A Scalable Approach for Managing Access Control in Information Centric Networks

Rafael Hansen da Silva, Weverton Luis da Costa Cordeiro,
Luciano Paschoal Gaspary

Institute of Informatics – Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500 – 91.501-970 – Porto Alegre, RS, Brazil
{rhsilva,weverton.cordeiro,paschoal}@inf.ufrgs.br

***Abstract*—One of the main challenges in Information Centric Networks (ICN) is providing access control to content publication and retrieval. Most of the existing approaches often consider a single user acting as publisher within a group. When dealing with multiple publishers, they may lead to a combinatorial explosion of cryptographic keys. Approaches that focus on multiple publishers, on the other hand, rely on specific network architectures and/or changes to operate. In this paper we propose a novel solution, supported by attribute-based encryption, for managing content access control. In our solution, we introduce secure content distribution groups, in which any member user can publish to and retrieve from. Unlike previous work, our 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 of content dissemination. In contrast to existing approaches, it offers higher access control flexibility, while reducing key management process complexity (in some scenarios, resulting in 97% less keys and objects in the network).**

I. Introduction

The Information Centric Network (ICN) paradigm has emerged as a promising direction for reshaping content distribution in the Future Internet [1]. In spite of its potentialities – for example, to make content distribution efficient and scalable, and to decrease data traffic in the network backbone [2] – there are several challenges that remain to be addressed. One of the most important, and critical for the success of ICN, is related to access control [1]. As contents are retrieved from distributed in-network caches, security mechanisms in place need to ensure that protected contents (*i.e.*, published with access restrictions) are consumed only by authorized users.

Existing approaches often focus on scenarios in which content sharing groups are composed of one publisher and several consumers [3], [4], thus being specially interesting for content providers such as YouTube, Google Play, iTunes Store, and NetFlix. However, those approaches might lead to a combinatorial explosion of cryptographic keys, if adopted on scenarios in which groups formed by multiple publishers and consumers are the norm. This overwhelming number of cryptographic keys might also increase resource overhead (*e.g.*, on network traffic, cache occupancy, etc.) or strain network operations (*e.g.*, for managing key lifecycle).

On the other hand, approaches focused on multiple publishers introduce novel components and/or entities to the network, for implementing content re-encryption or access control [5], [6]. Although effective, they are intrusive and less flexible for incremental adoption, vulnerable to malicious behavior of those entities and, in some cases, tailored for some specific ICN architectures or implementations.

To bridge this gap, in this paper we introduce a novel security model for managing user-generated content access control in ICN. The proposed model takes advantage of attribute-based encryption, and introduces the concept of secure content sharing groups, with user membership required for retrieving protected contents from groups. Access to published contents may be also refined using users' attributes, so that content retrieval can be further restricted to a specific subset of member users. The model reconciles support to multiple publishers and architecture-agnostic access control. As a result, the number of encryption keys is kept linearly proportional to the number of group members, without dependency on centralized entities. Our model is evaluated with regard to both the support for multiple publishers and operation overhead. The results achieved, through experiments in a controlled environment, have confirmed the effectiveness of our model, which introduces a minimal cost for publishing and retrieving contents (compared to existing approaches), while making content access control more robust and scalable.

The remainder of this paper is structured as follows. In Section II we discuss the most prominent related work. In Section III we describe our solution for secure content access control in ICN, while in Section IV we discuss the experimental environment used for evaluating our solution and our major findings. We close the paper in Section V with concluding remarks and directions for future research.

II. Related Work

Encryption is the most fundamental mechanism to implement secure and private content publication [7]. Nevertheless, symmetric and asymmetric encryption mechanisms are not sufficient if used separately in ICN: while the former requires some external resource (*e.g.*, phone or e-mail) for key distribution, the latter undermines in-networking caching facilities and makes content access control and lifecycle management increasingly complex, as the content must be encrypted for every target user.

In general, existing approaches attempt to maximize the use of in-network caches and minimize access control complexity.

978-3-901882-89-0 @2017 IFIP

In spite of that, they are different regarding the cardinality (of content publishing), the intrusiveness (*i.e.*, introduction or modification of network components), and to the encryption scheme used for protecting contents. Table I presents a general view of existing solutions, organized following these criteria.

Misra *et al.* [3] and Papanis *et al.* [4] propose that providers protect contents using symmetric encryption. As a result, both approaches take the most advantage of in-network caching. They are different however regarding the mechanism used for content access key distribution: while Misra *et al.* adopt the concept of broadcast encryption, Papanis *et al.* use Ciphertext-Policy Attribute-based Encryption (CP-ABE) [13]. Both assume only one publisher in the secure sharing group. Since each publisher needs to create a pair of keys for every user that should have access to contents, in a scenario in which everyone could act as a publisher, the required number of key pairs would be proportional to $\binom{n}{2}$.

Wood and Uzun [8] and Mannes *et al.* [9] also follow the single publisher model, although both use the proxy re-encryption technique, originally proposed by Ateniese *et al.* [14]. This technique uses proxy entities in the network to convert a content encrypted using the provider's public key into another encrypted using the user's public key. The main difference between both approaches lies in the intrusiveness criterion: while the solution of Wood and Uzun relies on intermediate nodes to redistribute re-encryption keys, Mannes *et al.* enable publishers themselves to implement that role. Although less intrusive, the latter requires that the content publisher be permanently available, for creating and distributing re-encryption keys upon content access.

The other approaches advance in the cardinality criterion, enabling multiple publishers in the same secure sharing group, thus avoiding the combinatorial explosion of cryptographic keys. However, they depend on the addition of entities to store contents and/or perform access control, which implies in modifications to the ICN architecture or in the dependency on the availability of those entities. The approaches of Singh *et al.* [5] and Ghali *et al.* [12] are particularly dependent on the honest behavior of these entities; if subverted, the privacy of contents controlled by them may be compromised.

An important aspect that must be observed in Table I is that none of the existing proposals aggregates the characteristics of non intrusiveness and native support to multiple publishers. In the following section, we present a security model that satisfies these requirements, without depending on specific components of the underlying architecture, and without increasing the complexity of the key management process.

III. Managing Content Access Control for Native Support of Multiple Publishers in ICN

Our design for secure content access control in ICN is built upon the notion that any user may act as content publisher and consumer, and seeks to minimize overhead (e.g. in terms of required cryptographic keys) to this end. The research challenges we thus approach in this paper, discussed next, target the conception of a solution agnostic of underlying ICN architecture, and flexible enough for incremental deployment (e.g. co-existing with legacy access control approaches).

We present in Figure 1 our security model for managing content access control in ICN. Secure content sharing is initiated when a user interacts with an instance of the *ICN Application*, running within his/her own device, to *create a group*. The ICN application corresponds to a piece of software that enables content sharing through the ICN paradigm, extended to support our model.

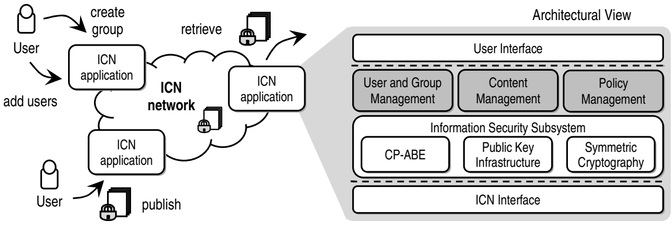


Fig. 1: Architectural view of our solution.

Only the user who created the group (referred to as *administrator* or *admin* in the remainder of this paper) may *add users* to it. As it will be discussed further in the following subsection, the process of adding users comprises the creation of a membership credential (a private key) and the delivery of that credential to the user. The key delivery must be done securely, for example, through use of an asymmetric encryption mechanism. Once enabled as a group member, the user may securely share contents (*i.e.*, *publish* and *retrieve* contents) with each other.

The architectural view presented in the right side of Figure 1 highlights (in gray) the components that belong to our proposal. The *User and Group Management* component aggregates the functionalities of group creation and membership control. The *Content Management* component is related to the secure content publishing and retrieval. Finally, the *Policy Management* component enables fine-grained access control, supporting for example granting and revoking content access. The *Information Security Subsystem* provides cryptographic primitives for user, group and content management, and borrows the idea of attribute-based encryption for policy definition; this subsystem is materialized by a symmetric encryption module, a Public Key Infrastructure (PKI) solution, and an attribute-based encryption mechanism (CP-ABE) [13] (also referred to as *CP-ABE component*). Observe that the components that form our solution are placecd exclusively between the user interface and the ICN network layers, thus being restricted to the software that runs on the user device.

Subsections III-A, III-B, and III-C describe, in more detail, the functionalities provided by each of the components highlighted in the architectural view from Figure 1. Subsection III-D closes the presentation of our proposal, by discussing possible attack strategies against our solution. For the discussion that follows, we adopt the set of notations and conventions summarized in Table II.

A. User and Group Management

Figure 2 illustrates the maintenance process of secure content sharing groups, highlighting the activities of group creation and membership management.

TABLE I: Proposals for content access control, organized according to cardinality, the intrusiveness, and encryption scheme.

Proposals	Cardinality		Intrusiveness		Encryption scheme	
	one publisher	multiple publishers	non intrusive	intrusive	symmetric	asymmetric
Misra et al. [3]	x		x		x	
Papanis et al. [4]	x		x		x	
Wood e Uzun [8]	x			x	x	
Mannes et al. [9]	x		x			x
Singh et al. [5]		x		x	none	
Fotiou et al. [10]		x		x	x	
Hamdane et al. [11]		x		x	x	
Ghali et al. [12]		x		x	none	

TABLE II: List of notations and conventions related to our security model.

Notation	Description
Entities and sets	
C	Original plain content
G	Secure content sharing group
U	User (member of a group)
$\mathbb{P}$	Access policy
$\mathcal{L}_G$	Attribute set from group G
$\mathcal{L}_{U,G} \subseteq \mathcal{L}_G$	User U's attribute set in group G
Crypto keys	
K_s	Symmetric encryption key
K_U	User U's public key
K_U^{-1}	User U's private key
K_G	Public key of the group G
M_G	Master key of the group G
$K_{\mathcal{L}_{U,G}}^{-1}$	User U's private key in group G
Crypto functions	
$\{X\}_{K_x}$	X encrypted using key K_x
$\{X\}_{(K_G,P)}$	X encrypted using group key K_G and policy P
Security Model	
$\widehat{X}$	Identifier of element X
C_P	Content C, protected
H_C	Enabler block of a protected content C_P

Group creation. As briefly described earlier, the admin initiates this process by interacting with the ICN application (flow 1 in Figure 2). This process basically comprises creating a pair of public K_G and master M_G keys for the group, which is done with the support of the CP-ABE component. The group comes to existence in the network once the public key K_G is disseminated in the network, as an object, using as identifier the group name (flow 2). The master key M_G must be kept private by the admin.

Each group has an attribute set $\mathcal{L}_G$, which are used to describe the member users. Each attribute is a string of arbitrary length, defined by the admin and that exist only in the scope of that group. For example, supposing a university group, possible attributes would be "Professor", "Undergrad", "M.Sc.", "Ph.D.", and "Post-doc". Note that there is no general rule for composing attributes, and their semantics can be drawn from the group context. The attribute list must also be published as an object in the network (flow 3).

Adding users to the group. The administrator initiates this process, through the ICN application (flow 4), by informing the attributes the new member will possess. This process unfolds into three steps: (*i*) creating the user's private key in the group $K_{\mathcal{L}_{U,G}}^{-1}$; (*ii*) publishing the key $K_{\mathcal{L}_{U,G}}^{-1}$ in a secure fashion in the network, so that only the intended user (the one that will be added to the group) can retrieve and decript it; and (*iii*) obtaining the key $K_{\mathcal{L}_{U,G}}^{-1}$ from the network (this last step performed by the user being added). These steps are described in detail next.

First, the private key $K_{\mathcal{L}_{U,G}}^{-1}$ is created with the support of the CP-ABE component. To this end, the admin must specify an attribute set $\mathcal{L}_{U,G} \subseteq \mathcal{L}_G$ for the user. The creation of $K_{\mathcal{L}_{U,G}}^{-1}$ also requires the master group key M_G. In the second step, the key $K_{\mathcal{L}_{U,G}}^{-1}$ (and the incorporated attributes $\mathcal{L}_{U,G}$) is then published in the network, so the target user may retrieve it. The delivery must occur in private, since the possession of $K_{\mathcal{L}_{U,G}}^{-1}$ materializes group membership. In other words, $K_{\mathcal{L}_{U,G}}^{-1}$ is employed for retrieveing protected contents published in the group (as it will be discussed in the following subsection). To perform this delivery, the admin must retrieve from the network the user's public key K_U and verify it using some PKI mechanism (flows 5 and 6). The key $K_{\mathcal{L}_{U,G}}^{-1}$ is encrypted using K_U, thus yielding the encrypted key $\{K_{\mathcal{L}_{U,G}}^{-1}\}_{K_U}$, which is published as a network object (flow 7).

In the third step, the target user needs to obtain $K_{\mathcal{L}_{U,G}}^{-1}$ and K_G from the network, so that he/she may use them to publish/consume contents to/from the group. The user starts this procedure (flow 8) by specifying the group he/she would like to join. The application then retrieves the group public key K_G (flows 9 and 10), and the user's private key in the group, which is encrypted $\{K_{\mathcal{L}_{U,G}}^{-1}\}_{K_U}$ (flows 11 and 12). The user's private key in the group is decrypted using his/her own private key K_U^{-1}. From this point on, the user is enabled to publish and consume contents from the group. Please note the group admin also needs to add himself/herself to the group, *i.e.*, also needs to create his/her own key $K_{\mathcal{L}_{U,G}}^{-1}$, if he/she desires to publish and consume contents from the group.

B. Content Management

The content management lifecycle comprises procedures related to secure content publication and retrieval. These procedures are supported by two elements: *protected content* and *enabler block*. A protected content corresponds to an encrypted content using symmetric encryption, whereas an enabler block contains the key required to decrypt a given content. In our model, a protected content has one (and only one) corresponding enabler block.

Formally, a protected content is a tuple $C_P = \langle \{C\}_{K_s}, \widehat{H_C} \rangle$, in which $\{C\}_{K_s}$ corresponds to the original content C, encrypted using the symmetric key K_s, and $\widehat{H_C}$ is the identifier of the enabler block of that protected content.

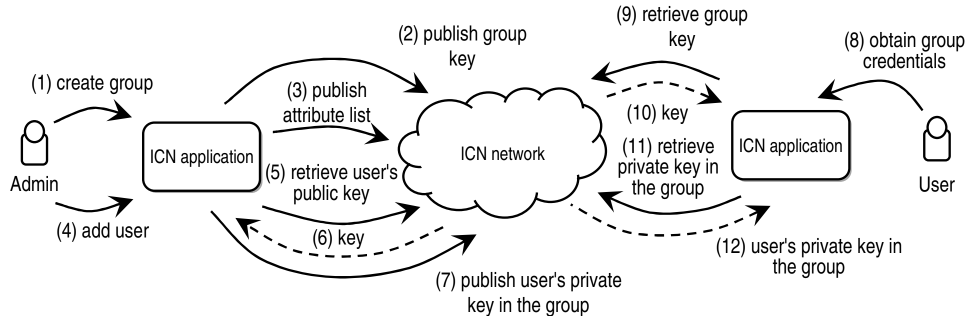

(1) create group
Admin
(4) add user
ICN application
(2) publish group key
(3) publish attribute list
(5) retrieve user's public key
(6) key
(7) publish user's private key in the group
ICN network
(9) retrieve group key
(10) key
(11) retrieve private key in the group
(12) user's private key in the group
ICN application
(8) obtain group credentials
User

Fig. 2: Maintenance process of secure sharing groups.

An enabler block is a tuple $H_C = \langle \{K_s\}_{(K_G,\mathbb{P})}, \widehat{K_G} \rangle$, in which $\{K_s\}_{(K_G,\mathbb{P})}$ corresponds to the key K_s (used to encrypt C) encrypted using (*i*) the group public key K_G and (*ii*) an access policy $\mathbb{P}$, and $\widehat{K_G}$ is the identifier of the group public key. Observe that this design enables various contents to be published using a single enabler block. This characteristic may be convenient if one wishes to publish several contents using a single access policy.

The process to build an enabler block comprises (*i*) the definition of the symmetric key that will be used to encrypt the content and (*ii*) the specification (by the user) of the *access control policy* $\mathbb{P}$. With regard to the symmetric key, it may be generated automatically, by the ICN application, or informed by the user. As for the policy $\mathbb{P}$, it determines which group members are authorized to decrypt the content, and is specified using elements from the group attribute list $\mathcal{L}_G$.

To illustrate the concept of access policies, recall the previously mentioned university content sharing group, in which users possess one (or more) of the following attributes: $\mathcal{L}_G = \{$Professor, Post-doc, Undergrad, M.Sc., Ph.D.$\}$. The user may, for example, specify a policy $\mathbb{P} = \{$Professor **or** Undergrad$\}$. In this case, the attribute-based decryption (using the CP-ABE component) may be carried out only by those users that possess the attributes “Professor” or “Undergrad” (or both). The methodology for forming these policies is discussed in more detail in Subsection III-C. Once $\mathbb{P}$ is determined, the CP-ABE is employed to encrypt K_s. This encryption is carried out using the group public key K_G and the policy $\mathbb{P}$. The encrypted key $\{K_s\}_{(K_G,\mathbb{P})}$ is then encapsulated within H_C.

After the enabler block is built, the protected content may then be formed. Its original construction comprises the encryption of the original content C to be disseminated in the network. For that encryption process, the symmetric key K_s, encapsulated within the enabler block, is used.

Content publication. Figure 3(a) illustrates the dynamics of a content publishing process. The user initiates this process by interacting with the ICN application (flow 1 in Figure 3(a)), by informing the content C to be published. At this point, five steps are executed. First, the application resorts to the network to obtain an updated version of the group attribute list $\mathcal{L}_G$ (flows 2 and 3), and makes that list available to the user. Then, the user creates an access policy $\mathbb{P}$, according to the desired access restrictions (flows 4 and 5). The third step, carried out by the application, consists in encrypting the content C using a symmetric key K_s. The fourth step corresponds to the construction of the enabler block H_C of the content, as previously discussed. In the last step, the protected content C_P is built, encapsulating the encrypted content $\{C\}_{K_s}$ and the identifier to the enabler block $\widehat{H_C}$. Finally, both the protected content C_P and the enabler block H_C are published in the network (flows 6 and 7).

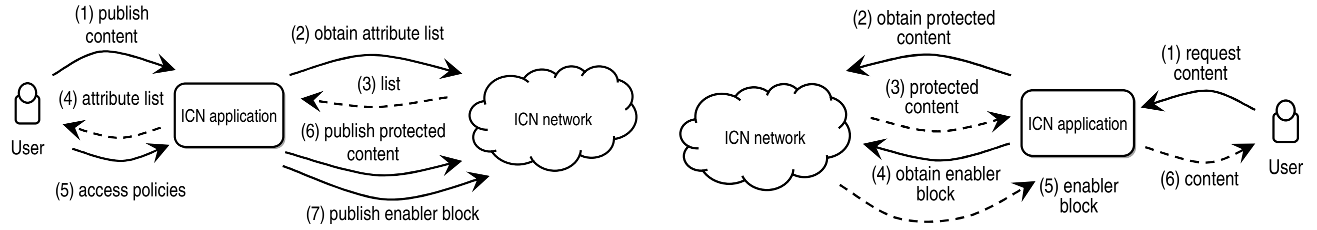
Content retrieval. The retrieval process, illustrated in Figure 3(b), initiates when the user requests a content (flow 1). The application requests to the network the corresponding protected content C_P (flows 2 and 3), which is obtained from the nearest source. When opening C_P, the application identifies which enabler block H_C is related to that content (through the identifier $\widehat{H_C}$ contained in the tuple). The application then requests H_C to the network (flows 4 and 5) to retrieve the encrypted symmetric key $\{K_s\}_{(K_G,\mathbb{P})}$. The retrieved key $\{K_s\}_{(K_G,\mathbb{P})}$ is submitted to the CP-ABE component for decryption. To this end, the user uses his/her own private key in the group $K^{-1}_{\mathcal{L}_{U,G}}$. The key $\{K_s\}_{(K_G,\mathbb{P})}$ is decrypted iff the access policy $\mathbb{P}$ used to encrypt it is compatible with the attributes used by the group admin upon the creation of $K^{-1}_{\mathcal{L}_{U,G}}$ (when adding the user to the group). In case it is successfully decrypted, the plain content is delivered to the user (flow 6).

C. Policy Management

The proposed security model, implemented by the *Policy Management* component, enables to determine when and which users may have access to published contents. In other words, the model aggregates mechanisms that enable one to assign, limit, and revoke content access authorizations, based on defined policies.

One may use relational ($>$, $<$, $=$) and logical (**and**, **or**) operators to write policies. With support of the CP-ABE component [13], those operators enable one to determine when and what users have access to a content. Figure 4 presents the set of rules that define the policy construction process. In this set, `<attribute>` $\in \mathcal{L}_G$ and `<integer>` $\in \mathbb{N}$.

Granting access to contents. It consists of defining a policy (using one or more attributes) that a subset of group members must satisfy to decrypt a content. In order to gain access to the content, the user's private key in the group $K^{-1}_{\mathcal{L}_{U,G}}$ must satisfy the restriction described by the policy. For example, suppose two users: *John Smith*, having attributes



(a) Publication

(b) Retrieval

Fig. 3: Step-by-step process for content publication and retrieval.

```
⟨policy⟩ ::= ⟨attribute⟩ | '(' ⟨policy⟩ ')'
      | ⟨attribute⟩ and ⟨policy⟩ | ⟨attribute⟩ or ⟨policy⟩
      | ⟨attribute⟩ = ⟨integer⟩ | ⟨attribute⟩ < ⟨integer⟩ | ⟨attribute⟩ > ⟨integer⟩
      | ⟨integer⟩ of '(' ⟨collection⟩ ')'

⟨collection⟩ ::= ⟨policy⟩ ',' ⟨policy⟩ | ⟨collection⟩
```

Fig. 4: Rules for defining content access policies.

$\mathcal{L}_{\text{John Smith},G} = \{\text{Professor, Researcher}\}$, and *Jane Smith*, having attributes $\mathcal{L}_{\text{Jane Smith},G} = \{\text{Student, Researcher}\}$. An access policy $\mathbb{P}_1 = \{\text{Professor}\}$ enables content access for *John Smith* only. An access policy $\mathbb{P}_2 = \{\text{Researcher}\}$, in turn, enables access to both. The rules presented earlier do not allow the construction of "wildcard" policies, *i.e.*, for universal access. One way to reach all users is citing one by one their attributes, in the access policy. The admin may also define an attribute common to all users (*e.g.*, "All") for reaching all users; in this case, every content publisher may use it in policies that aim at universal access.

User and policy attributes may also be valued. Such attributes may be created to indicate, for example, the user's level in the corporate hierarchy. Suppose that user *John Smith* has attribute "Level = 5" and *Jane Smith* has attribute "Level = 2". In case one wishes to publish a content solely to users from level 3 or higher (*John Smith*, in this example), it is sufficient to define a policy $\mathbb{P} = \{\text{Level} > 2\}$ (assuming that the hierarchy levels are given by discrete numbers).

Revoking access to contents. Revocation is based on the possibility to publish a newer version of an enabler block (for example, when the older version expire in the router cache). Therefore, the user may reformulate the policy of that block to restrict the content access for some particular user. There are two strategies that may be employed. The first one is to define a unique attribute for each user. In this case, revocation would require selecting all users except that/those whose access must be revoked. In a scenario with users *John Smith*, *Jane Smith*, *Alice Bloggs*, and *Bob Bloggs*, and a content published with $\mathbb{P} = \{\text{All}\}$, revoking the access of *John Smith* to this content requires that the policy be reformulated to $\mathbb{P}' = \{1\,\text{of}\,(\text{Jane Smith, Alice Bloggs, Bob Bloggs})\}$. The complexity of defining such restriction for groups with dozens of users or more can be trivially solved through a user interface, thus not being necessary to transport it to the model.

The second strategy for implementing content access revocation is through expiration, using the relational operators to compare attribute values. To illustrate, suppose that *John Smith* has the attribute "Created = 1435708800" (2015-07-01 00:00:00) and *Alice Bloggs*, "Created = 1446336000" (2015-11-01 00:00:00). The semantics of these attributes corresponds to the date and time (timestamp format) that each one was added to the group. Suppose now a content having the policy $\mathbb{P} = \{\text{Created} > 1420070400\}$. That policy grants access only to those users added to the group after January 1st, which applies to *John Smith* and *Alice Bloggs*. *John Smith*'s access may be revoked through expiration, in this case, by publishing a new enabler block with $\mathbb{P}' = \{\text{Created} > 1443657600\}$ (October 1st). We emphasize that both strategies may be used in conjunction, thus enabling short and long term revocation.

D. Possible Attack Strategies

Having presented an overview of our model, we now discuss about its robustness in face of possible strategies an attacker may take advantage of to gain access to protected contents. Firstly, it is important to emphasize that our model was designed considering three basic assumptions, often adopted by those systems that deal with access control: (*i*) the group admin is trustworthy; (*ii*) group members maintain in secret their respective private keys in the group $K^{-1}_{\mathcal{L}_{U,G}}$; and (*iii*) a member with access to a protected content does not advertise the symmetric key used to encrypt it.

The implications of the assumptions enumerated above are described next. The first one establishes that attribute assignment to users is done in a trustworthy fashion. It means that the admin will not assign attributes maliciously, for example, by assigning attributes to an adversary or even to users who are not compatible with them. That assumption is similar to the trust assigned to managers of strategic and/or secretive projects, for example, of open-source software (in which the admission of an adversary to the development team may lead to the inclusion of malicious code in the developed software). The second assumption is related to the security of contents which are accessible by some specific users. This assumption is equivalent to keeping in private access credentials to a system (*e.g.*, a key to an ssh server or the password to a pay-per-use content portal). The third assumption, in turn, implies in the privacy of protected contents that have been accessed by authorized users. In this case, leaking the symmetric key (obtained from the enabler block) is equivalent to leaking the plain content itself.

Considering these assumptions, an attacker may only subvert our security model only if he/she compromises (through either physical or remote access) the admin/user's device to obtain the group private key or the users' private keys in the group. This kind of attack is out of scope, as protection strategies against it require mechanisms for ensuring the security of admin/users' devices. Assuming their privacy is not compromised, our model remains resilient even if the attacker compromises the network (including routers, enabler blocks, public keys, attribute lists, and so on).

The proposed model is also robust to colluding attacks. For example, suppose a user with the attribute "Professor" and another with attribute "Researcher". Even if both users collude, they will not be able to decrypt a content protected under the policy $\mathbb{P} = \{\text{Professor } \textbf{and} \text{ Researcher}\}$, since the policy requires that a same group member possess both attributes simultaneously. As previously discussed, only users belonging to the group and that entirely satisfy the access control policies defined may access published contents. Finally, the model does not prevent that non-member users publish contents in the group. This is possible because publishing contents within a group only requires the group public key and attribute list, both available in the network. Group members may avoid access to undesired contents by verifying their origin using, for example, self-certification mechanisms provided by the network itself [7].

IV. Evaluation

In order to assess the efficacy and effectiveness of our model, various experiments were carried out in a controlled environment. The experiments had the goal of verifying the scalability of the model, its operational overhead, and the impact to users' quality of experience (QoE), in scenarios with a varying number of users acting as publishers and consumers. For comparison, we considered the solution of Papanis *et al.* [4] and a generic, secure content sharing solution based on RSA. The former was chosen as it allows, similarly to our proposal, that users be dynamically added to content dissemination pools (feature not supported by Misra *et al.* [3]), whereas the latter was chosen for its popularity. The other approaches were not considered as they are either based on asymmetric encryption (thus leading to results similar of RSA) or intrusive.

A. Environment Settings and Evaluation Scenarios

The proposed model was implemented over the CCN (*Content Centric Networking*) architecture [7], using as basis CCNx 0.8.2 running on top of Java SE v8 virtual machines. For the attribute based encryption mechanism, we used the *cpabe* 0.11 software [16]. With the goal of following the key management standard proposed for ICN architectures [17], each protected content is accompanied of a respective metadata, which contains the identifier of the corresponding enabler block and the content validity, among others. Similarly, each enabler block is accompanied by its respective metadata.

The physical substrate used in the experiments comprised two servers, each equipped with 1 Intel Xeon E5-2420 processor (1.9 GHz, 12 Threads, and 15MB cache), 32GB RAM memory (1,333 MHz), 1 HD SAS (1TB capacity), and 2 Gigabit Ethernet network cards. Both have Debian/Linux 7.7 (kernel 3.14.21) and Xen Hipervisor installed. The servers were directly connected to each other using two Ethernet cables. The logical topology used for the evaluation, a subset of Internet2, is illustrated in Figure 5. The mapping of logical elements to the physical substrate is also presented in the figure. Each logical node in the topology corresponds to one virtual machine; each of them was instantiated with the following settings: 2 virtual processors, 2 GB RAM memory, and 40 GB disk space. The network links between nodes were emulated using *bridge-utils* v1.5, all with ≈ 98Mbps.

For the experimental evaluation we considered two scenarios, whose relevant settings are summarized in Table III. For simplicity, all contents were published using a universal access policy (*i.e.*, any group member may decrypt it). That decision was based on preliminary experiments, which enabled us to observe that the number of attributes has marginal effect over operational costs (publication/retrieval time, network traffic, etc.) of our model. Finally, for each experiment, we carried out 30 runs and computed confidence intervals for a significance level $\alpha = 0.05$.

B. Operational Overhead

The first part of our evaluation comprised an analysis of the operational overhead of our solution in an environment composed of a publisher and a consumer only. To this end, we used a subset of the topology shown in Figure 5, formed by nodes #4 (publisher) and #6 (consumer). Figure 6 presents an overview of the results achieved for content publication (curves "Pub.") and retrieval (curves "Ret."). For the sake of comparison, we considered the solution of Papanis *et al.* (curves "Papanis") and one scenario without any security control mechanism (curves "Plain"). For legibility, the plots are shown with y axis on log scale.

The main conclusion one may draw from the results depicted in Figure 6 is that the overhead of our solution is marginal when compared to Papanis *et al.* Focusing on the average content dissemination time (Figure 6(a)), for example, our solution was 0.6% more efficient on average for publication. With regard to processing load (Figure 6(b)), there are slightly higher for our solution (0.5% for publication and 1.4% for retrieval). Finally, observe that the measured network traffic (Figure 6(c)) indicate similar performance of both solutions.

When comparing our solution to the no security control scenario, note that the operational overhead is amortized proportionally to the size of the published content. These results suggest that our solution incurs in a relatively small impact to the users' quality of experience (QoE). In the case of content publication, time overhead decreases from 1,400% (difference in the cost between our solution and "Plain"), on average (for 1 MB contents), to 72% (1,000 MB). In this comparison, the average overhead on retrieval time was only 8% (aspect of higher importance for the QoE of a large fraction of users). The results obtained for our model are similar to those observed for Papanis *et al.*

It is important to emphasize that the processing and publication/retrieval time overhead are mainly due to the use

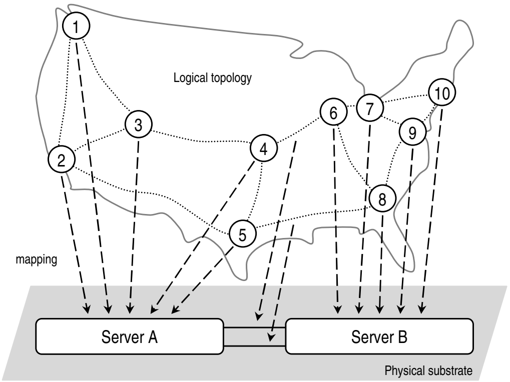
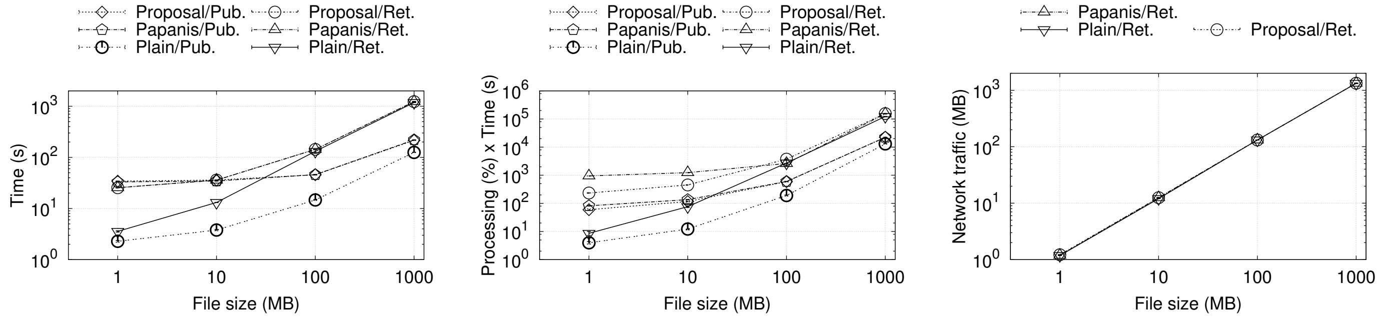


Fig. 5: Network topology considered.

TABLE III: Evaluation scenarios.

Parameters	Evaluated scenarios	
	A	B
Number of users	10	30
File size	100MB	100MB
Published files	10	30
Router cache capacity	1GB	1GB
Cache expiration	1 hour	1 hour
Chunk size	4KB	4KB
Content popularity	Zipf [1] $(s = 2.0)$	Zipf [1] $(s = 2.0)$

[1] According to *Pentikousis et al.* [15]



(a) Average dissemination time

(b) Processing load

(c) Network traffic

Fig. 6: Overhead of our solution considering publication and retrieval times, processing load, and network traffic generated.

of encryption to protect the content and access keys. When there is no security solution in place, the processing and time measured refer only to the content publication/retrieval in the network. About the measured network traffic (Figure 6(c)), the overhead was constant and marginal, corresponding mainly to the enabler block of the protected content (which is also disseminated on the network).

C. Amount of Keys and Objects Generated

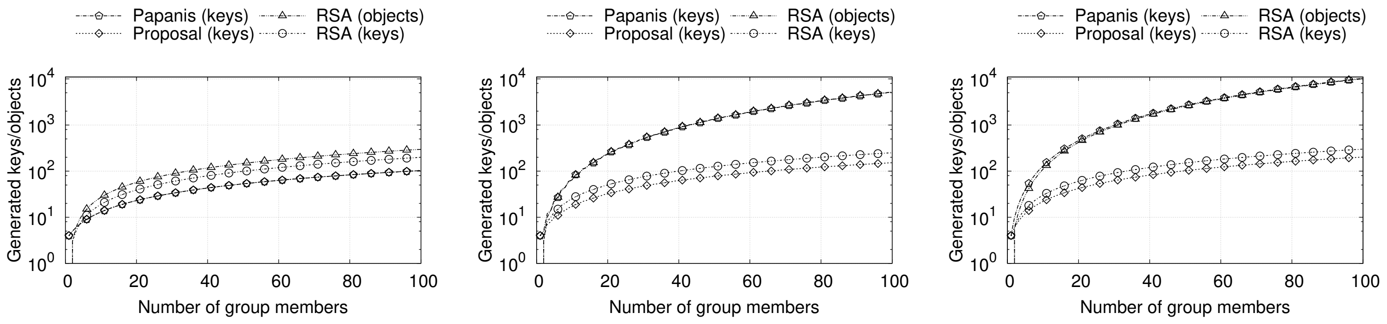
Another aspect evaluated is concerned to the number of keys/objects required for secure content dissemination, in the situations in which one, half, and all users act as publishers in the network, respectively. In this analysis were considered, in addition to our solution, the one of Papanis *et al.* [4] and the RSA based one. In the case of Papanis *et al.*, it is instantiated for each publisher. In the RSA based model, (*i*) each user has a pair of public and private keys; (*ii*) each content is encrypted using a unique symmetric key; and (*iii*) the content key is encrypted using each target user's public key.

The results for this evaluation are presented in Figure 7 (y axis is presented in log scale). Observe that our solution requires the lowest number of encryption keys, in contrast with Papanis *et al.* and RSA. More importantly, our solution maintains the number of keys proportional to the number of group members, regardless of the number of publishers within the group. Conversely, the number of keys required/objects published increases significantly for the former two proposals. For Papanis *et al.*, one may observe an increase of up to 9,900%, in contrast to 96% in the case of our solution. With regard to Papanis *et al.*, that increase is related to the issue of combinatorial explosion of cryptographic keys. Although in the scenario using RSA the number of keys remains relatively constant, the number of objects published in the network grows significantly. The reason is that, although the symmetric key is unique for each content, it needs to be encrypted individually for every target user, to ensure that only authorized users may access the content.

D. Dissemination Time and Number of Registered Objects

The goal of this evaluation, whose results are summarized in Figure 8, was to assert the performance of our solution. More specifically, we focused on the total time required for content dissemination, and the overhead to the routers' *Forward Information Base* (FIB), in an environment with multiple publishers and consumers. This evaluation considered as basis the complete logical topology illustrated in Figure 5 and the scenarios described in Table III. Each user publishes 1 and recovers n contents, *i.e.*, 10 contents are published in scenario A and 30 in scenario B. For comparison, we have considered the solutions of Papanis *et al.*, one based on RSA, and one case without security mechanism ("Plain").

One may observe from Figures 8(a) and 8(b) that the users' QoE (measured by the time required for content dissemination) is marginally affected in our solution, if compared to

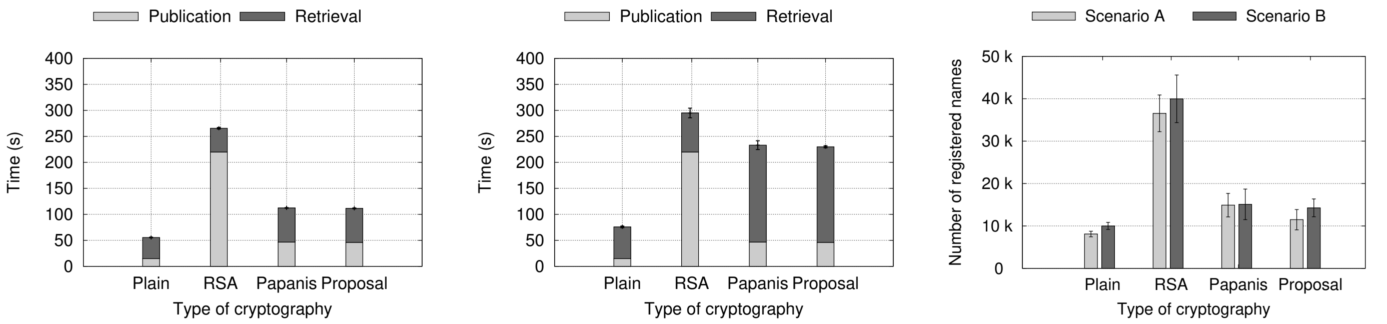


(a) 1 publisher

(b) $n/2$ publishers

(c) n publishers

Fig. 7: Number of keys/objects required for secure content sharing.



(a) Time measured on scenario A

(b) Time measured on scenario B

(c) FIB for Scenarios A and B

Fig. 8: Content publication/retrieval times, and number of names registered on the routers' FIBs, for scenarios A and B.

Papanis *et al.* More importantly, it is substantially better when compared to the RSA based solution. That performance is obtained causing relatively less impact to the network, as one may observe in Figure 8(c).

The relatively higher content retrieval time for our solution and Papanis *et al.* is explained by the fact that the RSA based solution does not require content or enabler block metadata. In other words, each user may directly locate contents and their respective keys without obtaining their metadata, thus being irrelevant to publish those metadata. On the other hand, publication time is significantly higher for RSA, since that n different encrypted versions of a same key for a same content must be published in the network, one for each target user. That aspect can be seen on the plot of Figure 8(c), as our solution reduces in up to 68% (for scenario B) the number of names registered on the routers' FIBs.

V. Final Considerations

Secure content publishing and retrieval in ICN is a reality, with several solutions that offer a wide range of access control capabilities. Although promising, some solutions cause a significant overhead to the network, as their key management (and distribution) schemes are subject to a combinatorial explosion of keys. Those not susceptible to this problem, however, rely on specific ICN architectures or deployments, insert (or modify) network components, and are less flexible for gradual adoption.

To fill in this gap, we presented a novel solution, centered on the concept of users' groups, for secure content sharing. From the results achieved, we assessed the efficacy and effectiveness of our solution. In summary, it requires a comparatively lower number of keys and objects in the network (in some cases, up to 97% less keys). That gain is reached without degrading users' quality of experience (*e.g.*, the time required to publish/retrieve contents), in contrast to what occurs for other solutions. In addition to these benefits, our solution may be independently and autonomously adopted by a subset of ICN users, without depending on modifications in the network. Finally, it enables content publishing and retrieval even if the group admin (or the content publisher, in the case of retrieval) becomes unavailable.

As prospective directions for future research, we intend to investigate mechanisms for speeding up the dissemination of novel content access control policies, as well as mechanisms for making access revocation simpler and more efficient. Although these are classical access control issues (and thus have been exhaustively investigated), they demand a novel approach for solving it in this context, because of features such as caching, etc.

Acknowledgements

This research work was supported in part by funding from Project 462091/2014-7, granted by CNPq (MCTI/CNPQ/Universal 14/2014 - Faixa A).

References

[1] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 1024–1049, Second 2014.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.

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

[4] J. P. Papanis, S. I. Papapanagiotou, A. S. Mousas, G. V. Lioudakis, D. I. Kaklamani, and I. S. Venieris, "On the use of attribute-based encryption for multimedia content protection over information-centric networks," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 4, pp. 422–435, 2014.

[5] S. Singh, A. Puri, S. S. Singh, A. Vaish, and S.Venkatesan, "A trust based approach for secure access control in information centric network," *Journal of Information and Network Security*, vol. 1, no. 2, pp. 97–104, 2012.

[6] N. Fotiou and G. C. Polyzos, "Securing content sharing over icn," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ser. ACM-ICN '16. New York, NY, USA: ACM, 2016, pp. 176–185.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," *Commun. ACM*, vol. 55, no. 1, pp. 117–124, Jan. 2012.

[8] C. Wood and E. Uzun, "Flexible end-to-end content security in ccn," in *11th Consumer Communications and Networking Conference (CCNC 2014)*, Jan 2014.

[9] E. Mannes, C. Maziero, L. C. Lassance, and F. Borges, "Controle de acesso baseado em reencriptação por proxy em redes centradas em informação (in portuguese)," in *Brazilian Symposium on Information and Computer Systems Security (SBSeg 2014)*, 2014.

[10] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," in *ACM SIGCOMM workshop on Information-centric networking (ICN '12)*, 2012, pp. 85–90. [Online]. Available: http://doi.acm.org/10.1145/2342488.2342507

[11] B. Hamdane, M. Msahli, A. Serhrouchni, and S. El Fatmi, "Data-based access control in named data networking," in *Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom 2013)*, Oct 2013, pp. 531–536.

[12] C. Ghali, M. A. Schlosberg, G. Tsudik, and C. A. Wood, "Interest-based access control for content centric networks (extended version)," *CoRR*, vol. abs/1505.06258, 2015.

[13] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy (SP 2007)*, 2007, pp. 321–334.

[14] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, Feb. 2006.

[15] K. Pentikousis, B. Ohlman, E. Davies, S. Spirou, G. Boggia, and P. Mahadevan, "Information-centric networking: Evaluation methodology draft-irtf-icnrg-evaluation-methodology-03," October 2015. [Online]. Available: https://tools.ietf.org/html/draft-irtf-icnrg-evaluation-methodology-03

[16] J. Bethencourt, A. Sahai, and B. Waters, "Advanced crypto software collection," April 2015. [Online]. Available: http://acsc.cs.utexas.edu/cpabe/

[17] C. Bian, Z. Zhu, A. Afanasyev, E. Uzun, and L. Zhang, "Deploying key management on ndn testbed," Tech. Rep., 2013. [Online]. Available: http://www.named-data.net/techreport/TR009-publishkey-rev2.pdf