Controle de acesso baseado em reencriptação por *proxy* em Redes Centradas em Informação

Elisa Mannes[1], Carlos Maziero[1], Luiz Carlos Lassance[2], Fábio Borges[3]

[1]Programa de Pós-graduação em Informática – Universidade Federal do Paraná (UFPR)

[2]Confederação das Cooperativas Centrais de Crédito Rural com Interação Solidária (CONFESOL)

[3]Technische Universität Darmstadt/CASED, Telecooperation Group

elisam@inf.ufpr.br, maziero@utfpr.edu.br

luiz@confesol.com.br, fabio.borges@cased.de

Abstract. *Information-centric networks (ICN) represent a promising approach to the Future Internet, addressing the shortcomings of the current Internet with a suitable infrastructure for content distribution. By naming, routing, and forwarding content instead of machine addresses, the ICN shift the protagonists at the network layer from hosts to contents. One implication is the in-network cache, which allows a better use of communication channels and faster delivery of content to the user. However, the ability to receive content from caches generates concerns about access control. In this context, we propose a solution for access control in ICN based on proxy re-encryption. The proposed solution ensures that only authorized users are able to access content, while maintaining the beneficial effects of caching in ICN, even in face of malicious entities.*

Resumo. *As redes centradas em informação (ICN) representam uma abordagem promissora para a Internet do Futuro, pois aborda as atuais deficiências da Internet com uma infraestrutura mais adequada para a distribuição de conteúdo. Ao nomear, rotear e encaminhar conteúdo ao invés de endereços de máquina, a ICN desloca o protagonismo da camada de rede das máquinas para os conteúdos. Uma das implicações dessa mudança é o cache nos dispositivos de rede, que permite uma melhor utilização dos canais de comunicação e uma entrega mais rápida do conteúdo ao usuário. Entretanto, a possibilidade de receber conteúdo dos caches gera preocupações com relação ao controle de acesso. Neste contexto, propõe-se uma solução para controle de acesso em ICN baseada em reencriptação por proxy. A solução proposta garante que somente usuários autorizados acessem o conteúdo na rede enquanto se mantém os benefícios do sistema de cache em ICN, mesmo diante de uma entidade maliciosa.*

1. Introdução

As redes centradas em informação (ICN - *Information-centric Networks*) [Ahlgren et al. 2012, Brito et al. 2012] propõem superar as dificuldades atuais da Internet modificando a principal entidade da rede de máquinas para conteúdos. Esse novo paradigma traz características especiais para a Internet, pois nomear, rotear e encaminhar conteúdo na rede ao invés de endereços de máquina permite a implementação de *cache*

nos dispositivos da rede, por exemplo. Um mecanismo de *cache* diretamente na rede potencializa um melhor desempenho na entrega do conteúdo e torna a arquitetura mais adequada para os atuais padrões de tráfego, inclusive para dispositivos móveis. Entretanto, o paradigma de ICN também modifica os aspectos relacionados à segurança de redes. Por exemplo, a nomeação dos conteúdos exige que mecanismos de segurança sejam focados no conteúdo ao invés de prover segurança para máquinas, *links* e sessões. Além do mais, o emprego de *caches* na rede resulta em conteúdos recuperados de qualquer dispositivo por qualquer usuário, trazendo novos desafios com relação à **privacidade** e ao **controle de acesso**, já que não é obrigatório que o usuário se conecte ao provedor de conteúdo para requisitar o conteúdo. Esse problema toma proporções ainda maiores quando considerados os ambientes móveis, em que qualquer dispositivo pode rotear e armazenar conteúdos em *cache*, incluindo dispositivos maliciosos ou comprometidos.

As soluções atuais para controle de acesso na distribuição de conteúdo, apesar de serem transferíveis para ICN, geralmente inviabilizam a proposta do uso de *cache* na rede. O uso de criptografia assimétrica, por exemplo, inibe o compartilhamento das cópias em *cache* por diversos usuários, pois o conteúdo é encriptado para cada usuário individualmente. Além do mais, a arquitetura atual exige que os usuários se autentiquem em servidores específicos para garantir a segurança ao requisitar conteúdos. Novamente, essa solução prejudica a implementação de mecanismos de *cache*. Além dessas soluções tradicionais, existem soluções de controle de acesso desenvolvidas especialmente para uso em arquiteturas de ICN [Ion et al. 2013, Misra et al. 2013, Papanis et al. 2013, Fotiou et al. 2012, Singh et al. 2012, Hamdane et al. 2013]. Contudo, a maioria emprega o uso de criptografia simétrica e foca na garantia de que somente usuários autorizados tenham acesso à chave utilizada. Essa estratégia pode representar um problema caso a chave seja divulgada por uma entidade maliciosa, já que é a mesma para todos os usuários.

Neste artigo, propomos uma solução de controle de acesso para conteúdo protegido em ICN focando em três aspectos principais: (i) o conteúdo pode ser armazenado em qualquer dispositivo e recuperado por qualquer usuário; (ii) os usuários que acessam o conteúdo não podem decifrá-lo, a menos que sejam autorizados pelo provedor de conteúdo; (iii) não há a adição de novas entidades na rede para a aplicação ou a validação de políticas de acesso. A reencriptação por *proxy* [Ateniese et al. 2006] é um esquema de criptografia em que uma mensagem encriptada com uma chave pública A pode ser transformada em uma mensagem encriptada com uma chave pública B, sem expor o conteúdo original nem as chaves privadas correspondentes. Essa transformação é tradicionalmente feita por uma entidade semi-confiável denominada *proxy de reencriptação*, usando uma *chave de reencriptação* definida a partir das chaves A e B. A reencriptação por *proxy* pode potencialmente ser usada como mecanismo de controle de acesso a conteúdos em ICN da seguinte forma: o conteúdo original é encriptado com a chave pública do provedor, gerando um conteúdo encriptado único. Um usuário interessado no conteúdo pode recuperá-lo no *cache* mais próximo; em seguida ele deve interagir com o provedor, visando obter a chave de reencriptação necessária para reencriptar aquele conteúdo com sua chave pública, permitindo seu acesso. Ainda que o controle de acesso seja desejável para a maioria dos conteúdos na Internet, neste trabalho focamos as especificidades de conteúdos populares, em que um grande conjunto de usuários esteja interessado no mesmo conteúdo, tais como vídeos, *e-books*, *streaming* e atualizações de *software*. Nesses cenários, o mecanismo de *cache* é utilizado em seu potencial máximo e os benefícios da ICN surgem de uma forma

mais substancial. Entretanto, a solução proposta é aplicável para outros tipos de conteúdo.

Este artigo está organizado como segue: a Seção 2 descreve os esforços atuais para o controle de acesso em ICN e discute suas principais deficiências. A Seção 3 explica os fundamentos de reencriptação por *proxy*, que fundamenta a solução proposta. A Seção 4 detalha a proposta e descreve seus aspectos técnicos. A Seção 5 valida a solução proposta e analisa o desempenho com relação ao tempo computacional para diferentes tamanhos de mensagens e chaves. A Seção 6 discute o uso da solução proposta com relação ao desempenho, segurança e adequabilidade para a arquitetura de ICN. Por fim, a Seção 7 conclui o artigo e sugere trabalhos futuros.

2. Controle de acesso em ICN

A possibilidade de armazenamento de conteúdo nos dispositivos da rede gera uma grande preocupação com relação ao controle de acesso dos conteúdos, pois as cópias em *cache* podem ser acessadas por qualquer usuário, inclusive aqueles que não têm autorização de acesso ao conteúdo. Para serviços que requerem o pagamento de mensalidades, tais como *Netflix*, *Hulu*, *Amazon*, *Apple* e *Play Store* e *Steam*, fica ainda mais evidente a necessidade de assegurar o controle de acesso para conteúdo armazenado em *cache*. Tais serviços geralmente requerem um rigoroso controle das contas de usuários, do número de reproduções do conteúdo e da quantidade de dispositivos autorizados, por exemplo. Restringir o conhecimento do nome do conteúdo somente para os usuários autorizados não é suficiente, já que em ICN as ações de roteamento e de encaminhamento são realizadas diretamente pelo nome do conteúdo e, desta forma, os nomes dos conteúdos podem ser facilmente descobertos. Assim, esse tipo de aplicação requer uma solução de controle de acesso mais robusta e adequada para o uso em ICN. De outra forma, é pouco provável que a arquitetura de ICN seja adotada para a distribuição de conteúdos protegidos.

A encriptação do conteúdo é apontada como a ação mais básica para garantia que somente usuários autorizados, que possuam uma chave válida, possam acessá-lo [Jacobson et al. 2012]. Contudo, a encriptação de um mesmo conteúdo para usuários diferentes não é conveniente, pois o conteúdo encriptado para um usuário específico não pode ser acessado por outro e, portanto, as cópias armazenadas em *cache* não são aproveitadas. Como alternativa, outras abordagens foram propostas. Uma delas propõe a encriptação do conteúdo com uma chave simétrica, aproveitando o mecanismo de *cache*, enquanto as chaves simétricas e as licenças são encriptadas para grupos de usuários. Exemplos de tais abordagens são as soluções propostas por [Misra et al. 2013] e [Papanis et al. 2013], que exploram a criptografia de *broadcast* e a baseada em atributos, respectivamente. Contudo, essas soluções protegem parcialmente o conteúdo, pois o uso da criptografia simétrica pode representar um ponto de vulnerabilidade no caso de a chave simétrica ser divulgada na rede. Uma exceção é a solução proposta por [Ion et al. 2013], em que o conteúdo é encriptado de acordo com atributos e as políticas de acesso são aplicadas pelo próprio texto encriptado ou pela chave do usuário. Entretanto, essa abordagem encripta o conteúdo por grupos de atributos e, desta forma, o conteúdo em *cache* pode não servir a todos os usuários. Outras soluções propostas [Fotiou et al. 2012, Singh et al. 2012, Hamdane et al. 2013] requerem o uso de servidores terceirizados para aplicar as políticas de acesso, que além de necessitar de uma infraestrutura terceira, pode depender de peculiaridades das arquiteturas de ICN.

A reencriptação por *proxy* já foi explorada no contexto de controle de

acesso a conteúdos por [Xiong et al. 2012, Kissel and Wang 2013, Wood and Uzun 2014]. [Xiong et al. 2012] apresenta uma solução em que o provedor de conteúdo cria grupos com um respectivo par de chaves pública-privada. A reencriptação ocorre através de um *proxy* localizado na nuvem. O conteúdo é dividido em duas partes e somente a menor parte é reencriptada. [Kissel and Wang 2013] também propõe que o proprietário do conteúdo crie um grupo de usuários com um respectivo par de chave pública-privada. Ao entrar no grupo, é dado ao usuário uma chave de re-encriptação que permite que ele re-encripte e posteriormente decripte os conteúdos do grupo. Para revogar o acesso de um usuário, é proposto que o grupo seja desfeito e recriado com um novo par de chaves e que seja dado aos usuários autorizados a nova chave de re-encriptação, sendo o primeiro trabalho a propor a transferência da função de *proxy* para o usuário. Esses dois trabalhos consideram que os provedores de conteúdo têm controle sobre o conteúdo, podendo revogar o acesso a qualquer tempo, o que não é garantido em ICN. Paralelamente ao nosso trabalho, [Wood and Uzun 2014] explora a re-encriptação por *proxy* no contexto de ICN. Contudo, a análise conduzida pelos autores os levam a propor o uso da criptografia simétrica para encriptar os conteúdos e a aplicação da re-encriptação para a proteção de chaves simétricas, o que incorre nas mesmas deficiências apontadas anteriormente. Desta forma, observa-se que fornecer uma solução de controle de acesso para ICN não é uma tarefa trivial e requer o alinhamento de vários objetivos, principalmente com relação à segurança dos dados e ao desempenho na entrega de conteúdo através dos *caches*.

3. Reencriptação por *proxy*

A ideia geral dos esquemas tradicionais de reencriptação por *proxy* (PRE - *proxy re-encryption* [Ateniese et al. 2006]) é permitir a transformação de uma mensagem encriptada com a chave pública de um usuário A, para uma mensagem encriptada com a chave pública de um usuário B. Essa transformação acontece em uma terceira entidade considerada semi-confiável, o *proxy*. O usuário A autoriza o *proxy* a transformar as mensagens encriptadas com a sua chave pública para a chave pública do usuário B ao concedê-lo uma **chave de reencriptação** $rk_{A \to B}$. A Figura 1 ilustra o funcionamento de um esquema básico de reencriptação por *proxy* composto pelos usuários A e B e por um *proxy*. Neste exemplo, o usuário A encripta um conteúdo C com a sua chave pública $kp(A)$, gerando $\{C\}_{kp(A)}$. Caso o usuário A queira permitir que o usuário B acesse o conteúdo C, ele envia ao *proxy* o conteúdo $\{C\}_{kp(A)}$ e uma chave de reencriptação $rk_{A \to B}$, calculada com base na chave pública $kp(B)$ do usuário B. O *proxy* então utiliza a chave de reencriptação enviada por A para reencriptar o conteúdo para B, gerando $\{C\}_{kp(B)}$. O *proxy* envia o conteúdo $\{C\}_{kp(B)}$ para o usuário B, que o decripta utilizando a sua chave privada $kv(B)$.

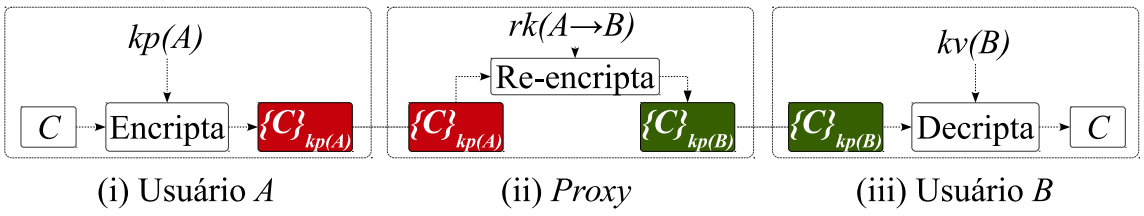

Figura 1. Visão geral do esquema de reencriptação por *proxy*

Em geral, os esquemas de reencriptação por *proxy* garantem duas asserções básicas: o *proxy* não pode ser capaz de acessar o conteúdo da mensagem que reencripta e, de

posse da mensagem encriptada e da chave de reencriptação, não pode recuperar as chaves privadas de A ou B. [Ateniese et al. 2006, Chow et al. 2010] definem os algoritmos que compõem um esquema de reencriptação por *proxy*, definidos abaixo:

CONFIGURAÇÃO: recebe como entrada um parâmetro de segurança k e tem como saída uma tupla de parâmetros globais PARAM.

GERAÇÃO DE CHAVES: gera pares de chaves pública-privada (kp, kv).

ENCRIPTAÇÃO: ao receber $kp(A)$ e uma mensagem m, gera uma mensagem encriptada $\{m\}_{kp(A)}$.

DECRIPTAÇÃO: ao receber $kv(A)$ e $\{m\}_{kp(A)}$, gera como saída a mensagem m.

GERAÇÃO DE CHAVE DE REENCRIPTAÇÃO: tem como entrada a chave privada $kv(A)$ e a chave pública $kp(B)$ e como saída uma chave de reencriptação $rk_{A\to B}$.

REENCRIPTAÇÃO: ao entrar a chave de reencriptação $rk_{A\to B}$ e o texto encriptado $\{m\}_{kp(A)}$, tem como saída $\{m\}_{kp(B)}$.

Para fundamentar a solução proposta neste artigo, são necessárias três propriedades fundamentais dos esquemas de reencriptação por *proxy* (contudo outras propriedades podem ser incorporadas para agregar outras funcionalidades):

- *Unidirecionalidade*: a delegação de direitos de decriptar de $A \to B$ não implica na delegação de $B \to A$;
- *Salto único*: somente mensagens originais podem ser reencriptadas;
- *Segurança contra conluio*: o usuário B e o *proxy* em conluio não conseguem recuperar a chave privada de A.

Com base nos esquemas de reencriptação por *proxy* tradicionais, propomos uma solução para controle de acesso em ICN. Para alinhar o esquema de reencriptação por *proxy* com as peculiaridades da ICN, a entidade *proxy* é eliminada do processo de reencriptação. Contudo, as funções tradicionalmente desempenhadas pelo *proxy* são transferidas para o usuário. Sendo assim, na solução proposta, há somente duas entidades envolvidas na reencriptação: a fonte e o usuário. Neste artigo, emprega-se o esquema de reencriptação por *proxy* proposto por [Chow et al. 2010].

4. Controle de acesso usando reencriptação

Diferentemente das abordagens existentes, nosso objetivo é propor uma solução de controle de acesso que seja alinhada ao funcionamento das arquiteturas de ICN, garantindo a disponibilidade do conteúdo em qualquer *cache* na rede enquanto permite o controle de acesso ao conteúdo pelo provedor do mesmo. Além do mais, é desejável que a solução não utilize entidades extras para o controle de acesso nem modifique as funções do núcleo de qualquer arquitetura de ICN, mantendo o processo simples e seguindo as especificações da arquitetura de ICN. Na nossa visão, o paradigma de ICN deve ser mantido simples para entregar o conteúdo da melhor forma, sem sobrecarregar os roteadores com a verificação de políticas de acesso ou criar uma nova infraestrutura de servidores para validar o acesso de usuários e serviços ao conteúdo em *cache*. Para isso, propomos a utilização de um esquema de reencriptação por *proxy* para o controle de acesso aos conteúdos em ICN. Neste esquema, os provedores de conteúdo encriptam os conteúdos com chaves públicas

correspondentes e distribuem o conteúdo na rede conforme as requisições dos usuários. Os usuários podem recuperar os conteúdos tanto do provedor de conteúdo como dos *caches*. Para decifrá-los, os usuários devem solicitar uma chave de reencriptação para o provedor de conteúdo. As próximas subseções detalham a solução proposta.

4.1. Modelo de rede

Neste trabalho, consideramos as especificidades da arquitetura NDN (*Named-Data Network* [Jacobson et al. 2012]), mas por não modificar entidades na rede, a solução proposta pode ser aplicada a qualquer arquitetura de ICN. A infraestrutura da NDN é composta por provedores de conteúdo (P), roteadores (R) e usuários (U). Os provedores de conteúdo anunciam os nomes dos seus conteúdos na rede. Cada conteúdo é dividido em *chunks* de 4Kb, que são individualmente nomeados e a ligação entre o conteúdo e o seu nome é assinada criptograficamente, para que os usuários possam validar a integridade e a autenticidade do conteúdo, conforme [Smetters and Jacobson 2009]. Para requisitar um conteúdo na rede, o usuário deve enviar um pacote de *Interesse* e, em resposta, recebe um pacote de *Dados* com o *chunk* solicitado. Os nomes de conteúdos vindos do mesmo provedor de conteúdo compartilham prefixos em comum que permitem a agregação nas tabelas de roteamento nos roteadores.

Além de rotear e encaminhar os conteúdos nomeados para os usuários, os roteadores têm a função de armazenar conteúdo em seus *caches* de acordo com as políticas de *cache*, desta forma permitindo um melhor desempenho na entrega dos conteúdos. Ao receber um pacote de *Interesse*, o roteador verifica seu *cache* (*CS - content store*). Caso o conteúdo solicitado esteja armazenado no *cache*, ele é rapidamente entregue para a *face*[1] cujo pedido foi recebido. Caso o conteúdo não esteja presente no *cache*, o roteador verifica sua tabela de interesses pendentes (*PIT - pending interest table*). Se alguma *face* solicitou o mesmo conteúdo e ainda não foi atendida, o pedido é agregado a essa entrada, adicionando a *face* pela qual o pedido foi recebido. Desta forma, os roteadores evitam que requisições para o mesmo conteúdo sejam enviadas repetidamente para a rede. Se não houver um pedido pendente para o conteúdo solicitado, uma nova entrada é criada e o roteador então consulta a sua base de informação de encaminhamento (*FIB - forwarding information base*) para rotear o pedido em direção ao provedor do conteúdo solicitado. Cada roteador no caminho em direção ao provedor do conteúdo realiza todas essas etapas. O pacote de *Dados* contendo o conteúdo retorna pelo mesmo caminho em que foi solicitado, consumindo as entradas da PIT nos roteadores envolvidos.

A Figura 2 ilustra o funcionamento básico da arquitetura NDN, contendo um provedor de conteúdo, $P1$, quatro roteadores, $R1, R2, R3$ e $R4$ e dois usuários, $U1$ e $U2$. $P1$ possui dois conteúdos, A e B, que deseja disponibilizar para seus usuários. Na Figura 2(a), o usuário $U1$ envia um pacote de *Interesse* para solicitar o conteúdo A para o roteador $R4$, que roteia o pedido em direção ao provedor de conteúdo seguindo a rota $R4 \rightarrow R2 \rightarrow R1$. O pacote de *Dados* contendo o conteúdo A segue o caminho contrário em direção a $U1$. Cada roteador armazena a cópia em seu *cache*, caso a cópia ainda não esteja presente. Na Figura 2(b), o usuário $U2$ solicita o mesmo conteúdo A para $R4$ e como $R4$ tem o conteúdo em seu *cache*, a requisição é prontamente atendida.

[1] A arquitetura NDN nomeia as interfaces de *faces* para representar tanto interfaces de redes quanto interfaces de aplicações.

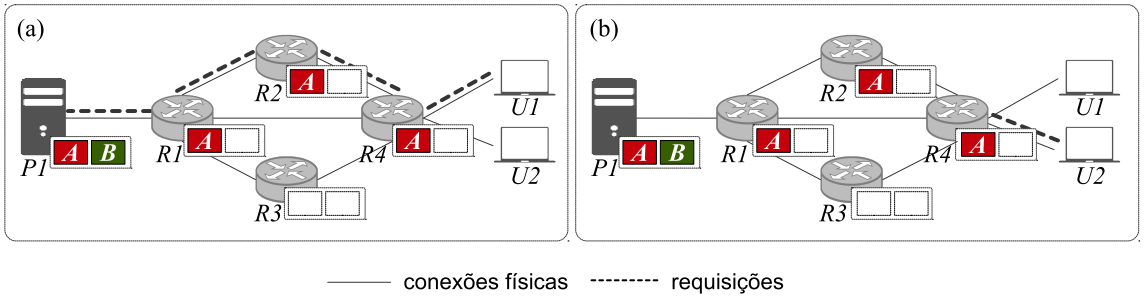

Figura 2. Infraestrutura da arquitetura NDN

4.2. Modelo de ameaças

Assume-se que os provedores de conteúdo exigem que os usuários sejam devidamente registrados na aplicação para ter acesso ao conteúdo protegido. Além disso, o provedor de conteúdo deve validar o usuário (verificar sua identidade e chave pública com uma infraestrutura de chave pública, por exemplo) para certificar-se que o usuário é de fato legítimo para o serviço e garantir seu acesso de acordo com as políticas de uso (tipo de usuário, inscrição, idade). Também se assume que o provedor de conteúdo se comporta corretamente, ou seja, não distribui conteúdo protegido ou direitos de acesso a usuários não autorizados. Os roteadores seguem o comportamento do modelo *honesto porém curioso*, em que eles desempenham corretamente suas funções (roteamento, encaminhamento e armazenamento de conteúdo em *cache*), porém podem ser curiosos e tentar acessar o conteúdo que estão roteando.

Considera-se que entidades maliciosas ($\mathcal{A}$) são usuários ilegítimos que não têm acesso ao conteúdo do provedor, ou ainda usuários legítimos que tentam acessar conteúdo ao qual não têm autorização. A intenção dessas entidades maliciosas é obter acesso ao conteúdo protegido sem ter as obrigações de usuários autorizados, tais como pagamento, verificação de dados pessoais ou ainda tipos de acesso diferenciados, como contas básicas e avançadas. Eles podem explorar o conteúdo protegido na rede pelas seguintes formas:

- aprender/descobrir o nome do conteúdo e requisitá-lo na rede;
- espionar os canais de comunicação de usuários ou interferir em pontos de acesso;
- examinar ou sondar *caches* próximos ou acessar diretamente o seu próprio *cache*.

Além disso, da mesma forma que podem solicitar conteúdo protegido na rede, as entidades maliciosas também podem recuperar as chaves de reencriptação (Seção 3) a partir dos *caches*. Também se assume que a ligação entre o nome do conteúdo e o conteúdo é devidamente encriptada e que os usuários são capazes de verificar a integridade e a autenticidade do conteúdo [Smetters and Jacobson 2009]. Por fim, assume-se também que os usuários têm acesso ao conteúdo oferecido pelo provedor de conteúdo através de uma aplicação específica e, desta forma, não há necessidade de descobrir o nome do conteúdo de antemão ou por meios não confiáveis.

4.3. Solução proposta

A solução proposta está dividida em três domínios: *domínio do provedor de conteúdo*, *domínio da rede* e *domínio do usuário*. O domínio do provedor de conteúdo engloba a encriptação do conteúdo e a geração de chaves de reencriptação para os usuários. O

domínio da rede refere-se ao roteamento e ao encaminhamento de conteúdo na rede seguindo o paradigma de ICN (Seção 4.1). O domínio do usuário é composto pela aplicação do provedor de conteúdo e pelas operações de reencriptação e decriptação do conteúdo. Tanto o provedor de conteúdo quanto uma infraestrutura de chave pública especializada podem ser responsáveis por distribuir pares de chaves pública-privada aos provedores de conteúdo e aos usuários, seguindo as especificações do algoritmo CONFIGURAÇÃO. Cada par de chave pública-privada é composto por duas chaves públicas e duas chaves privadas; essa característica é introduzida por [Chow et al. 2010] para garantir que a chave privada da fonte não seja descoberta em caso de conluio do *proxy* com o usuário no esquema original. Essa característica é extremamente importante na solução proposta, já que se considera a transferência da função do *proxy* com o usuário. A partir da aplicação disponibilizada pelo provedor de conteúdo, os usuários são autenticados e podem navegar e requisitar conteúdos, podendo ser atendidos tanto pelo provedor de conteúdo quanto por *caches* mais próximos. A seguir, são detalhadas as ações realizadas pelo provedor de conteúdo e pelos usuários para acessar um conteúdo protegido na ICN.

Encriptação e distribuição de conteúdo: o provedor de conteúdo possui um conjunto $\mathcal{C} = \{A, B, \ldots, Z\}$ de conteúdos que deseja disponibilizar. Cada conteúdo em $\mathcal{C}$ é segmentado em *chunks* e cada *chunk* é individualmente encriptado com a chave pública kp do conteúdo (todos os *chunks* pertencentes ao mesmo conteúdo são encriptados com a mesma chave). A chave privada correspondente (kv) é guardada em segredo pelo provedor de conteúdo, como de costume. A ligação entre o nome do conteúdo e o conteúdo é realizada pelo provedor com seu par de chaves pública-privada, $kp(P)$ e $kv(P)$, como sugerido em [Smetters and Jacobson 2009]. O conteúdo é distribuído conforme as requisições dos usuários, sendo armazenado em *cache* na rede de acordo com as políticas de *cache* adotadas. Neste estágio, o conteúdo pode estar em qualquer lugar na rede, porém, como a chave privada correspondente ao conteúdo é conhecida somente pelo provedor, nenhum usuário é capaz de decifrá-lo. A Figura 3(a) detalha o funcionamento dessas operações, em que o provedor possui um conjunto de conteúdos, A, B, C e D e os encripta com os seus respectivos pares de chaves, formando o conjunto $\{A\}_{kp(A)}$, $\{B\}_{kp(B)}$, $\{C\}_{kp(C)}$ e $\{D\}_{kp(D)}$, que é disponibilizado para os usuários.

Geração e distribuição da chave de reencriptação: para um usuário decriptar um conteúdo, A por exemplo, este deve ser reencriptado. Para isso, o usuário legítimo U que deseja decriptar A deve solicitar uma chave de reencriptação $rk_{A \to U}$ para o provedor. Para tal, U envia um pacote de *Interesse* para o provedor de conteúdo, requisitando uma chave de reencriptação para o conteúdo A. O provedor de conteúdo verifica se o usuário é autorizado a acessar o conteúdo A e então calcula a chave de reencriptação $rk_{A \to U}$, com base na chave pública do usuário, $kp(U)$, e na chave privada utilizada para encriptar o conteúdo, $kv(A)$, e envia para o usuário um pacote de *Dados* contendo a chave de reencriptação. A Figura 3(b) detalha essa operação. Somente o usuário U é capaz de decriptar o conteúdo A utilizando $rk_{A \to U}$, já que é necessário o uso da chave privada do usuário U (ao menos que o usuário U também divulgue sua chave privada). É inútil para uma entidade maliciosa em potencial requisitar o conteúdo e interceptar uma chave de reencriptação: ela pode ser capaz de reencriptar o conteúdo, mas a mensagem encriptada resultante só poderá ser decriptada pelo usuário que possuir a chave privada correspondente à chave de reencriptação.

Reencriptação e decriptação do conteúdo: após receber o conteúdo A e a chave de

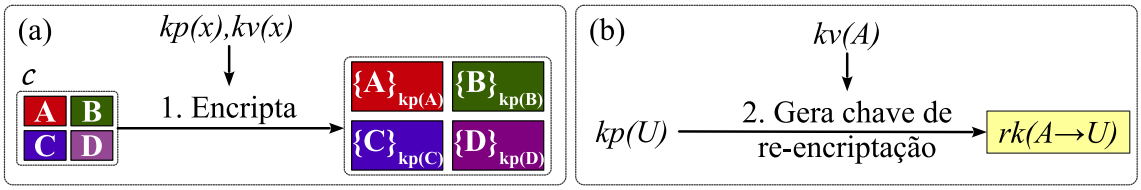

Figura 3. Domínio da fonte: (a) encriptação e (b) geração de chave de reencriptação

reencriptação $rk_{A \to U}$, o usuário U pode decriptar o conteúdo. A Figura 4(a) detalha a operação de reencriptação realizada pelo usuário U. Primeiramente, é necessário re-encriptar o conteúdo A com a chave de reencriptação correspondente. A saída deste procedimento é uma mensagem $\{A\}_{kp(U)}$, encriptada com a chave pública do usuário U, $kp(U)$. Então, o conteúdo $\{A\}_{kp(U)}$ pode ser decriptado com a chave privada do usuário U, $kv(U)$, recuperando o conteúdo A que pode ser consumido pela aplicação, conforme ilustra a Figura 4(b). As chaves de reencriptação são exclusivas de cada usuário e de cada conteúdo, portanto, cada usuário deve requisitar sua chave para o provedor de conteúdo. Desta forma, o provedor de conteúdo pode negar o envio de chaves de reencriptação para usuários que não cumpram os requisitos impostos.

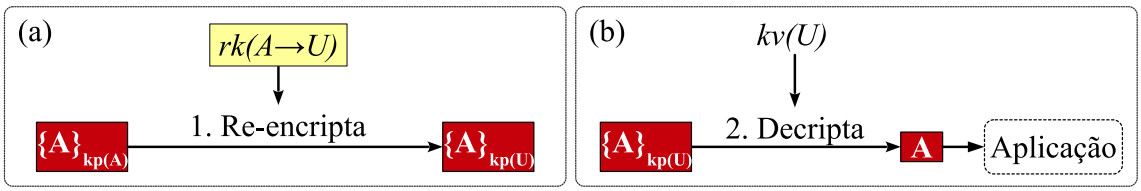

Figura 4. Domínio do usuário: (a) reencriptação e (b) decriptação

Invalidação da chave de reencriptação: uma vez que o usuário U possua a chave de reencriptação $rk_{A \to U}$ para o conteúdo A, ele é capaz de decriptar o conteúdo A sempre que desejar. Além disso, qualquer conteúdo que tenha sido encriptado pelo provedor de conteúdo com a mesma chave pública utilizada para encriptar o conteúdo A, $kp(A), kv(A)$ pode ser decriptado por U com a chave $rk_{A \to U}$. Essa é a principal razão pela qual é obrigatório que cada conteúdo tenha um par distinto de chaves pública-privada. Essa peculiaridade dificulta a invalidação das chaves de reencriptação. Ainda assim, é necessário que o provedor de conteúdo possa negar acesso aos usuários que não tenham mais permissão para acessar aos conteúdos, mesmo que tais usuários já possuam a chave de reencriptação. Uma forma de invalidar as chaves de reencriptação é renovando periodicamente a encriptação dos conteúdos com chaves públicas diferentes. Desta forma, os conteúdos teriam chaves de reencriptação correspondentes diferentes, forçando os usuários a solicitar as novas chaves de reencriptação sempre que desejarem acessar um conteúdo e suas chaves de reencriptação forem inválidas.

5. Avaliação

O objetivo da avaliação é validar o desempenho computacional do esquema de reencriptação por *proxy* proposto por [Chow et al. 2010]. Este esquema não possui uma implementação disponível, portanto, é fundamental avaliar a sua viabilidade computacional. Para isso, implementamos em Python, versão 2.7, os seis algoritmos do esquema:

CONFIGURAÇÃO, GERAÇÃO DE CHAVES, ENCRIPTAÇÃO, DECRIPTAÇÃO, GERAÇÃO DE CHAVE DE REENCRIPTAÇÃO e REENCRIPTAÇÃO[2]. A Tabela 1 descreve os parâmetros utilizados na validação[3].

Tabela 1. Parâmetros utilizados na avaliação da solução

Parâmetro	Valor	Parâmetro	Valor
Tamanho da chave (k)	1024, 2048 bits	Funções de *hash* H_1, H_3, H_4	mod q
Tamanho da mensagem (ℓ_0)	0.5, 1, 2, 4, 8, 16, 32 Kb	Função de *hash* H_2	mod $2^{(\ell_0+\ell_1)}$
Parâmetro de segurança (ℓ_1)	160 *bits*		

A validação do esquema foi realizada em um *notebook* com processador Core 2 Duo 1.66GHz, 32bits, 2Gb RAM e sistema operacional Ubuntu 13.10. A métricas adotadas foram o tempo para encriptar e gerar as chaves de reencriptação, que são ações desempenhadas pela fonte, e o tempo para reencriptar e decriptar o conteúdo, ações desempenhadas pelo usuário. A validação é realizada com diferentes tamanhos de mensagens, que representam os *chunks* enviados pela ICN. Um conteúdo é formado por um conjunto de *chunks*. Ressaltamos que, apesar do tamanho padrão de um *chunk* na arquitetura NDN ser 4Kb [Salsano et al. 2012], variamos os tamanhos dos *chunks* para ter uma noção mais clara do comportamento do esquema de reencriptação por *proxy*. Além disso, as mensagens foram encriptadas com diferentes tamanhos de chaves. Comparou-se o desempenho computacional do esquema de reencriptação por *proxy* com um outro esquema de criptografia assimétrica, o RSA. Os resultados apresentados são a média de 35 execuções do algoritmo, com um intervalo de confiança de 95%. Os algoritmos de CONFIGURAÇÃO e de GERAÇÃO DE CHAVES podem opcionalmente ser executados em uma infraestrutura de chaves públicas e, portanto, não consideramos os custos computacionais desses algoritmos. Contudo, assume-se que os provedores de conteúdos e os usuários conheçam de antemão suas respectivas chaves públicas-privadas e os parâmetros do sistema.

A Figura 5 apresenta os resultados obtidos com as operações desempenhadas pelo provedor de conteúdo: ENCRIPTAÇÃO e GERAÇÃO DE CHAVES DE REENCRIPTAÇÃO. A operação de encriptação, apresentada na Figura 5 (a), tem desempenho satisfatório e similar ao RSA, encriptando mensagens de 1 a 32Kb em menos de 200ms. Contudo, enquanto o esquema de [Chow et al. 2010] apresenta um comportamento linear com relação ao tamanho da mensagem a encriptar, o RSA apresenta um comportamento de crescimento ao aumentar o tamanho da mensagem, o que fica evidenciado no caso de mensagens de 32Kb, que o RSA tem desempenho inferior ao esquema de reencriptação por *proxy* adotado. Contudo, vale ressaltar que para pacotes de 4Kb, que é o padrão da arquitetura NDN, tanto o esquema de reencriptação por *proxy* quanto o RSA tem desempenho similar, com tempo de processamento abaixo de 200ms. Porém, deve-se considerar que o esquema de reencriptação por *proxy* possui a operação extra de geração de chaves de reencriptação na fonte, operação não presente no RSA. O tempo para gerar a chave de reencriptação também é menor que 200ms para mensagens de 4Kb. Porém, o tempo de processamento aumenta com o tamanho das mensagens, conforme ilustra a Figura 5 (b). Como o tamanho padrão de *chunks* é 4Kb, isso não representa uma grande questão. Contudo, pode ser

[2]A implementação desses algoritmos está disponível em *http://www.inf.ufpr.br/elisam/proxy*.

[3]Neste estágio, escolhemos funções de *hash* simples apenas para a validação da solução, portanto, não aferimos a segurança das mesmas.

um problema caso adote-se tamanhos de *chunks* grandes, já que no esquema proposto a fonte deve gerar chaves de reencriptação para os conteúdos a medida em que os usuários solicitam acesso. Também vale ressaltar que os tempos de processamento podem variar de acordo com o *hardware* utilizado.

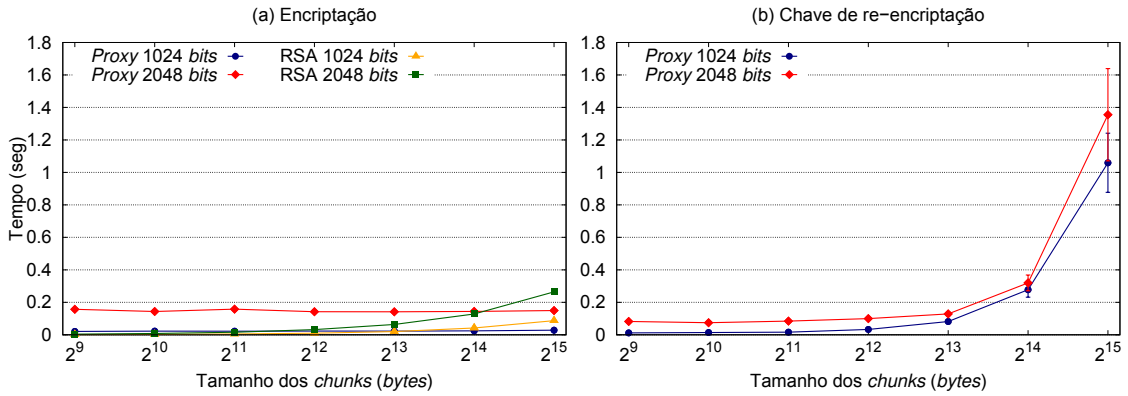

Figura 5. Avaliação da encriptação e da geração de chave de reencriptação

A Figura 6 apresenta os resultados obtidos com as operações desempenhadas pelo usuário: REENCRIPTAÇÃO e DECRIPTAÇÃO. A reencriptação constitui uma tarefa extra em relação aos esquemas tradicionais de criptografia assimétrica. De todas as operações do esquema de reencriptação por *proxy*, a reencriptação é a que apresentou a maior carga computacional, conforme ilustra a Figura 6(a). Para *chunks* de 4Kb, o tempo para processamento da reencriptação com uma chave de 2048 *bits* é de aproximadamente 800ms e aumenta conforme o tamanho das mensagens. Em contrapartida, a operação de decriptação é a operação menos custosa em termos de processamento, como apresenta a Figura 6(b). Para *chunks* de 4Kb a decriptação ocorre em aproximadamente 10ms para chaves de 1024 *bits* e 70ms para chaves de 2048 *bits*. Ainda assim o esquema de reencriptação por *proxy* se apresenta mais escalável que o RSA, que tem um alto custo computacional para decriptação, mesmo considerando *chunks* pequenos como o de 4Kb, em que o tempo para decriptação é próximo de 1 segundo (de fato, na prática, o RSA é utilizado para cifrar *hashes*). Isso ocorre mesmo que se considere a soma dos tempos das operações de reencriptação e decriptação para o esquema de reencriptação por *proxy*, conforme ilustra a Figura 6(c). A junção dessas operações é justificável, pois são operações dependentes e sempre são executadas em conjunto.

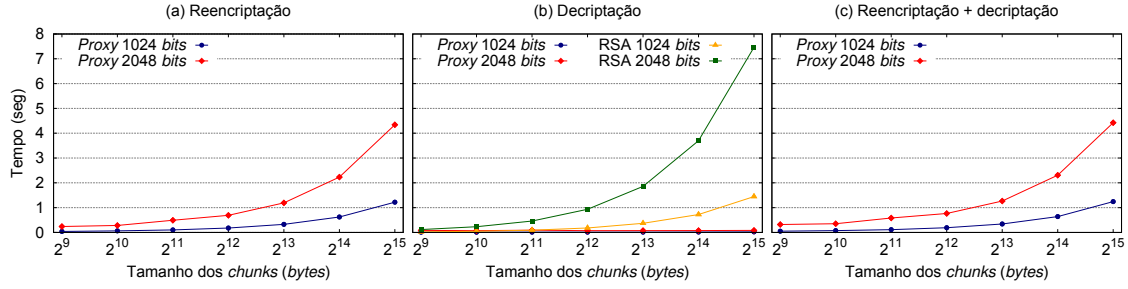

Figura 6. Avaliação da reencriptação e decriptação

6. Discussão

O principal objetivo da solução proposta é fornecer uma forma adequada para os provedores controlarem o acesso aos seus conteúdos em ICN. Os principais obstáculos para alcançar tal propriedade vêm de duas características intrínsecas do paradigma de ICN: (i) a implementação de conteúdo nomeado e (ii) o *cache* ubíquo de tais conteúdos. A seguir, apresenta-se uma discussão acerca da solução proposta com relação ao desempenho, segurança e adequação da solução sob o foco do paradigma de ICN.

Desempenho: o provedor de conteúdo precisa computar os algoritmos de encriptação e de cálculo das chaves de reencriptação. De acordo com os resultados obtidos na validação, a carga extra imposta pelo cálculo das chaves de reencriptação, em comparação com outras alternativas, não deve representar um grande impacto na questão de desempenho do provedor de conteúdo. A quantidade de chaves de reencriptação que a fonte deve calcular depende da quantidade de usuários e da quantidade de conteúdo que essa fonte disponibiliza. Contudo, a chave é criada sob demanda, especificamente para o conteúdo que o usuário deseja acessar; desta forma, a fonte não desperdiça recursos ao calcular chaves de reencriptação que não serão utilizadas. De qualquer forma, pode-se assumir que os provedores de conteúdo podem superar eventuais problemas com desempenho ao adotar estratégias de balanceamento de carga. Do ponto de vista do cliente, a questão é a tarefa extra de reencriptar o conteúdo antes de decifrá-lo, em comparação aos métodos tradicionais de criptografia assimétrica. Enquanto que essa sobrecarga é razoável para *chunks* padrão de 4Kb, ela se torna inviável com tamanhos de *chunks* maiores e possivelmente impraticável com tamanhos de chaves maiores que 2048 *bits*, inclusive para *chunks* de 4Kb. Essa sobrecarga deve ser cuidadosamente investigada, principalmente ao levar em consideração que os usuários podem utilizar dispositivos móveis com recursos escassos de memória e processamento para acessar aos conteúdos da fonte. Uma forma de melhorar os tempos de computação para a reencriptação no cliente é realizar os cálculos ao receber o primeiro *chunk* e então reutilizá-los nos *chunks* seguintes. Além disso, como a função de reencriptar está no usuário, uma investigação sobre como associar as funções de reencriptação e decriptação de forma mais otimizada é desejável e está sendo investigada. Um outro ponto importante com relação ao desempenho está relacionado à reencriptação periódica do conteúdo com chaves diferentes, para realizar a revogação de chaves de reencriptação. Uma das questões que apoiam esse processo é um dos problemas abertos em ICN apontados em [Kutscher et al. 2014]. Nesse documento, levanta-se uma preocupação com relação à robustez das chaves públicas-privadas dos provedores de conteúdos contra ataques de força bruta, já que entidades maliciosas podem recuperar um conjunto relativamente grande de conteúdo criptografado com a mesma chave. Desta forma, a reencriptação periódica dos conteúdos pode ser relevante para evitar tais ataques.

Segurança: ao utilizar um esquema de criptografia assimétrica ao invés de criptografia simétrica, como tradicionalmente proposto para controle de acesso em ICN, torna-se potencialmente mais difícil que usuários não autorizados acessem conteúdos protegidos. Por exemplo, em soluções que empregam a criptografia simétrica, é suficiente que uma entidade maliciosa divulgue o segredo para corromper o conteúdo. Na solução proposta, seria necessário que um usuário legítimo divulgasse tanto a sua chave privada como a sua chave de reencriptação. Mesmo assim, somente o conteúdo relacionado àquela chave estaria corrompido. O provedor pode, simultaneamente, implementar medidas que restringem a quantidade de aplicações concomitantes com a mesma conta de usuário, tornando ainda

menos provável que os usuários divulguem suas chaves privadas e de reencriptação, sob o risco de serem penalizados. Apesar de os esquemas de reencriptação por *proxy* tradicionalmente considerarem os *proxies* entidades confiáveis, a solução proposta se abstém de tal asserção ao eliminar a entidade *proxy* da rede e transferir as funções de reencriptação para o usuário, que a executa através da aplicação. Desta forma, os usuários não têm incentivos para agir maliciosamente ao realizar as funções que antes eram atribuídas a um *proxy*. Além disso, a solução proposta permite que os conteúdos sejam armazenados em *cache* sem restrições, inclusive na presença de usuários maliciosos que de alguma forma descobrem o conteúdo em *cache* e o solicitam. Como não possuem a chave de reencriptação, não podem acessar o conteúdo. Além disso, se o usuário malicioso tentar solicitar a chave de reencriptação para a fonte, a fonte simplesmente nega o pedido, fazendo com que o usuário malicioso tenha o conteúdo mas não consiga acessá-lo.

Adequação à ICN: um dos principais objetivos da solução proposta é a adequação ao paradigma de ICN. Neste sentido, a solução não implica mudanças na arquitetura de ICN, já que somente os provedores de conteúdo e os usuários estão envolvidos nas ações de encriptação e decriptação do conteúdo. Como a rede não é carregada com requisitos específicos, ela fica livre para rotear e encaminhar os pacotes para quem quer que requisite, na sua melhor forma. Além disso, nenhuma função de segurança é transferida para elementos da rede: os roteadores não precisam verificar chaves ou validar políticas de acesso. Entretanto, o processo de revogação de chaves ainda implica em pelo menos uma desvantagem: por uma janela de tempo, o conteúdo antigo pode ser acessado por usuários que possuem a chave de reencriptação correspondente, caso o conteúdo antigo ainda esteja em algum *cache*. Além disso, para renovar o conteúdo disponível na rede, é necessário que o armazenamento de conteúdo nos *caches* tenha um tempo de vida limitado; de outra forma, os usuários continuam requisitando pelo conteúdo antigo e os *caches* nunca seriam renovados. De qualquer forma, o paradigma de ICN já prevê uma noção de atualidade para os conteúdos em *cache*, em que a fonte pode substituir por um conteúdo mais recente. Uma outra alternativa seria incorporar um carimbo de tempo no nome do conteúdo e configurar a aplicação para que requisite o conteúdo com o carimbo de tempo apropriado. De qualquer forma, a questão da revogação de chaves necessita de uma investigação mais profunda.

7. Conclusão

Este trabalho propôs uma solução de controle de acesso baseada em reencriptação por *proxy* que permite que somente usuários autorizados possam acessar conteúdos em uma arquitetura de ICN, mesmo na presença de entidades maliciosas. Além disso, a solução proposta garante os benefícios do uso do *cache* e não introduz mudanças significativas nos provedores e na rede. Cada conteúdo é encriptado com uma chave pública correspondente; para acessar o conteúdo, os usuários devem requisitar uma chave de reencriptação para o provedor. Desta forma, o provedor de conteúdo tem um controle de acesso ativo para o conteúdo, permitindo ou negando a chave de reencriptação de acordo com suas políticas. Mesmo que uma entidade maliciosa recupere o conteúdo e uma chave de reencriptação, ainda assim não é possível que ela acesse o conteúdo. As simulações realizadas mostram que a solução proposta apresenta uma sobrecarga mínima nos provedores de conteúdo e nos usuários. Como trabalhos futuros, planeja-se refinar a solução para invalidação de chaves de reencriptação, além de simular a solução proposta com diferentes políticas de *cache*. Planeja-se ainda explorar a junção das funções de reencriptação e decriptação no usuário como forma de melhorar o desempenho do usuário ao decriptar um conteúdo.

Referências

Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and Ohlman, B. (2012). A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36.

Ateniese, G., Fu, K., Green, M., and Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transaction on Information System Security*, 9(1):1–30.

Brito, G. M. d., Velloso, P. B., and Moraes, I. M. (2012). *Redes Orientadas a Conteúdo: Um Novo Paradigma para a Internet*, chapter 5, pages 211–264. Minicursos do XXX Simpósio Brasileiro de Redes de Computadores de Sistemas Distribuídos.

Chow, S., Weng, J., Yang, Y., and Deng, R. (2010). Efficient unidirectional proxy re-encryption. In Bernstein, D. and Lange, T., editors, *Progress in Cryptology – AFRICACRYPT 2010*, volume 6055 of *Lecture Notes in Computer Science*, pages 316–332.

Fotiou, N., Marias, G. F., and Polyzos, G. C. (2012). Access control enforcement delegation for information-centric networking architectures. In *2nd ACM SIGCOMM Workshop on Information-centric networking (ICN '12)*, pages 85–90.

Hamdane, B., Msahli, M., Serhrouchni, A., and El Fatmi, S. (2013). Data-based access control in named data networking. In *9th International Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom '13)*, pages 531–536.

Ion, M., Zhang, J., and Schooler, E. (2013). Toward content-centric privacy in ICN: attribute-based encryption and routing. In *3rd ACM SIGCOMM Workshop on Information-centric networking (ICN '13)*, pages 39–40.

Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M., Briggs, N., and Braynard, R. (2012). Networking named content. *Communications of the ACM*, 55(1):117–124.

Kissel, Z. and Wang, J. (2013). Access control for untrusted content distribution clouds using unidirectional re-encryption. In *2013 International Conference on High Performance Computing and Simulation (HPCS)*, pages 49–56.

Kutscher, D., Pentikousis, K., Psaras, I., Corujo, D., Saucez, D., Schmidt, T., and Waehlisch, M. (2014). ICN research challenges. http://www.ietf.org/id/draft-kutscher-icnrg-challenges-02.txt. Work in progress.

Misra, S., Tourani, R., and Majd, N. E. (2013). Secure content delivery in information-centric networks: design, implementation, and analyses. In *3rd ACM SIGCOMM workshop on Information-centric networking (ICN '13)*, pages 73–78.

Papanis, J. P., Papapanagiotou, S. I., Mousas, A. S., Lioudakis, G. V., Kaklamani, D. I., and Venieris, I. S. (2013). On the use of attribute-based encryption for multimedia content protection over information-centric networks. *Transactions on Emerging Telecommunications Technologies*, pages 1–14.

Salsano, S., Detti, A., Cancellieri, M., Pomposini, M., and Blefari-Melazzi, N. (2012). Transport-layer issues in information centric networks. In *2nd Edition of the ICN Workshop on Information-centric Networking*, ICN '12, pages 19–24. ACM.

Singh, S., Puri, A., Singh, S. S., Vaish, A., and Venkatesan, S. (2012). A trust based approach for secure access control in information centric network. *International Journal of Information and Network Security (IJINS)*, 1(2):97–104.

Smetters, D. and Jacobson, V. (2009). Securing network content. Technical report, PARC TR-2009-1.

Wood, C. and Uzun, E. (2014). Flexible end-to-end content security in ccn. In *IEEE Consumer Communications and Networking Conference*, CCNC '14, pages 1–8.

Xiong, H., Zhang, X., Zhu, W., and Yao, D. (2012). Cloudseal: End-to-end content protection in cloud-based storage and delivery services. In *Security and Privacy in Communication Networks*, volume 96, pages 491–500.

