteúdo associado a uma URI responsável por identificar o conteúdo na rede ICN, informados a localização do conteúdo na máquina local e a URI do conteúdo. *Ccnputmeta* publica um arquivo de metadados (na forma de um outro objeto CCNx) associado a um conteúdo disponível na rede ICN, dados a URI do conteúdo, o nome que o arquivo de metadados possuirá na rede e a localização do arquivo de metadados na máquina local. *Ccngetfile* recupera um conteúdo publicado na rede ICN, informados a URI do conteúdo que o identifica na rede e a localização de onde será salvo o conteúdo. O último método, *ccngetmeta*, recupera o arquivo de metadados associado a um determinado conteúdo, dados a URI do conteúdo, o nome do arquivo de metadados a ser recuperado e a localização de onde será armazenado o arquivo de metadados. Após apresentar os métodos em que o protótipo se baseia, aborda-se, a seguir, como a solução proposta foi materializada.

Tabela 4.4: Principais métodos CCNx empregados.

Nome	Descrição	Entrada	Saída
ccnputfile	Publica um arquivo como um conteúdo CCNx	URI e localização onde o arquivo está na máquina local	
ccngetfile	Recupera um conteúdo publicado e o salva como um arquivo local	URI e localização onde o conteúdo será salvo	Conteúdo (trata-se de um objeto recuperado da rede)
ccnputmeta	Associa um arquivo de metadados a um conteúdo CCNx	URI do conteúdo, nome do arquivo de metadados e localização onde o arquivo de metadados está na máquina local	
ccngetmeta	Recupera um arquivo de metadados associados a um conteúdo CCNx	URI do conteúdo, nome do arquivo de metadados e localização onde o arquivo de metadados será salvo na máquina local	Metadados (trata-se de outro objeto recuperado da rede)

Fonte: do autor (2016).

4.2 Implementação da Proposta

A proposta foi implementada no âmbito do ambiente CCNx lançando mão dos diferentes métodos introduzidos na seção anterior. Nesta seção, para cada operação importante previamente mencionada no Capítulo 3, realiza-se a descrição dos procedimentos desenvolvidos.

Criação de grupo. Como mencionado na Seção 3.1, esta operação consiste na criação do par de chaves do grupo e na propagação da chave pública na rede ICN, sendo disparada pelo administrador do grupo. Para realizar essa operação, sintetizada no Algoritmo 1, implementou-se um procedimento que recebe como entrada URI da chave pública do grupo, que identificará a chave na rede ICN, onde devem ser armazenadas, na máquina local, as chaves pública K_G e mestra M_G do grupo geradas e URI da lista de atributos. Finalizada a operação, retorna-se as chaves para o administrador do grupo na localização informada e publica-se a chave pública K_G na rede ICN.

Algoritmo 1: CRIAÇÃO DE UM GRUPO.

Entrada: $caminhoMaquinaLocalChavePublicaDoGrupo$, $caminhoMaquinaLocalChaveMestraDoGrupo$, $uriChavePublicaDoGrupo$, $uriListaAtributos$

Saída: $ChavePublicaDoGrupo$, $ChaveMestraDoGrupo$

1 **início**
2 $cpabe - setup$ –p $caminhoMaquinaLocalChavePublicaDoGrupo$ -m $caminhoMaquinaLocalChaveMestraDoGrupo$
3 $ccnputfile\ uriChavePublicaDoGrupo\ caminhoMaquinaLocalChavePublicaDoGrupo$
4 $arquivoBaseAtributos \leftarrow$ criaArquivoAtributos()
5 $ccnputfile\ uriListaAtributos\ arquivoBaseAtributos$
6 **fim**

Adição de usuário a um grupo. Esta operação corresponde à criação de uma chave privada de um usuário no grupo e a sua divulgação de modo seguro na rede ICN, operação essa executada pelo administrador do grupo. Para realizar essa operação, re-

sumida no Algoritmo 2, implementou-se um procedimento que recebe como entrada o caminho da chave pública K_G e mestra M_G do grupo, a URI da chave pública do grupo (lembrando que o CCNx emprega o esquema de nomeação hierárquica), URI da lista de atributos e a URI da chave privada do usuário no grupo. Ressalta-se que a recuperação da chave pública do usuário na linha 3, por questões de segurança, deve ser recuperada tendo como fonte uma PKI que empregue nomes amigáveis (por exemplo, SDSI/SPKI) (HALPERN; MEYDEN, 2000), como descrito em (SMETTERS; JACOBSON, 2009). Terminada a operação, publica-se a chave privada do usuário no grupo criptografada $\{K^{-1}_{\mathcal{L}_{U,G}}\}_{K_U}$ na URI informada.

Algoritmo 2: ADIÇÃO DE UM USUÁRIO.

Entrada: $uriChavePublicaUsuario$, $uriChavePrivadaUsuarioNoGrupo$, $uriListaAtributos$, $caminhoMaquinaLocalChavePublicaDoGrupo$, $caminhoMaquinaLocalChaveMestraDoGrupo$, $atributos$
Saída: -

1 **início**
2 inicializaConstantes($caminhoMaquinaLocalChavePrivadaUsuarioNoGrupo$, $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupoCriptografada$, $caminhoMaquinaLocalChavePublicaUsuario$)
/* Inicio - Recupera a chave publica do usuario */
3 $ccngetfile\ uriChavePublicaUsuario\ caminhoMaquinaLocalChavePublicaUsuario$
/* Fim - Recupera a chave publica do usuario */
4 $arquivoBaseAtributos \leftarrow$ abreArquivoAtributos()
5 **se** *not* $arquivoBaseAtributos$ *contém* $atributos$ **então**
6 concatena($arquivoBaseAtributos$, $atributos$)
7 $ccnputfile\ uriListaAtributos\ arquivoBaseAtributos$
8 **fim**
9 $atributos$ | cpabe-keygen -o $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupo$ $caminhoMaquinaLocalChavePublicaDoGrupo$ $caminhoMaquinaLocalChaveMestraDoGrupo$
10 $encryptFileRSA\ caminhoMaquinaLocalChavePrivadaUsuarioNoGrupoCriptografada$ $caminhoMaquinaLocalChavePublicaUsuario$ $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupo$
11 $ccnputfile\ uriChavePrivadaUsuarioNoGrupo$ $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupoCriptografada$
12 remove($caminhoMaquinaLocalChavePrivadaUsuarioNoGrupo$, $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupoCriptografada$, $caminhoMaquinaLocalChavePublicaUsuario$)
13 **fim**

Concessão de acesso. A operação de concessão de acesso consiste em definir uma política que um determinado conjunto de usuários do grupo deve satisfazer para descriptografar um conteúdo. Para executar essa operação, ilustrada no Algoritmo 3, informa-se a URI do Bloco Habilitador, a URI da chave pública do grupo, a localização da chave pública K_G, a localização da chave simétrica K_S e a política empregada P.

Finalizada a operação, disponibiliza-se o bloco habilitador H_C na rede ICN associado à URI informada.

Algoritmo 3: CONCESSÃO DE ACESSO A UM CONTEÚDO.

Entrada: $uriBlocoHabilitador$, $uriChavePublicaDoGrupo$, $caminhoMaquinaLocalChavePublica$, $caminhoMaquinaLocalChaveSimetrica$, $politica$
Saída: -
1 **início**
2 inicializaConstantes($caminhoMaquinaLocalArquivoMetadados$, $caminhoMaquinaLocalChaveSimetricaCriptografada$, $nomeMetadados$)
3 $politica$ | $cpabe - enc$ -k $caminhoMaquinaLocalChavePublica$ $caminhoMaquinaLocalChaveSimetrica$ -o $caminhoMaquinaLocalChaveSimetricaCriptografada$
4 $ccnputfile\ uriBlocoHabilitador\ caminhoMaquinaLocalChaveSimetricaCriptografada$
5 geraMetadados($caminhoMaquinaLocalArquivosMetadados$, $uriChavePublicaDoGrupo$)
6 $ccnputmeta\ uriBlocoHabilitador\ nomeMetadados\ caminhoMaquinaLocalArquivosMetadados$
7 remove($caminhoMaquinaLocalArquivosMetadados$, $caminhoMaquinaLocalChaveSimetricaCriptografada$)
8 **fim**

Publicação de conteúdos. A operação de publicação de conteúdos consiste em uma operação executada por um usuário. Esse método permite a publicação de um conteúdo protegido que apenas será acessado pelos usuários com os atributos necessários para descriptografar o bloco habilitador indicado. Para tal, executa-se o procedimento ilustrado no Algoritmo 4, informando a localização de um determinado conteúdo C, a URI que será associada ao conteúdo, a URI do bloco habilitador e o caminho da chave simétrica K_S correspondente ao bloco habilitador H_C indicado. Ao término da execução do procedimento, publica-se na rede um conteúdo protegido C_P na rede ICN associado a um bloco habilitador H_C.

Algoritmo 4: PUBLICAÇÃO DE UM CONTEÚDO.

Entrada: $uriConteudo$, $uriBlocoHabilitador$, $caminhoMaquinaLocalChaveSimetrica$, $caminhoMaquinaLocalConteudo$
Saída: -
1 **início**
2 inicializaConstantes($localizacaoMetadados$, $metadadosNome$, $conteudoCriptografado$)
3 $encryptFileAES\ conteudoCriptografado\ caminhoMaquinaLocalChaveSimetrica\ caminhoMaquinaLocalConteudo$
4 $ccnputfile\ uriConteudo\ conteudoCriptografado$
5 geraMetadados($localizacaoMetadados$, $uriBlocoHabilitador$)
6 $ccnputmeta\ uriConteudo\ metadadosNome\ localizacaoMetadados$
7 **fim**

Recuperação de conteúdos. A recuperação de conteúdos corresponde a uma operação disparada por um usuário do grupo. Essa operação permite a recuperação de

um conteúdo protegido C_P e a descriptografia do conteúdo C, que apenas será possível caso a chave privada do usuário no grupo $K^{-1}_{\mathcal{L}_{U,G}}$ atenda a política P estabelecida no bloco habilitador H_C. Para executar o procedimento correspondente, apresentado no Algoritmo 5, aponta-se a URI do conteúdo protegido C_P a ser recuperado na rede ICN (recordando que o CCNx encaminha os conteúdos pelo caminho inverso), o local no qual o conteúdo C deve ser armazenado e a localização da chave privada do usuário no grupo $K^{-1}_{\mathcal{L}_{U,G}}$. Após encerrado o procedimento, o usuário autorizado possuirá acesso ao conteúdo C na localização indicada.

Algoritmo 5: RECUPERAÇÃO DE UM CONTEÚDO.

Entrada: $uriConteudo$, $destinoConteudoNaMaquinaLocal$, $localizacaoChavePrivadaCpabeNaMaquinaLocal$
Saída: $Conteudo$
1 **início**
2 inicializaConstantes($localizacaoChavePublicaCpabeNaMaquinaLocal$, $localizacaoArquivoMetadadosNaMaquinaLocal$, $nomeMetadados$, $localizacaoChaveSimetricaNaMaquinaLocal$, $localizacaoChaveSimetricaCriptografadaNaMaquinaLocal$, $localizacaoConteudoCriptografadoNaMaquinaLocal$, $nomeMetadadosChaveSimetrica$, $localizacaoChavePublicaNaMaquinaLocal$)
3 $ccngetfile\ uriConteudo\ localizacaoConteudoCriptografadoNaMaquinaLocal$
4 $ccngetmeta\ uriConteudo\ nomeMetadados\ nomeMetadadosChaveSimetrica$
5 $uriBlocoHabilitador \leftarrow$ obtemValorMetadados($localizacaoArquivoMetadadoNaMaquinaLocal$, "<blocoHabilitador>")
6 $ccngetfile\ uriBlocoHabilitador\ localizacaoChaveSimetricaCriptografadaNaMaquinaLocal$
7 $ccngetmeta\ uriBlocoHabilitador\ nomeMetadados\ localizacaoArquivoMetadadosNaMaquinaLocal$
8 $uriChavePublica \leftarrow$ obtemValorMetadados($localizacaoArquivoMetadadosNaMaquinaLocal$, "<chavePublica>")
9 $ccngetfile\ uriChavePublica\ localizacaoChavePublicaNaMaquinaLocal$
10 $cpabe - dec\ localizacaoChavePublicaCpabeNaMaquinaLocal\ localizacaoChavePrivadaCpabeNaMaquinaLocal\ localizacaoChaveSimetricaCriptografadaNaMaquinaLocal$ -o $localizacaoChaveSimetricaNaMaquinaLocal$
11 $decryptFileAES\ localizacaoConteudoCriptografadoNaMaquinaLocal\ localizacaoChaveSimetricaNaMaquinaLocal\ destinoConteudoNaMaquinaLocal$
12 remove($localizacaoChaveSimetricaNaMaquinaLocal$, $localizacaoChaveSimetricaCriptografadaNaMaquinaLocal$)
13 **fim**

Revogação de acesso. Essa operação baseia-se em duas possibilidades: publicar uma nova versão de um bloco habilitador ou expirar as chaves dos usuários. A primeira possibilidade consiste no usuário dono de um conteúdo publicar uma nova versão de um determinado bloco habilitador H_C relacionado a um conteúdo C. Com isso, o usuário publicador poderia reformular a política P vinculada a esse bloco habilitador

H_C, excluindo determinados usuários $U_1, ..., U_N$ associados a um conjunto de atributos $\mathcal{L}_{U_1,...,U_N,G} \subseteq \mathcal{L}_G$. Essa opção de revogação é sumarizada no Algoritmo 3. A segunda opção de revogação de acesso consiste em convencionar que as chaves privadas dos membros do grupo expiram e estão associadas a um *timestamp* (TANENBAUM; BOS, 2014). Implementa-se essa funcionalidade modificando o procedimento de adição de usuários (Algoritmo 6), com o objetivo de guardar quais atributos $\mathcal{L}_{U,G} \subseteq \mathcal{L}_G$ foram utilizados para gerar a chave privada do usuário no grupo $K^{-1}_{\mathcal{L}_{U,G}}$. A nova operação de adição acrescenta um atributo *timestamp* para datar quando a chave privada do usuário no grupo $K^{-1}_{\mathcal{L}_{U,G}}$ foi criada. Além disso, toda vez que for publicado um conteúdo protegido C_P, o usuário publicador deve determinar um intervalo de tempo para restringir o acesso de usuários. Por fim, quando o administrador desejar renovar as chaves privadas de um conjunto de usuários no grupo, ele pode empregar o procedimento sumarizado no Algoritmo 7, com o objetivo de automatizar o processo de renovação das chaves, sendo uma operação suporte à técnica de expiração de chaves.

Algoritmo 6: ADIÇÃO DE UM USUÁRIO COM SUPORTE À REVOGAÇÃO.

Entrada: $uriChavePublicaUsuario$, $uriChavePrivadaUsuarioNoGrupo$, $uriListaAtributos$, $caminhoMaquinaLocalChavePublicaDoGrupo$, $caminhoMaquinaLocalChaveMestraDoGrupo$, $atributos$, $identificacaoGrupo$

Saída: -

1 **início**

2 inicializaConstantes($caminhoMaquinaLocalChavePrivadaUsuarioNoGrupo$, $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupoCriptografada$, $caminhoMaquinaLocalChavePublicaUsuario$)

`/* Inicio - Recupera a chave publica do usuario */`

3 $ccngetfile\ uriChavePublicaUsuario\ caminhoMaquinaLocalChavePublicaUsuario$

`/* Fim - Recupera a chave publica do usuario */`

4 $arquivoBaseAtributos \leftarrow$ abreArquivoAtributos()

5 **se** *not* $arquivoBaseAtributos$ *contém* $atributos$ **então**

6 concatena($arquivoBaseAtributos$, $atributos$)

7 $ccnputfile\ uriListaAtributos\ arquivoBaseAtributos$

8 **fim**

`/* inicio da modificacao */`

9 $timestamp \leftarrow$ obtemTimestamp()

10 $atributos \leftarrow$ concatena($atributos$, $timestamp$)

11 $identificacaoUsuario \leftarrow$ geraIdentificacaoUsuario()

12 $caminhoMaquinaLocalDoArquivo \leftarrow$ geraLocalizacaoDoArquivo($identificacaoUsuario$, $identificacaoGrupo$)

13 salvaAtributosDoUsuario($identificacaoUsuario$, $caminhoMaquinaLocalDoArquivo$, $uriChavePublicaUsuario$, $uriChavePrivadaUsuarioNoGrupo$)

`/* fim da modificacao */`

14 $atributos$ | $cpabe - keygen$ -o $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupo$ $caminhoMaquinaLocalChavePublicaDoGrupo$ $caminhoMaquinaLocalChaveMestraDoGrupo$

15 $encryptFileRSA\ caminhoMaquinaLocalChavePrivadaUsuarioNoGrupoCriptografada$ $caminhoMaquinaLocalChavePublicaUsuario$ $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupo$

16 $ccnputfile\ uriChavePrivadaUsuarioNoGrupo$ $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupoCriptografada$

17 remove($caminhoMaquinaLocalChavePrivadaUsuarioNoGrupo$, $caminhoMaquinaLocalChavePrivadaUsuarioNoGrupoCriptografada$, $caminhoMaquinaLocalChavePublicaUsuario$)

18 **fim**

Algoritmo 7: RENOVAÇÃO DAS CHAVES DOS USUÁRIOS DE UM GRUPO.

Entrada: $identificacaoGrupo$, $listaDeUsarios$
Saída: -
1 **início**
2 carregaInformacoesDoGrupo($identificacaoGrupo$)
3 **para** *cada* $usuario \in listaDeUsarios$ **faça**
4 carregaInformacaoUsuario($identificacaoGrupo$, $usuario$)
5 adicaoDeUmUsuarioSuporteRevogacao($uriChavePublicaUsuario$, $uriChavePrivadaUsuarioNoGrupo$, $uriListaAtributos$, $caminhoMaquinaLocalChavePublicaDoGrupo$, $caminhoMaquinaLocalChaveMestraDoGrupo$, $atributos$, $identificacaoGrupo$)
6 **fim**
7 **fim**

No capítulo seguinte, mostra-se o conjunto de experimentos realizados, com o intuito de compreender as sobrecargas adicionadas pela solução proposta quando comparada à disponibilização e à recuperação de conteúdos sem o emprego de mecanismos de controle de acesso. Além disso, compara-se a implementação ao modo clássico que poderia ser empregado, RSA, e ao trabalho de Papanis *et al.* (PAPANIS et al., 2014) em relação a um conjunto de métricas.

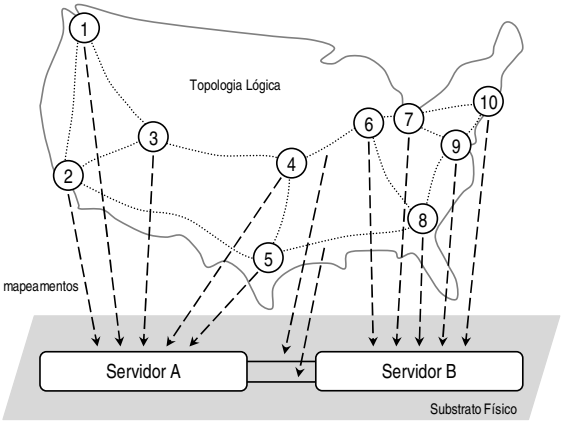
5 AVALIAÇÃO

Para aferir a eficácia e eficiência do protótipo implementado, uma série de experimentos foi realizada em um ambiente controlado, como demonstrado na Seção 5.1. Os experimentos visaram verificar o custo de operação (Seção 5.2), a quantidade de chaves e de objetos gerados (Seção 5.3), e a escalabilidade do modelo e o impacto à qualidade de experiência (QoE) dos usuários, em cenários com um número variado de usuários atuando como publicadores e recuperadores (Seção 5.4.). Para comparação, considerou-se a solução de Papanis *et al.* (PAPANIS et al., 2014) e uma solução de compartilhamento seguro de conteúdos baseada no algoritmo RSA.

5.1 Configuração do Ambiente e Cenários de Avaliação

O modelo proposto foi implementado sobre a arquitetura CCN (*Content Centric Networking*) (JACOBSON et al., 2012), usando como base o *software* CCNx 0.8.2 executando sobre a máquina virtual Java SE versão 8. Para o mecanismo de criptografia baseada em atributos, utilizou-se o *software cpabe* 0.11 (BETHENCOURT; SAHAI; WATERS, 2015). Visando a seguir o padrão para gerenciamento de chaves proposto para arquiteturas ICN (BIAN et al., 2013), cada conteúdo protegido é acompanhado de um respectivo arquivo de metadados, que contém o identificador do bloco habilitador correspondente e a validade do conteúdo, entre outros. Da mesma forma, cada bloco habilitador é acompanhado por um respectivo arquivo de metadados contendo a URI da chave pública do grupo.

Figura 5.1: Topologia base da avaliação.



Fonte: do autor (2016).

Tabela 5.1: Cenários avaliados.

Parâmetros	Cenários avaliados			
	A	B	C	D
Quantidade de usuários	2	2	10	30
Tamanho dos arquivos	1MB, 10MB, 100MB e 1000MB	1MB, 10MB, 100MB e 1000MB	100MB	100MB
Soluções testadas	Proposta	Proposta, Papanis e Plano	Proposta, Papanis, Plano e RSA	Proposta, Papanis, Plano e RSA
Criptografia dos conteúdos	AES-CBC 256	AES-CBC 256 e sem (plano)	AES-CBC 256 e sem (plano)	AES-CBC 256 e sem (plano)
Arquivos publicados	1	1	10	30
Capacidade da *cache*	1GB	1GB	1GB	1GB
Expiração da *cache*	1 hora	1 hora	1 hora	1 hora
Tamanho do *chunk*	4KB	4KB	4KB	4KB
Popularidade dos conteúdos	-	-	Zipf [1] ($s = 2.0$)	Zipf [1] ($s = 2.0$)

[1] Valor obtido conforme *Pentikousis et al.* (PENTIKOUSIS et al., 2015)

Fonte: do autor (2016).

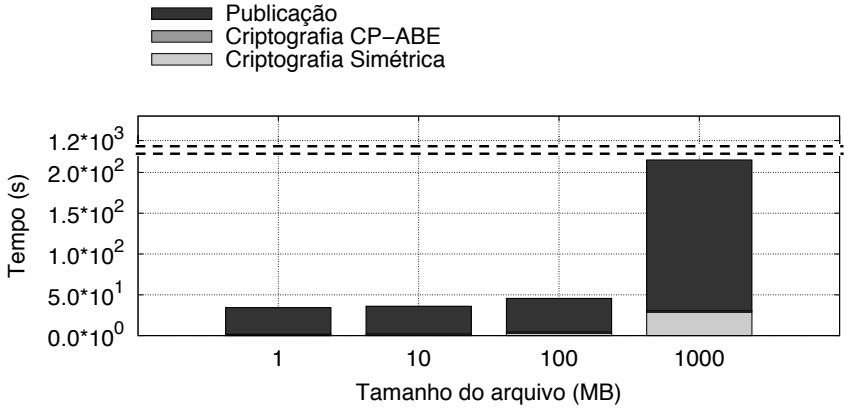
O substrato físico usado nos experimentos compreendeu dois servidores, cada um equipado com 1 processador Intel Xeon E5-2420 (1.9GHz, 12 Threads e 15MB *cache*), 32GB de memória RAM (1333MHz), 1 HD SAS (1TB de capacidade) e 2 interfaces de rede Gigabit Ethernet. Ambos possuem Debian/Linux 7.7 (kernel 3.14.21) e Hipervisor Xen instalados. Os servidores foram conectados diretamente entre si usando dois cabos Ethernet. A topologia lógica usada para a avaliação, um subconjunto da Internet2, é ilustrada na Figura 5.1. O mapeamento dos elementos lógicos para o substrato físico também é apresentado na figura. Cada nodo lógico na topologia corresponde a uma máquina virtual; cada máquina foi instanciada com as seguintes configurações: 2 processadores virtuais, 2GB de RAM e 40GB de disco. Os enlaces entre os nodos foram emulados empregando o *software bridge-utils* versão 1.5, todos com velocidade de $\approx$ 98Mbps.

Para a avaliação experimental foram considerados quatro cenários, cujos parâmetros mais relevantes são sumarizados na Tabela 5.1. Por simplificação, todos os conteúdos foram publicados usando uma política de acesso universal (ou seja, qualquer usuário membro do grupo pode decifrá-lo). Essa decisão foi embasada em experimentos preliminares, os quais permitiram observar que a quantidade de atributos não tem efeito relevante sobre custos (tempo de publicação/recuperação, tráfego de rede, etc.) do modelo proposto. Por fim, para cada experimento, foram realizadas 30 execuções e calculado o intervalo de confiança com nível de significância $\alpha = 0.05$. Detalhado o ambiente de experimentação utilizado, na próxima seção descreve-se os resultados obtidos com os experimentos relativos aos custos da solução.

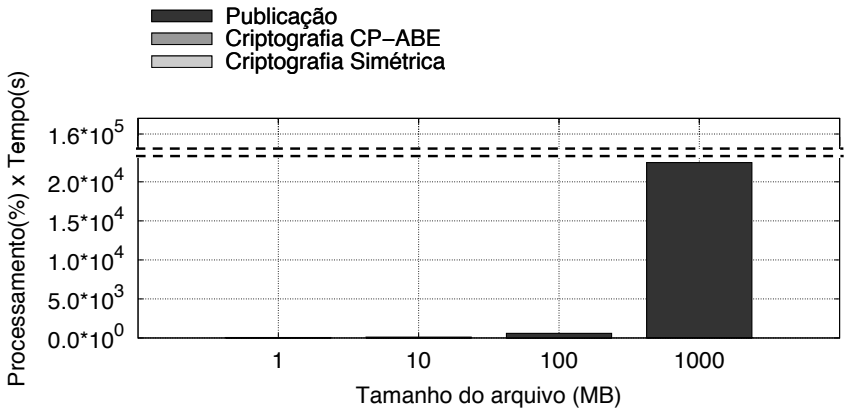
5.2 Custos da Solução Proposta

A primeira parte da avaliação compreende uma análise dos custos da solução em um ambiente isolado, formado por apenas um publicador e um recuperador. Para isso, utilizou-se subconjunto da topologia ilustrada na Figura 5.1, formada pelos nodos 4 (publicador) e 6 (recuperador). Essa avaliação desdobra-se em duas partes. A primeira aborda as sobrecargas adicionadas pelos mecanismos empregados pela solução proposta (Subseção 5.2.1). A segunda compara a solução proposta em relação ao estado da arte (Subseção 5.2.2).

Figura 5.2: Custos envolvidos na publicação.



(a) Tempo.



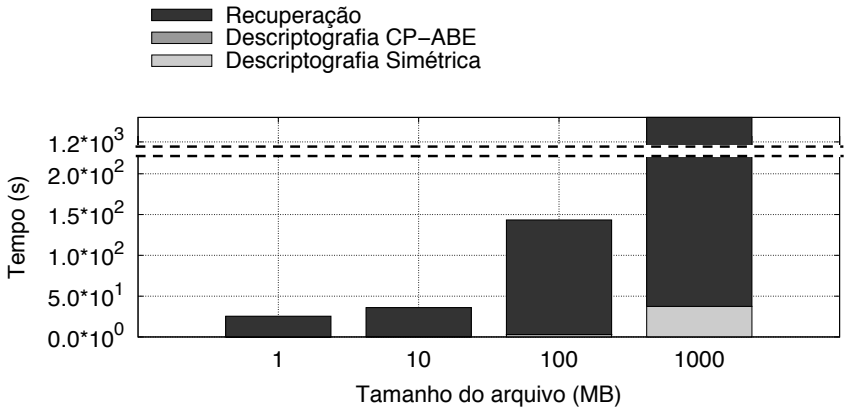
(b) Carga de processamento.

Fonte: do autor (2016).

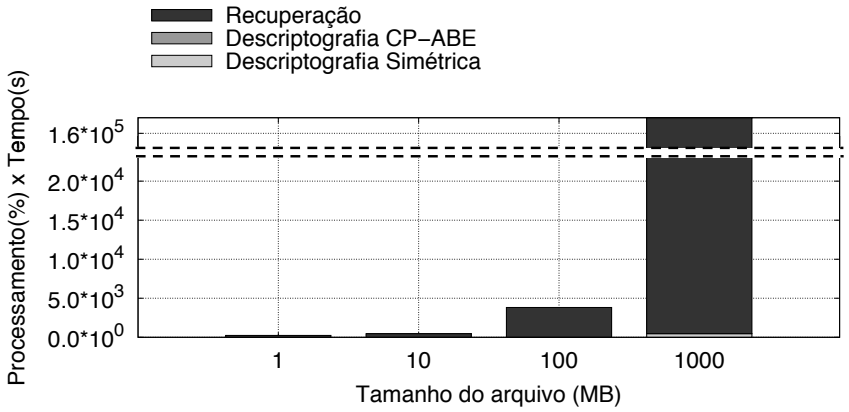
5.2.1 Sobrecarga dos Mecanismos Empregados

Na primeira parte da avaliação, foram analisados os custos das principais etapas de publicação e recuperação. Com esse objetivo, considerou-se as operações de publicação de conteúdo, criptografia CP-ABE, criptografia simétrica, recuperação de conteúdo, descriptografia CP-ABE e descriptografia simétrica. Ressalta-se que para essa parte da avaliação foi utilizado o cenário A resumido na Tabela 5.1. Por legibilidade, os gráficos são apresentados com o eixo y em escala linear. Vale destacar que os valores nas operações envolvendo arquivos de 1000MB geram valores relativamente superiores em relação aos outros. Por esse motivo, introduziu-se uma quebra nos gráficos com o intuito de possibilitar a visualização dos custos de publicação/recuperação de todos os conteúdos.

Figura 5.3: Custos envolvidos na recuperação.



(a) Tempo.



(b) Carga de processamento.

Fonte: do autor (2016).

A principal conclusão que se alcança, a partir dos resultados expostos pelas Figura 5.2 e Figura 5.3, é de que as operações de publicação e recuperação de um conteúdo na

rede ICN correspondem aos maiores custos quando comparados aos de criptografia e descriptografia. Analisando o tempo médio de disponibilização de um conteúdo protegido (Figura 5.2(a)), a operação de publicação corresponde a 86%, a de criptografia simétrica a 13% e a de criptografia CP-ABE a 1% na publicação de um conteúdo com 1000MB, sendo esse o caso em que as operações de criptografia possuem o maior impacto. Referente ao tempo médio de recuperação de um conteúdo protegido (Figura 5.3(a)), a operação de recuperação corresponde a 97%, a de descriptografia simétrica a 3% e a de descriptografia CP-ABE é desprezível na recuperação de um conteúdo com 1000MB, sendo esse o caso em que as operações de descriptografia possuem a maior influência. Relativo aos custos de processamento na publicação e recuperação de um conteúdo protegido (Figura 5.2(b) e Figura 5.3(b)), a operação de publicação corresponde a quase 100% dos custos, sendo marginais os custos das operações de criptografia/descriptografia simétrica e CP-ABE. É importante mencionar que outros custos, tais como de inicialização do grupo e de geração das chaves privadas dos usuários no grupo, são negligenciáveis. Expostos os custos relativos a cada etapa de publicação/recuperação de um conteúdo protegido, realiza-se uma comparação entre a solução proposta em relação ao estado da arte.

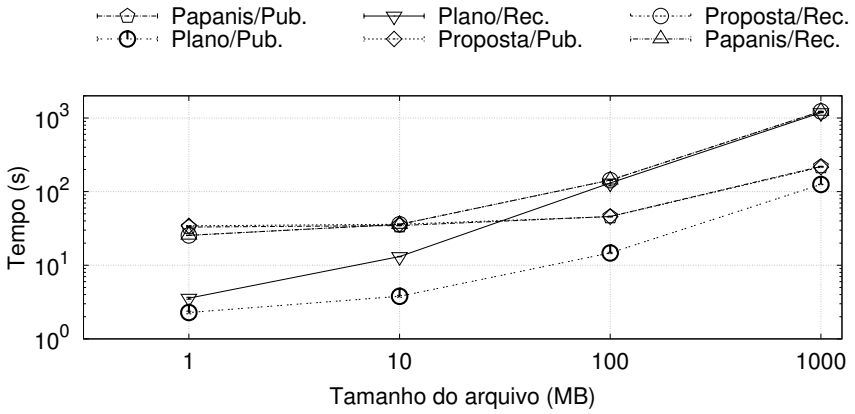
5.2.2 Análise Comparativa da Solução em Relação ao Estado da Arte

Na segunda parte da avaliação de custos, compara-se a solução proposta com a solução de Papanis *et al.* e com a publicação/recuperação de conteúdos sem o emprego de qualquer mecanismo de segurança ("Plano"). Vale a pena destacar que essa avaliação empregou o cenário B resumido, na Tabela 5.1. Esse cenário é idêntico ao A, exceto pela comparação com duas outras propostas já referidas. A solução de Papanis *et al.* (PAPANIS et al., 2014) consiste em uma abordagem que também emprega a técnica CP-ABE, como já discutido em 2.3, e a solução "Plano" serve de base para obter-se os sobrecustos introduzidos pelas outras soluções, uma vez que não utiliza mecanismos de controle de acesso. A Figura 5.4 apresenta uma visão geral dos resultados obtidos para a publicação (curvas "Pub.") e a recuperação (curvas "Rec.") de conteúdos. Por legibilidade, os gráficos são apresentados com o eixo y em escala logarítmica.

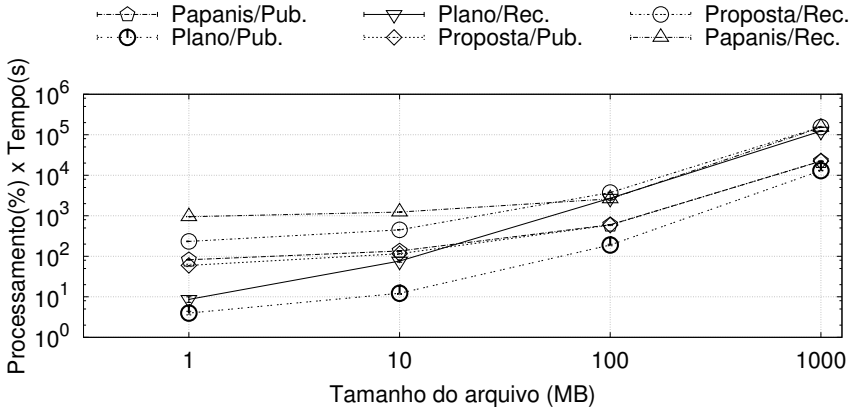
A principal conclusão que se pode tirar, a partir dos resultados da Figura 5.4, é a de que o sobrecusto da solução proposta é marginal quando comparado a Papanis *et al.* Focando no tempo médio de disseminação de conteúdos (Figura 5.4(a)), por exemplo, a solução proposta foi inclusive 0,6% mais eficiente em média na publicação. Em relação

aos custos de processamento (Figura 5.4(b)), esses são ligeiramente maiores na solução proposta (0,5% na publicação e 1,4% na recuperação). Por fim, observa-se que a medição do tráfego gerado na rede (Figura 5.4(c)) indica desempenhos similares de ambas as soluções.

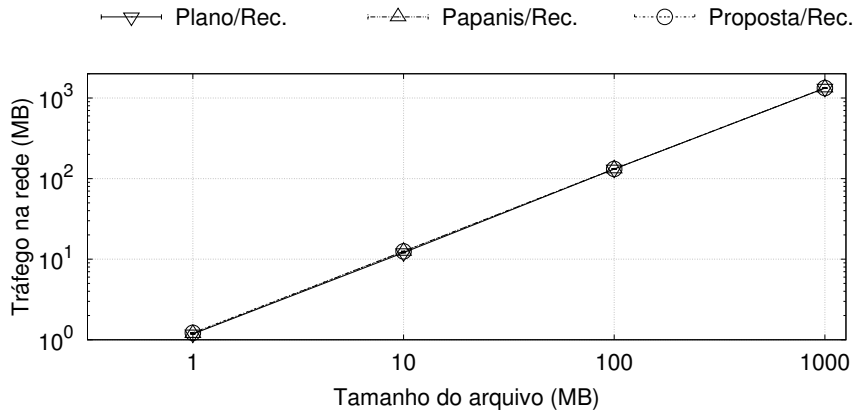
Figura 5.4: Custos da solução proposta em termos de tempo de publicação/recuperação, carga de processamento e tráfego gerado.



(a) Tempo.



(b) Carga de processamento.



(c) Tráfego na rede.

Fonte: do autor (2016).

Quando comparados à solução sem mecanismos de segurança, note que os custos são amortizados de forma proporcional ao tamanho do conteúdo publicado. Esses resultados sugerem que a solução proposta incorre em impacto relativamente pequeno para a qualidade de experiência (QoE) dos usuários. Na publicação, por exemplo, o sobrecusto de tempo diminui de 1400% (diferença do custo entre a solução proposta e a solução "Plano"), em média (com conteúdos de 1MB), para 72% (1000MB). Nessa comparação, o sobrecusto médio foi de apenas 8% no tempo de recuperação de conteúdos (aspecto de maior importância para a QoE de grande parte dos usuários). Os resultados obtidos para o modelo proposto são inclusive similares ao observado para Papanis *et al.*

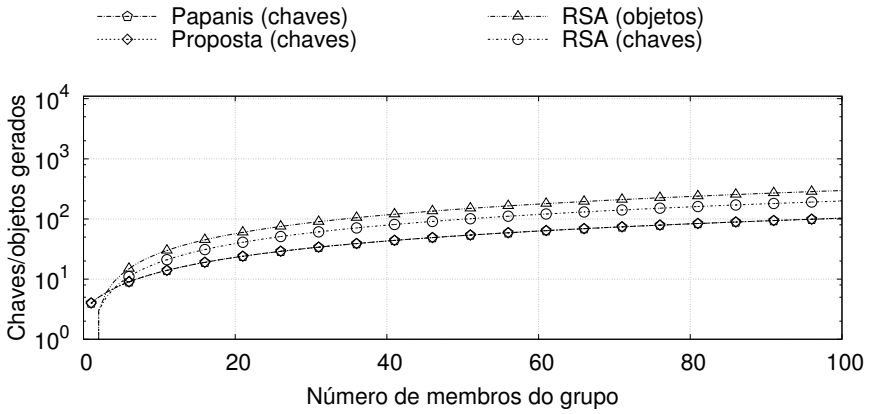
É importante mencionar que os custos adicionais de processamento e de tempo devem-se ao uso de criptografia para cifrar/decifrar o conteúdo e as chaves de acesso. Ressalta-se que a proposta utiliza uma técnica de criptografia consideravelmente segura e que os sobrecustos relativos à criptografia poderiam ser amortizados empregando variações mais leves de algoritmos criptográficos, como 3DES, AES-128 CBC e AES-192 CBC. Quando não há solução de segurança sendo utilizada, o processamento e o tempo referem-se apenas à publicação/recuperação do conteúdo na rede. Sobre o tráfego gerado na rede (Figura 5.4(c)), o sobrecusto foi constante e marginal, correspondendo principalmente ao bloco habilitador do conteúdo protegido (que também é disseminado na rede). Discutido os sobrecustos introduzidos pelos mecanismos empregados na solução proposta e realizada uma análise comparativa em relação ao estado da arte, a seguir (Seção 5.3), analisa-se a quantidade de chaves e objetos requeridos por cada proposta.

5.3 Quantidade de Chaves e de Objetos

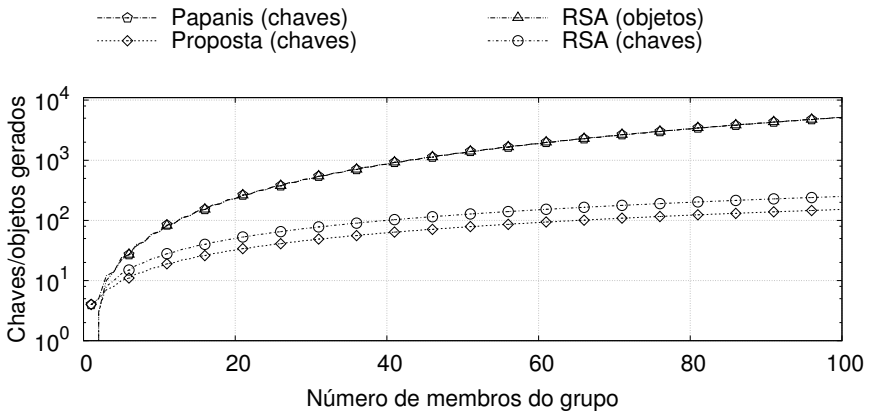
Outro aspecto observado na avaliação está relacionado à quantidade de chaves/objetos necessários para a disseminação segura de conteúdos, nas situações em que um, metade e todos os usuários atuam como publicadores na rede, respectivamente. Nas duas análises expostas, além da solução proposta, foram consideradas a de Papanis *et al.* (PAPANIS et al., 2014) e a baseada no algoritmo RSA. No caso de Papanis *et al.*, ela é instanciada para cada usuário publicador. No modelo de segurança baseado no RSA, (*i*) cada usuário possui um par de chaves pública e privada; (*ii*) cada conteúdo é cifrado usando uma chave simétrica única; e (*iii*) a chave do conteúdo é cifrada usando a chave pública de cada usuário alvo. A primeira análise corresponde à quantidade de chaves/objetos gerados com o intuito de possibilitar a publicação/recuperação de conteúdos por todos os

usuários. A segunda análise, por sua vez, refere-se à quantidade de chaves que todos os usuários precisam solicitar relativas ao seu grupo.

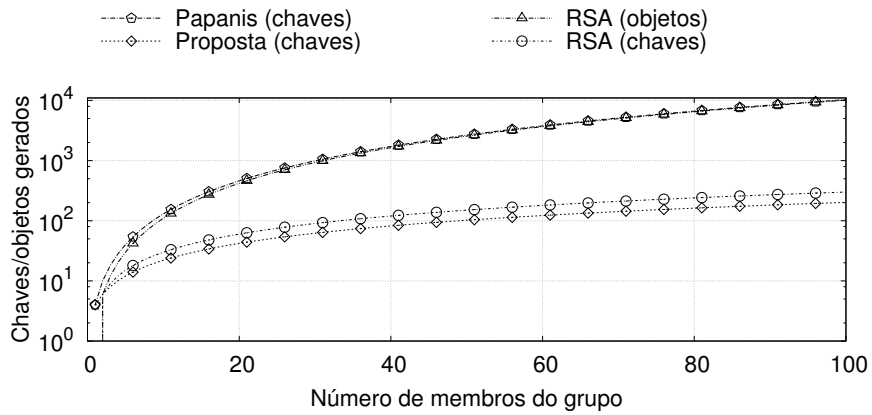
Figura 5.5: Número de chaves/objetos necessários para a troca de conteúdos.



(a) 1 publicador.



(b) $n/2$ publicadores.



(c) n publicadores.

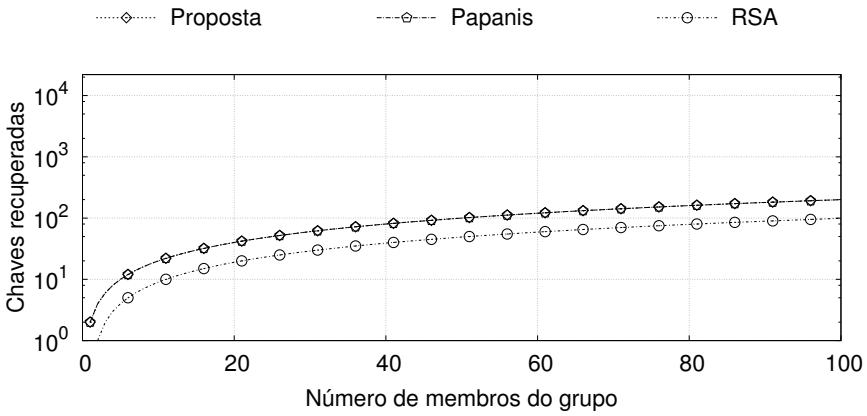
Fonte: do autor (2016).

Os resultados da primeira avaliação são apresentados na Figura 5.5 (o eixo y é apresentado em escala logarítmica). Observa-se que a solução proposta requer o menor

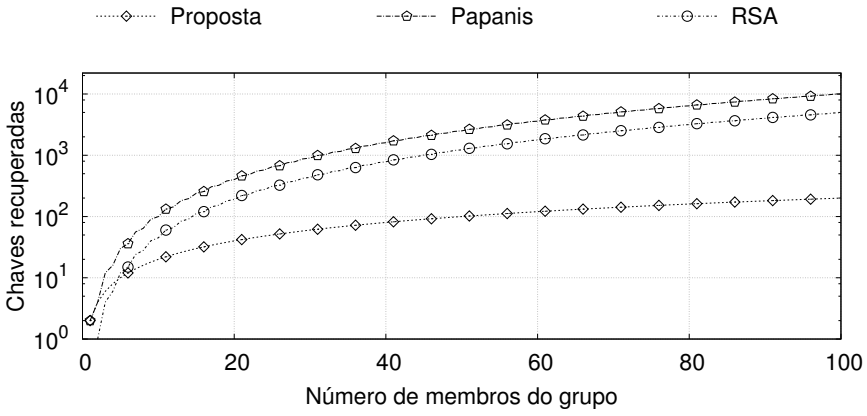
número de chaves criptográficas, em comparação com Papanis *et al.* e RSA. Além disso, a solução proposta mantém a proporcionalidade do número das chaves relativas ao número de usuários pertencentes ao grupo, independentemente do número de publicadores. Mais importante, o número de chaves necessárias/objetos publicados aumenta significativamente para as duas últimas. Para Papanis *et al.*, observa-se um aumento de até 9900.0%, em contraste com 96% na solução proposta. Em relação a Papanis *et al.*, esse aumento se relaciona ao problema da explosão combinatória de chaves (conforme discutido na Subseção 2.3). Embora no cenário usando RSA o número de chaves permaneça relativamente constante, o número de objetos publicados na rede cresce significativamente. O motivo é que, embora a chave simétrica seja única para cada conteúdo, ela precisa ser criptografada individualmente para cada usuário alvo, de modo a garantir que apenas usuários autorizados possam acessar o conteúdo.

Na Figura 5.6, ilustra-se os resultados da segunda avaliação, que refere-se à quantidade de chaves recuperadas por todos usuários relativas ao grupo (apresenta-se o eixo y em escala logarítmica). A solução proposta exige que os usuários recuperem o menor número de chaves criptográficas, em relação com Papanis *et al.* e RSA, exceto no caso onde há apenas uma fonte publicadora. Em relação ao número de chaves recuperadas, há uma diferença significativa para as outras soluções. Para Papanis *et al.*, observa-se o aumento de até 9900% (de uma para n fontes, onde n igual 100), enquanto na solução proposta não há diferença no número de chaves recuperadas pelos usuários independente do número de publicadores. Como recém mencionado, em relação a Papanis *et al.*, esse aumento relaciona-se ao problema da explosão combinatória de chaves (conforme discutido na Subseção 2.3). O mesmo ocorre no cenário RSA, onde também constata-se um aumento de 9900% variando o número de fonte 1 para n, onde n igual 100.

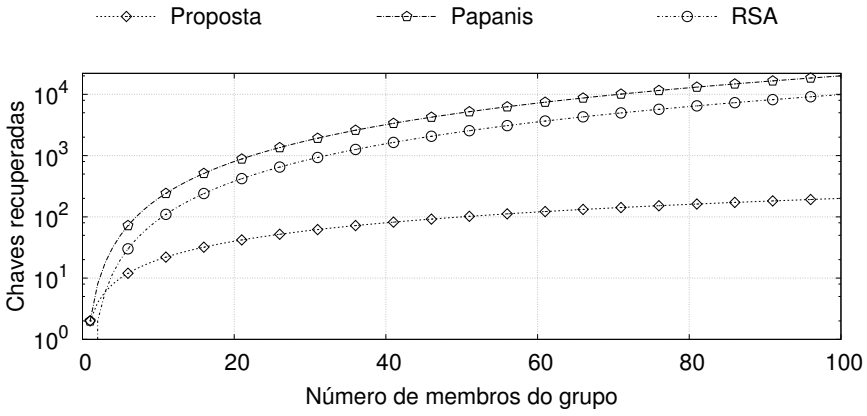
Figura 5.6: Quantidade de chaves recuperadas.



(a) 1 fonte.



(b) $N/2$ fontes.



(c) N fontes.

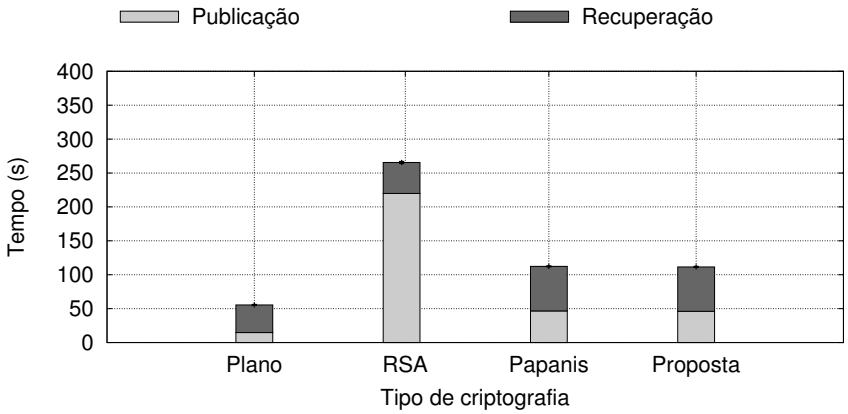
Fonte: do autor (2016).

Explorada a sobrecarga gerada pelas soluções na quantidade de chaves e objetos necessários para a publicação/recuperação, aborda-se, a seguir, o impacto causado na utilização de cada solução na FIB dos roteadores. Também, discute-se os tempos de disseminação de conteúdos sobre uma rede que emula um subconjunto da Internet2.

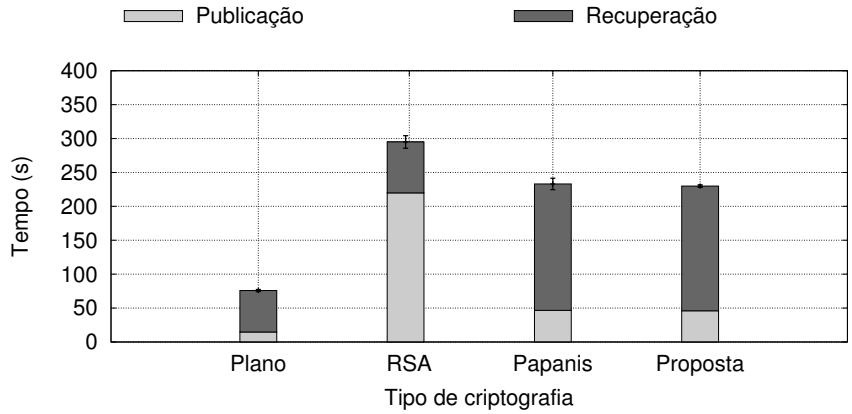
5.4 Tempos de Disseminação e Quantidade de Objetos Registrados

Esta parte da avaliação tem como objetivo, aferir o desempenho da solução proposta - mais especificamente o tempo para disseminação de conteúdos e a sobrecarga gerada à *Forward Information Base* (FIB) dos roteadores - em um ambiente com múltiplos publicadores e consumidores. Vale ressaltar que essa avaliação usou como base a topologia completa ilustrada na Figura 5.1 e os cenários C e D, sumarizados na Tabela 5.1. Cada usuário publica 1 e recupera n conteúdos, ou seja, são publicados 10 conteúdos no cenário C e 30 no cenário D. Para comparação, foram empregadas a solução de Papanis *et al.*, uma baseada no algoritmo RSA e outra sem mecanismos de segurança ("Plano").

Figura 5.7: Tempo de publicação/recuperação dos conteúdos.



(a) Tempos medidos no cenário C.



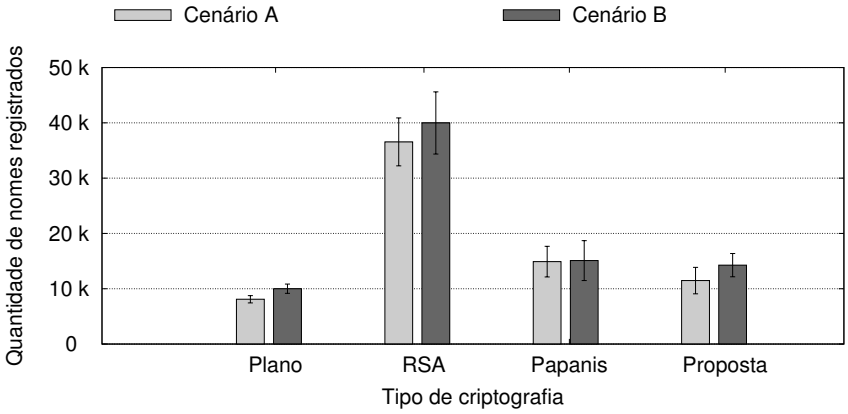
(b) Tempos medidos no cenário D.

Fonte: do autor (2016).

Observa-se, nas Figuras 5.7(a) e 5.7(b), que a qualidade de experiência (QoE) do usuário (medida pelo tempo necessário para disseminação do conteúdo) é marginalmente afetada na solução proposta, se comparada a de Papanis *et al.* Esse desempenho é al-

cançado causando relativamente menos impacto à rede, conforme pode ser observado na Figura 5.8.

Figura 5.8: Quantidade de nomes registrados na FIB.



Fonte: do autor (2016).

O tempo relativamente maior de recuperação nas soluções proposta e a de Papanis *et al.* explica-se pelo fato de que a solução baseada no RSA não requer arquivo de metadados de conteúdo ou de bloco habilitador. Em outras palavras, cada usuário pode localizar diretamente os conteúdos e suas respectivas chaves sem a necessidade de obter o arquivo de metadados dos mesmos, sendo, portanto, irrelevante publicá-los. Por outro lado, o tempo de publicação é significativamente maior no RSA, visto que n versões criptografadas da chave de um mesmo conteúdo devem ser publicadas na rede, uma para cada usuário alvo. Essa característica se reflete no gráfico da Figura 5.8, com a solução proposta reduzindo em até 68% (no cenário D) a quantidade de nomes registrados na FIB dos roteadores.

6 CONCLUSÃO

A publicação segura de conteúdos em ICN é uma realidade, com diversas soluções que oferecem os mais variados níveis de controle de acesso. Apesar de promissoras, algumas soluções causam uma sobrecarga significativa na rede ao tornar o processo de gerência (e de distribuição) de chaves combinatorialmente complexo. As soluções que não estão suscetíveis a esse problema, no entanto, são dependentes de arquitetura específica ICN, inserem (ou modificam) componentes na rede e são pouco flexíveis para adoção gradual.

Com o intuito de suprir essa lacuna, foi apresentada uma nova solução, centrada nos conceitos de grupos de usuários, para o compartilhamento seguro de conteúdos. A partir dos resultados alcançados, foi possível aferir a eficácia e eficiência da solução proposta. Em resumo, esta requer um número comparativamente menor de chaves e objetos na rede (em alguns casos até 97% menos chaves). Esse ganho é alcançado sem degradar a qualidade de experiência do usuário (por exemplo, o tempo necessário para publicar/recuperar conteúdos), ao contrário do que ocorre em outras soluções. Além desses benefícios, a solução proposta pode ser adotada de forma independente e autônoma por um subconjunto de usuários, sem depender de modificações na rede. Por fim, ela permite a publicação e recuperação de conteúdos mesmo que o administrador do grupo (ou o publicador do conteúdo, no caso de recuperação) torne-se indisponível.

Com base nos resultados expostos e retomando o estudo de caso mencionado na introdução (cenário de *User-Generated Content*), considera-se para o caso analisado que a solução proposta obteve um desempenho satisfatório. Além disso, a proposta permite que todas as funcionalidades desejadas sejam executadas, dentro do contexto de grupos com muitos publicadores/recuperadores que disponibilizam arquivos de tamanho considerável como, por exemplo, arquivos de vídeo.

Alguns aspectos pertinentes à viabilidade técnica da solução merecem ser discutidos. Um desses aspectos consiste na atualização de uma política de acesso de um dado conteúdo. Quando executa-se a operação de atualização de uma política de acesso em uma rede ICN, por um dado momento pode-se ter um determinado conteúdo criptografado protegido/restrito por duas ou mais políticas de acesso. Esse fato ocorre, pois mesmo que a rede ICN empregue um mecanismo de versionamento de conteúdos, lembrando que a própria política também corresponde a um conteúdo, a nova política leva um tempo para ser propagada na rede ICN. Apesar de ser uma limitação reconhecida na solução proposta,

ela não é resolvida nessa primeira interação do trabalho. É válido mencionar que outras soluções de controle em ICN padecem da mesma limitação. Por esse motivo, o problema referente à atualização de uma política de acesso merece ser investigado futuramente.

Outro aspecto reside em um publicador de conteúdos não desejar mais que um dado conteúdo seja acessível aos usuários de seu grupo. Para realizar esse procedimento, o publicador remove o conteúdo de seu repositório local, impedindo a sua disseminação a partir do repositório fonte. Contudo, caso haja cópias desse conteúdo nas *caches* dos roteadores da rede ICN, o mesmo não será removido de imediato da rede. A ação de remoção de conteúdo apenas possuirá efeito quando o conteúdo em questão tiver expirado em todas as *caches* que o possuem. Isso ocorre porque o publicador do conteúdo não possui nenhum controle sobre os roteadores pertencentes à rede ICN. Essa característica infringe a noção tradicional de controle de acesso, sendo esse um problema em aberto em ICN. O problema de remoção de conteúdos é um tema particularmente relevante de ser pesquisado e também merecedor de investigações futuras.

REFERÊNCIAS

AHLGREN, B. et al. A survey of information-centric networking. **IEEE Communications Magazine**, v. 50, n. 7, p. 26–36, 2012. ISSN 0163-6804.

ATENIESE, G. et al. Improved proxy re-encryption schemes with applications to secure distributed storage. **ACM Transactions on Information and System Security**, ACM, New York, NY, USA, v. 9, n. 1, p. 1–30, fev. 2006. ISSN 1094-9224. Disponível em: <http://doi.acm.org/10.1145/1127345.1127346>.

BETHENCOURT, J.; SAHAI, A.; WATERS, B. Ciphertext-Policy Attribute-Based Encryption. In: IEEE SYMPOSIUM ON SECURITY AND PRIVACY (SP 2007), 28., 2007, Oakland, California, USA. **Proceedings...** The Claremont Resort, Oakland, California, USA: IEEE, 2007. p. 321–334. ISBN 0-7695-2848-1. Disponível em: <https://www.cs.utexas.edu/~bwaters/publications/papers/cp-abe.pdf>.

BETHENCOURT, J.; SAHAI, A.; WATERS, B. **Advanced Crypto Software Collection**. 2015. Disponível em: <http://acsc.cs.utexas.edu/cpabe/>. Acesso em: Abril de 2015. Disponível em: <http://acsc.cs.utexas.edu/cpabe/>.

BIAN, C. et al. **Deploying key management on NDN testbed**. [S.l.], 2013. 1-4 p. Disponível em: <http://www.named-data.net/techreport/TR009-publishkey-rev2.pdf>. Acesso em: Junho, 2015.

BISHOP, M. **Introduction to Computer Security**. [S.l.]: Addison-Wesley Professional, 2004. ISBN 0321247442.

BONEH, D.; SAHAI, A.; WATERS, B. Functional encryption: A new vision for public-key cryptography. **Communications of the ACM**, v. 55, p. 56–64, 2012. ISSN 0001-0782.

BRITO, G. M. de; VELLOSO, P. B.; MORAES, I. M. Redes orientadas a conteúdo: Um novo paradigma para a internet. In: **Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2012)**. 1. ed. Porto Alegre, RS, Brasil: SBC, 2012. cap. 5, p. 211–264.

CHIOCCHETTI, R.; ROSSI, D.; ROSSINI, G. ccnSim: An highly scalable CCN simulator. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC), 2013, 12., 2013, Budapest, Hungary. **Proceedings...** IEEE, 2013. p. 2309–2314. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6654874>.

CISCO. **Forecast and Methodology, 2014-2019 White Paper**. 2015. URL: http://httpd.apache.org/. Acesso em: Outubro de 2015. Disponível em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html>.

CODIO, S. **Ciphertext-Policy Attribute-Based Encryption (CP-ABE)**. 2011. University Lecture. URL: http://courses.cs.vt.edu/cs6204/Privacy-Security/Presentations/CP-ABE.pdf. Acesso em: Março de 2015. Disponível em: <http://courses.cs.vt.edu/cs6204/Privacy-Security/Presentations/CP-ABE.pdf>.

COLTUN, R. et al. **OSPF for IPv6**. IETF, 2008. RFC 5340 (Proposed Standard). (Request for Comments, 5340). Http://www.ietf.org/rfc/rfc5340.txt. Acesso em: Março de 2015. Disponível em: <http://www.ietf.org/rfc/rfc5340.txt>.

ELAYOUBI, S.-E.; ROBERTS, J. Performance and Cost Effectiveness of Caching in Mobile Access Networks. In: THE 2ND INTERNATIONAL CONFERENCE ON INFORMATION-CENTRIC NETWORKING, 2., 2015, San Francisco, NY, USA. **Proceedings...** San Francisco, NY, USA: ACM, 2015. p. 79–88. ISBN 978-1-4503-3855-4. Disponível em: <http://doi.acm.org/10.1145/2810156.2810168>.

FOTIOU, N.; MARIAS, G. F.; POLYZOS, G. C. Access control enforcement delegation for information-centric networking architectures. In: ACM SIGCOMM WORKSHOP ON INFORMATION-CENTRIC NETWORKING (ICN '12), 2., 2012, Helsinki, Finland. **Proceedings...** New York, NY, USA: ACM, 2012. p. 85–90. ISBN 978-1-4503-1479-4. Disponível em: <http://doi.acm.org/10.1145/2342488.2342507>.

GHALI, C. et al. Interest-based access control for content centric networks (extended version). **CoRR**, abs/1505.06258, 2015. Disponível em: <http://arxiv.org/abs/1505.06258>.

GOYAL, V. et al. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 13., 2006, Alexandria, Virginia, USA. **Proceedings...** New York, NY, USA: ACM, 2006. p. 89–98. ISBN 1-59593-518-5. Disponível em: <http://doi.acm.org/10.1145/1180405.1180418>.

HALPERN, J. Y.; MEYDEN, R. van der. A logic for sdsi's linked local name spaces. **CoRR**, cs.CR/0001026, 2000. Disponível em: <http://arxiv.org/abs/cs.CR/0001026>.

HAMDANE, B. et al. Data-based access control in named data networking. In: CONFERENCE ON COLLABORATIVE COMPUTING: NETWORKING, APPLICATIONS AND WORKSHARING (COLLABORATECOM 2013), 9., 2013, Austin, TX, USA. **Proceedings...** Austin, TX, USA: IEEE, 2013. p. 531–536. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6680021&tag=1>.

HEMMATI, E.; GARCIA-LUNA-ACEVES, J. A New Approach to Name-Based Link-State Routing for Information-Centric Networks. In: THE 2ND INTERNATIONAL CONFERENCE ON INFORMATION-CENTRIC NETWORKING, 2., 2015, San Francisco, California, USA. **Proceedings...** New York, NY, USA: ACM, 2015. p. 29–38. ISBN 978-1-4503-3855-4. Disponível em: <http://doi.acm.org/10.1145/2810156.2810173>.

JACOBSON, V. et al. Networking named content. In: INTERNATIONAL CONFERENCE ON EMERGING NETWORKING EXPERIMENTS AND TECHNOLOGIES, 5., 2009, Rome, Italy. **Proceedings...** New York, NY, USA: ACM, 2009. p. 1–12. ISBN 978-1-60558-636-6. Disponível em: <http://doi.acm.org/10.1145/1658939.1658941>.

JACOBSON, V. et al. Networking named content. **Commun. ACM**, ACM, New York, NY, USA, v. 55, n. 1, p. 117–124, jan. 2012. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2063176.2063204>.

KOPONEN, T. et al. A data-oriented (and beyond) network architecture. **SIGCOMM Comput. Commun. Rev.**, p. 181–192, 2007. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/1282427.1282402>.

KURIHARAY, J.; UZUN, E.; WOOD, C. An encryption-based access control framework for content-centric networking. In: IFIP NETWORKING CONFERENCE (IFIP NETWORKING), 2015, 14., 2015, Ottawa, Canada. **Proceedings...** Ottawa, Canada: IEEE, 2015. p. 1–9. Disponível em: <https://www.ietf.org/mail-archive/web/icnrg/current/pdf9_JIQ5GBeS.pdf>.

LYNN, B. **On the implementation of pairing-based cryptosystems**. Tese (Doutorado) — Stanford University, 450 Serra Mall, Stanford, CA 94305, Estados Unidos, 2007. Disponível em: <https://crypto.stanford.edu/pbc/thesis.html>.

MANNES, E. et al. Controle de acesso baseado em reencriptação por proxy em Redes Centradas em Informação. In: SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSEG 2014), 14., 2014, Belo Horizonte, MG, Brasil. **Anais...** Belo Horizonte, MG, Brasil: SBC, 2014. p. 2–15. Disponível em: <http://www.sbseg2014.dcc.ufmg.br/files/anais.pdf>.

MISRA, S.; TOURANI, R.; MAJD, N. E. Secure content delivery in information-centric networks: design, implementation, and analyses. In: ACM SIGCOMM WORKSHOP ON INFORMATION-CENTRIC NETWORKING (ICN '13), 3., 2013, Hong Kong, China. **Proceedings...** New York, NY, USA: ACM, 2013. p. 73–78. ISBN 978-1-4503-2179-2. Disponível em: <http://doi.acm.org/10.1145/2491224.2491228>.

Oracle. **CryptoUtils java API Reference for Oracle Storage Cloud Service:**. 2015. URL: https://docs.oracle.com/cloud/latest/storagecs_common/CSSAP/oracle/cloud/storage/internal/CryptoUtils.html. Acesso em: Março de 2015. Disponível em: <https://docs.oracle.com/cloud/latest/storagecs_common/CSSAP/oracle/cloud/storage/internal/CryptoUtils.html>.

Oracle. **Java SE | Oracle Technology Network | Oracle**. 2015. URL: http://www.oracle.com/technetwork/pt/java/javase/overview/index.html. Acesso em: Março de 2015. Disponível em: <http://www.oracle.com/technetwork/pt/java/javase/overview/index.html>.

Palo Alto Research Center. **The CCNx Project**. 2015. URL: http://blogs.parc.com/ccnx/. Acesso em: Outubro de 2015. Disponível em: <http://blogs.parc.com/ccnx/>.

PAPANIS, J. P. et al. On the use of attribute-based encryption for multimedia content protection over information-centric networks. **Transactions on Emerging Telecommunications Technologies**, v. 25, n. 4, p. 422–435, 2014. ISSN 2161-3915. Disponível em: <http://dx.doi.org/10.1002/ett.2722>.

PARC, Inc. **CCNxCon2015**. 2015. URL: http://www.ccnx.org/ccnxcon-2015/. Acesso em: Julho de 2015. Disponível em: <http://www.ccnx.org/ccnxcon-2015/>.

PENTIKOUSIS, K. et al. **Information-centric Networking: Evaluation Methodology draft-irtf-icnrg-evaluation-methodology-03**. 2015. URL: https://tools.ietf.org/html/draft-irtf-icnrg-evaluation-methodology-03.

Acesso em: Outubro de 2015. Disponível em: <https://tools.ietf.org/html/draft-irtf-icnrg-evaluation-methodology-03>.

PURSUIT. **Pursuing a Pub/Sub Internet**. 2009. URL: http://www.fp7-pursuit.eu. Acesso em: Setembro de 2013. Disponível em: <http://www.fp7-pursuit.eu>.

Python Software Foundation. **pycrypto 2.6.1 : Python Package Index**. 2015. URL: https://pypi.python.org/pypi/pycrypto. Acesso em: Março de 2015. Disponível em: <https://pypi.python.org/pypi/pycrypto>.

Python Software Foundation. **Welcome to Python.org**. 2015. URL: https://www.python.org/. Acesso em: Março de 2015. Disponível em: <https://www.python.org/>.

QIAO, Z. et al. Survey of attribute based encryption. In: SOFTWARE ENGINEERING, ARTIFICIAL INTELLIGENCE, NETWORKING AND PARALLEL/DISTRIBUTED COMPUTING (SNPD), 15., 2014, Las Vegas, NV. **Proceedings...** IEEE, 2014. p. 1–6. Disponível em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6888687>.

SAHAI, A.; WATERS, B. Fuzzy Identity-Based Encryption. In: ADVANCES IN CRYPTOLOGY - EUROCRYPT 2005, 24., 2005, Aarhus, Denmark. **Proceedings...** Amsterdam, Netherlands: Springer Berlin Heidelberg, 2005. p. 457–473. ISBN 978-3-540-25910-7. Disponível em: <http://dx.doi.org/10.1007/11426639_27>.

SINGH, S. et al. A trust based approach for secure access control in information centric network. **Journal of Information and Network Security**, v. 1, n. 2, p. 97–104, 2012.

SMETTERS, D.; JACOBSON, V. Securing network content. **Relatório Técnico TR-2009-1, Xerox Palo Alto Research Center-PARC**, 2009.

TANENBAUM, A. S.; BOS, H. **Modern Operating Systems**. 4th. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2014. ISBN 013359162X, 9780133591620.

University of Paderborn. **Home « NetInf**. 2013. URL: http://www.netinf.org/home/home/. Acesso em: Setembro de 2013. Disponível em: <http://www.netinf.org/home/home/>.

WANG, G.; LIU, Q.; WU, J. Hierarchical Attribute-based Encryption for Fine-grained Access Control in Cloud Storage Services. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 17., 2010, Chicago, Illinois, USA. **Proceedings...** New York, NY, USA: ACM, 2010. p. 735–737. ISBN 978-1-4503-0245-6. Disponível em: <http://doi.acm.org/10.1145/1866307.1866414>.

WANG, L. et al. **OSPFN: An OSPF based routing protocol for Named Data Networking**. [S.l.], 2012. URL: http://www.named-data.net/techreport/TR003-OSPFN.pdf. Acesso em: Março de 2015. Disponível em: <http://www.named-data.net/techreport/TR003-OSPFN.pdf>.

WOOD, C.; UZUN, E. Flexible end-to-end content security in CCN. In: CONSUMER COMMUNICATIONS AND NETWORKING CONFERENCE (CCNC 2014), 11., 2014, Las Vegas, NV,USA. **Proceedings...** New York, NW, USA: IEEE, 2014. p.

858 – 865. ISBN 978-1-4799-2356-4. Disponível em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6940528>.

XYLOMENOS, G. et al. A survey of information-centric networking research. **IEEE Communications Surveys Tutorials**, v. 16, n. 2, p. 1024–1049, Second 2014. ISSN 1553-877X.