Received: 20 March 2017 | Revised: 30 June 2017 | Accepted: 12 August 2017

DOI: 10.1002/nem.2007

SPECIAL ISSUE PAPER

WILEY

Rendezvous-based access control for information-centric architectures

Nikos Fotiou[1] | Bander A. Alzahrani[2]

[1]Mobile Multimedia Laboratory, Department of Informatics, School of Information Sciences and Technology, Athens University of Economics and Business, 76 Patision Athens, 10434, Greece

[2]College of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Correspondence
Nikos Fotiou, Mobile Multimedia Laboratory, Department of Informatics, School of Information Sciences and Technology, Athens University of Economics and Business, 76 Patision, Athens 10434, Greece.
Email: fotiou@aueb.gr

Funding information
Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, Grant/Award Number: G-80-611-38

Summary

Information-centric networking (ICN) has been in the spotlight of many research efforts as it shifts the focus from (endpoint) locations to content items themselves. By leveraging content centrism and by using content and content names as the main pillar of all (inter-)networking functions, ICN architectures are expected to overcome many of the limitations of the current Internet architecture. Information-centric networking paradigm also advocates a shift in security solutions: Instead of securing the communication channel, ICN security solutions should secure the content itself. Therefore, end users should be able to access content stored in various locations in the network—even unsecured—in a private and secure way. Similarly, content owners should not lose the governance of their content items, no matter the network location where they are stored. In this paper, we design, implement, and evaluate an access control delegation mechanism for the publish-subscribe Internet architecture. Our solution does not introduce any new entity; instead it allows semitrusted publish-subscribe Internet rendezvous points to enforce access control policies. Moreover, our solution leverages identity-based proxy re-encryption to protect content confidentiality in the presences of malicious publishers, ie, nodes that host content items and do not respect the access control decisions of the rendezvous point.

KEYWORDS

identity-based encryption, proxy re-encryption, publish-subscribe Internet

1 | INTRODUCTION

It is a fact that the Internet design has remained relatively unchanged since its inception. In its core, the Internet is a host-oriented network, where every request for content is translated into a physical host address location. However, users are not interested anymore in connecting to a device, instead they are interested in retrieving information regardless of its location. This shift in the Internet usage has created many shortcomings, rooted to the host-oriented design of the core of the Internet. These shortcomings include lack of effective multicast, inefficient mobility handling, insecurity, and scalability issues.[1] To mitigate these shortcomings, various add-on "patches" have been deployed, including peer-to-peer networking and content distribution networks. Nevertheless, it has become evident that continuously patching the Internet architecture is not sustainable and a more radical change is required. Towards this direction, a new (inter-)networking paradigm has emerged, the information-centric networking (ICN) paradigm.

Information-centric networking is a new (inter-)networking technology, embraced by many research teams around the globe. As the name implies, ICN postulates the use of information as the main primitive of all network functions.

Therefore, in an ICN-based architecture, all the core operations, including routing, forwarding, and security, are based on information identifiers instead of host location identifiers. In an ICN-based architecture, users reveal to the network the identifiers of the content items in which they are interested and the network undertakes the task to locate the desired items and retrieve them. Content items can be replicated and cached in many network locations. Moreover, requests for the same content can be aggregated, creating this way opportunities for multicast.

Various ICN architectures have been proposed by a growing body of researchers such as CCN[2] and the publish-subscribe Internet (PSI) ICN architecture.[3*] In this paper, we are focused PSI ICN architecture, originally created by the PSIRP[4] project and further extended by the PURSUIT[5] project.

1.1 | The challenges of access control in ICN

In the current Internet, content owners either own the network devices that host their content or they have direct business relationships with their owners. Therefore, they do not (completely) lose the governance over their content, ie, they are in position of removing their content from the network, as well as of controlling who can access it. On the other hand, in ICN architectures, this is not always the case. Information-centric networking facilitates content replication in many network locations, which may as well reside outside of the administration realm of the content owner. This property of ICN creates some unique challenges when it comes to access control:

- Access control policy management becomes harder, since an access control policy should be replicated in and enforced by many network entities.
- User privacy protection becomes harder; a user may have to reveal sensitive information to an unreliable entity for the latter to enforce an access control policy. This information may as well include user credentials.
- Content protection against unauthorized access becomes weaker. Since a node that hosts a content item may not have business relationship with a content owner, it does not have any incentive to respect owner's access control polices.

1.2 | Our contributions

In this work, we design, implement, and evaluate an access control solution for the PSI ICN architecture. Although our solution is specific to PSI, it can be generalized to any rendezvous-based ICN architecture, ie, an architecture where a centralized network entity mediates content demand and supply. Our solution is built on our previous works described in Fotiou et al[6] and Fotiou and Polyzos.[7] In particular, the security mechanisms discussed in Fotiou and Polyzos[7] are applied to the solution presented in Fotiou et al[6] so as to remove the need for a trusted entity (in Fotiou et al,[6] this entity is referred to as the *Access Control Provider*). As a consequence, this work adds significant less communication overhead compared to Fotiou et al.[6]

The contributions of this paper can be summarized in the followings:

1. We design a solution that allows a semitrusted *rendezvous point* (ie, network nodes that perform information lookup) to enforce access control policies, without having access to users' credentials (or any other sensitive information).
2. We leverage identity-based proxy re-encryption (PRE) to protect content against *publishers* (ie, network nodes that disseminate content) that do not respect access control decisions.
3. We use content encryption independently of the rules of an access control policy, allowing for fast access control policy modifications, ie, in our solution policy modifications (such as user addition or revocation) does not require any re-encryption of the already encrypted content.

2 | BACKGROUND

2.1 | The PSI ICN architecture

The PSI[3] architecture is built on top of the so-called publish-subscribe paradigm, which allows producers to advertise and publish their content and consumers to subscribe to content. The main entities of the PSI architecture are the following:

*For a survey on information-centric research, interested readers are referred to Xylomenos et al[8]

- Content owner: This is a real-word entity that creates, names, and owns a piece of content.
- Subscriber: This is the network device of a user who is interested in a piece of content.
- Publisher: This is a network device that hosts a piece of content. A publisher may have in a direct business relationship with a content owner; therefore, it may have been appointed to host a content item, or it may act on its own (eg, a network cache).

All content items in the PSI architecture are uniquely identified by two of identifiers, namely, the rendezvous identifier (*RId*) and the scope identifier (*SId*). *SId*s are globally unique, and *RId*s are unique within a scope. *SId*s are used to give a "hint" about the network location of a content item. In particular, each *SId* is managed by a lookup node known as the *rendezvous node* (RN). We refer to the RN that manages an *SId* as the rendezvous point (RV) of the *SId*. The RV of an *SId* maintains a data structure that maps the *RId*s that belong to that *SId* into publisher's network locations. A scope may contain other scopes, creating this way a hierarchy of scopes. Moreover, a scope may be governed by an access control policy, which should be enforced by its RV.

All RNs are interconnected in a network known as the *rendezvous network*. A rendezvous network may be flat (eg, a distributed hash table) or hierarchical (eg, following a domain name service (DNS)–like structure). The RN records are populated using content "advertisements" issued by the publishers. The state created by these advertisements is also used for routing content "subscriptions" to the appropriate RN, ie, the RV of the scope that subscription concerns. When a subscription reaches an RV, the RV creates a forwarding path from a publisher to the subscriber(s). This path is contracted with the help of the *topology manager*: a network entity that is aware of the full topology graph. Then, the RV notifies a publisher about a subscription by including in the notification message an identifier of the forwarding path, known as the *forwarding identifier* (FId). All these interactions between the PSI network entities are illustrated in Figure 1. In this figure, there is a publisher, a subscriber, and a rendezvous network composed of 3 RNs. The middle RN manages a scope identified by "SId1/SId2," ie, this node is the RV for that scope. The publisher hosts an item with RId and RId1 and wants to advertise it under the scope SId1/SId2 (step 1). Hence, after this advertisement, this item will be uniquely identified by the pair (SId1/SId2, RId1). The subscriber issues a subscription message containing the unique item identifier (step 2), which is routed to the scope RV. Now, the RV computes a forwarding path from the publisher to the subscriber, assigns an identifier to that path (namely, *FId*), and sends a notification to the publisher (step 3). Then the publisher "publishes" the desired item, which is forwarded to the subscriber (step 4).

An implementation of the PSI architecture, code-named Blackadder,[9] is provided as an open source software.

2.2 | Identity-based PRE

Identity-based encryption (IBE) is a form of encryption where an arbitrary identity is used as a public key. An IBE ciphertext can only be decrypted by the owner of the corresponding secret key. Identity-based encryption schemes are typically composed of the following 4 algorithms:

- Setup: This algorithm is used by a trusted entity known as the *private key generator* (PKG). Setup generates a master-secret key (*MSK*) and some system parameters (*SP*). The *MSK* is securely stored by the PKG, whereas SP are published to everyone.

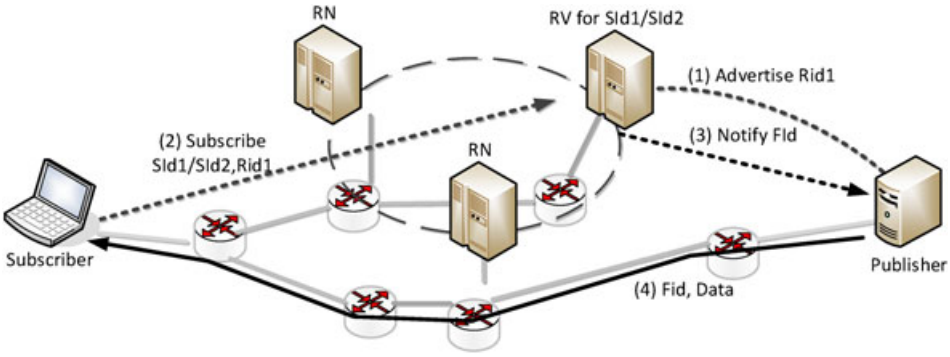

FIGURE 1 The publish-subscribe Internet architecture

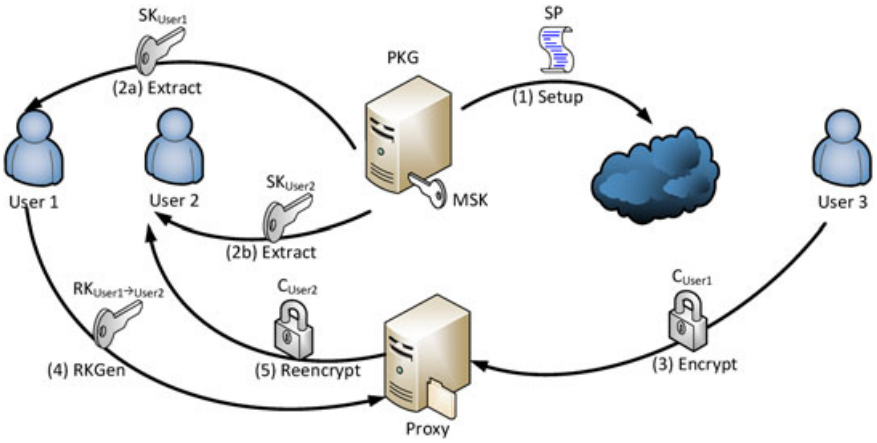

FIGURE 2 The IBE-PRE scheme of Ateniese and Green

- Extract: This algorithm is also executed by a PKG and generates the secret key SK_{ID} that corresponds to the identity ID_{user}. The key generation process uses as input SP, MSK, and ID_{user}.
- Encrypt: This is the main encryption algorithm, and it can be invoked by anybody. The encryption process takes as input an identifier ID_{user} and its corresponding SP and a message M. It generates a ciphertext C_{ID}.
- Decrypt: This algorithm is used for decrypting a ciphertext C_{ID}. The decryption process requires the secret key SK_{ID}.

Proxy re-encryption systems allow a semitrusted party (the proxy) to re-encrypt a ciphertext generated for a user A into a new ciphertext that another user B can decrypt. This process is achieved by using a re-encryption key generated by user A. A PRE system assures that the proxy and user B learn no information about the private key of A. Moreover, the proxy learns no information about both users' secret keys, as well as about the encrypted plaintext. Typically, a PRE scheme defines the following algorithms:

- KeyGen: This algorithm is used by a key generator and generates the public and the private key of each user.
- Encrypt: This algorithm can be used by any user. It encrypts a plaintext message msg using a public key P_{key}.
- RKGen: This algorithm is used by a user that owns a secret key S_{key1} (and the corresponding public key). The algorithm generates a re-encryption key $RK_{P_{key1}\rightarrow P_{key2}}$ where P_{key2} is the public key of another user.
- Reencrypt: This algorithm is used by a proxy. It accepts a re-encryption key $RK_{P_{key1}->P_{key2}}$ and transforms a ciphertext C into a new ciphertext that can be decrypted by a user that owns P_{key2}.
- Decrypt: This algorithm is used by a user, to decrypt ciphertext C.

Ateniese and Green[10] have implemented and identity-based PRE (IB-PRE) scheme. Figure 2 illustrates an instance of this scheme. In this example, there are three users identified by *User*1, *User*2, and *User*3 respectively. A PKG executes the IBE setup algorithm and generates the *MSK* and the *SP* (step 1). Then it creates the secret keys that correspond to identities *User*1 and *User*2 and distributes them accordingly (step 2). User *User*3, generates a ciphertext using the IBE encrypt algorithm and *User*1 as public key. Then he stores the ciphertext in a proxy (step 3). Supposedly user *User*2 want to access the stored item: *User*1 generates an suitable re-encryption key and transmits it to the proxy (step 4). The proxy uses this key and re-encrypts the ciphertext (step 5). User *User*2 is able to decrypt the new ciphertext using the IBE decrypt algorithm and his secret key.

3 | SYSTEM OVERVIEW

3.1 | Design

Our system extends the PSI architecture to support IB-PRE functions. In our system, every subscriber is identified by one or more identities, denoted by $ID_{subscriber}$, and owns the corresponding IBE secret key(s). The semantics of these identities are application specific. The IBE secret key that corresponds to each identity is generated, using out of band mechanisms, by a PKG with which the subscriber has business relationships. Each PKG is assumed to implement a process that guarantees that entities cannot lie about their identity. Multiple entities may have business relationships with the

same PKG. However, per-entity PKGs are also allowed. In addition to the PSI entities described in Section 2.1, our system considers a new type of publisher: the *trusted publisher*. A trusted publisher is a network entity trusted by the content owner to store (and advertise) access control policies and IBE re-encryption keys. Examples of trusted publishers are a web server owned by the content owner, a cloud provider with which the content owner has business relationships, etc. In our system, an access control policy is a list of *groups of subscribers*. The invocation of an access control policy is considered as the invocation of a function that takes an $ID_{subscriber}$ as input and outputs *true* if the subscriber belongs to a group specified by the access control policy and *false* otherwise. Access control policies are oblivious to the content items they protect.

Our proposed scheme is composed of the following processes: *content encryption and storage*, *content advertisement*, *subscriber authentication and authorization*, and *content forwarding*.

3.1.1 | Content encryption and storage

This process is executed by a content owner. During this process, a content owner generates an access control policy and assigns to it an identity, denoted by ID_{policy}. Moreover, a PKG with which the owner has business relationships creates the IBE secret key $SK_{ID_{policy}}$, which is transmitted to the owner using out of band, secured mechanisms. An access control policy can be used for protecting many content items. A content owner encrypts a content item, protected by ID_{policy} as follows. He generates a symmetric encryption key k and encrypts the item using a symmetric encryption algorithm and k; we refer to this ciphertext as $Enc_k(item)$. Each k can be used to encrypt only a single content item. Then, the owner uses IBE encrypt algorithm and ID_{policy} as a public key to encrypt k. We refer to the output of the latter encryption as $C_{policy}(k)$. Encrypting k with ID_{policy} may seem "strange" but, as we discuss in Section 4.1, it adds a significant security property: It protects our system against malicious RVs. In a nutshell, by using ID_{policy} as a public key, we make sure that even if the user authentication process fails, only authorized users will learn k.

As a next step and for each subscriber identity included in the used access control policy, the content owner generates the following IBE re-encryption key: $RK_{ID_{policy} \rightarrow ID_{subscriber}}$. Finally, he stores $Enc_k(item)$, $C_{policy}(k)$, a list of authorized subscriber identities, and a list of the corresponding re-encryption keys to a *trusted publisher*.

3.1.2 | Content advertisement

Upon receiving a protected content item, a trusted publisher advertises it to the appropriate RV, including in the advertisement message the list of the authorized subscribers, the list of the re-encryption keys, and the $C_{policy}(k)$ provided by the content owner. Any other publisher that may happen to have an item cached should advertise to the RV only the item's *RId*, ie, the list of authorized subscribers and the corresponding re-encryption keys, as well as $C_{policy}(k)$, are advertised only by trusted publishers. To implement this functionality, these lists can be digitally signed by the content owner.

At the end of this process, an RV maintains for each (protected) content item its *RId*, a list of authorized subscriber identities, the corresponding re-encryption keys, and $C_{policy}(k)$.

3.1.3 | Subscriber authentication and authorization

A subscriber may subscribe to a protected content item by sending an appropriate subscription message to the RV. This message should include $ID_{subscriber}$. The RV first checks whether $ID_{subscriber}$ is included in the list of authorized subscribers. If this is true, the RV initiates an authentication process. In particular, it generates a random challenge number r and sends to the subscriber $C_{ID_{subsciber}}(r)$. The subscriber uses his IBE private key and decrypts this ciphertext, then he responds to the RV with r. Finally, the RV selects the appropriate re-encryption key, it transforms $C_{policy}(k)$ into $C_{ID_{subscriber}}(k)$ and sends the latter ciphertext to the subscriber. The subscriber can use her IBE private key to decrypt $C_{ID_{subscriber}}(k)$ and retrieve k

3.1.4 | Content forwarding

After a successful subscriber authentication, the RV selects a publisher (trusted or not) and sends a notification message. The publisher simply forwards the (encrypted) content item to the subscriber. Any publisher along the path may cache it for later use.

Figure 3 gives an overview of our design.

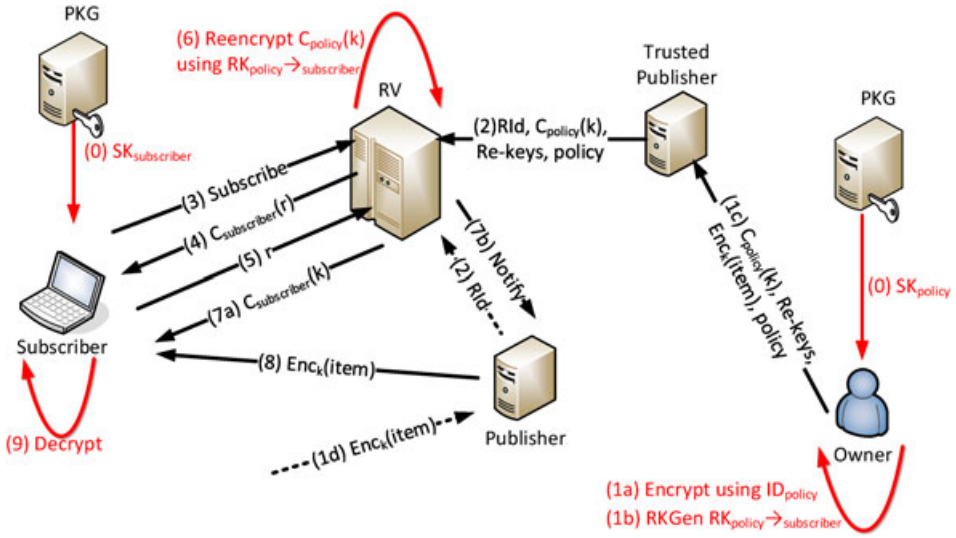

FIGURE 3 System overview. With red are the IBE-PRE specific operations

4 | EVALUATION

4.1 | Security evaluation

4.1.1 | Trust relationships and security assumptions

In our system, the following trust relationships should hold the following:

- **Trusted publishers and RVs are trusted to keep re-encryption keys secret:** The PRE scheme used in our system (presented in Green and Ateniese[10]) is not collusion resistant, ie, the algorithms used for generating secret keys allow an authorized subscriber to retrieve $SK_{ID_{policy}}$ if she combines her secret key and with re-encryption key. It should be noted, however, that this is possible only for policies for which a subscriber is authorized, ie, only if a re-encryption key of the form $RK_{ID_{policy} \to ID_{subscriber}}$ has been generated.
- **RVs are trusted to apply updates when necessary:** As we will discuss in Section 4.2.2, there are cases where $C_{policy}(k)$, the lists of authorized subscribers, and the lists of re-encryption keys have to be updated. It is critical that RVs not only to respect these updates but also to remove all withdrawn keys.
- **Authorized subscribers are trusted to keep their *SKs* and *k* secret:** Subscribers may have counter incentives to share their secret keys, eg, this may jeopardize their privacy. On the other hand, sharing k, ie, the symmetric key with which a file is encrypted, may pose no threat for them.

Moreover, our architecture makes the following security assumptions:

- **PSI control messages are transferred reliably:** Our scheme does not secure PSI control messages, instead it assumes that the security of these messages is handled by the underlay ICN layer.
- **There exists a reliable key revocation scheme for subscribers' keys:** Our scheme relies on an IBE key. It is a fact that key revocation in these schemes is nontrivial, as key revocation also means identity revocation. Our design assumes that a key revocation solution exists (eg, the serial number approach described in Fotiou and Polyzos[11]).
- **Subscribers can reliably determine content integrity, provenance, and authenticity:** Our solution considers only access control and content confidentiality. Content integrity, provenance, and authenticity verification are out of the scope of our system.
- **PKGs can reliably determine if a user is an owner of an identity and can securely distribute SKs:** In our system, it is assumed that users cannot generate a *SK* that corresponds to an identity or a policy that do not own. This can be performed by implementing various user identity verification techniques and/or by incorporating user identities inside policy identities.
- **PKGs are trusted to destroy secret keys immediately after they distribute them:** An inherent problem of all IBE-based systems is the so-called *key-escrow* problem, ie, the fact that users do not generate their secret keys

TABLE 1 Cryptographic operations computational overhead

Operation	Time, ms
Create $C_{policy}(k)$	6.2
Generate $RK_{ID_{policy} \rightarrow ID_{subscriber}}$	9.8
Re-encrypt $C_{policy}(k)$	1.0
Decrypt $C_{ID_{subscriber}}(k)$	5.3

by themselves, instead they are generated by a *PKG*. *PKGs* must be trusted therefore to not use this capability maliciously.

4.1.2 | Security properties

We now evaluate the security of our system under the following attack models: malicious RVs that do not respect the authorization process, malicious third parties that manipulate the messages between a subscriber and an RV, and malicious third parties that relay messages between a subscriber and an RV

Our system is secured against malicious RVs

Suppose that an RV does not authenticate subscribers. Assume also that a subscriber S requests a content item protected by policy P. If $RK_{P \rightarrow S}$ does not exist, then the RV cannot produce $C_S(k)$. On the contrary, if this re-encryption key exists, then it means that the subscriber is authorized. The *RV* generates $C_S(k)$ and sends it back to the subscriber. The subscriber can decrypt the latter ciphertext only if she knows the decryption key, ie, only if she is the real owner of S.

Message modifications can be detected

If a malicious user modifies $C_{subscriber}(r)$ (ie, message 4, which is sent from an RV to a subscriber), r (ie, message 5, which is sent from a subscriber to an RV) then the authentication protocol will result in an error; therefore, a subscriber should be able to detect the attack. Moreover, if an attacker modifies $C_{subscriber}(k)$ (ie, message 7a, which is sent from an RV to a subscriber), then the integrity verification check of the decrypted item (received with message 8) will fail.

Our system is secured against man-in-the-middle attacks

Suppose a malicious entity M that is able to intervene the communication between an authorized subscriber S and an RV. The goal of this attacker is to receive an item for which she is not authorized. The best thing that M can achieve is to convince the RV that she is S. In that case, she will learn $C_S(k)$, which is useless, since she cannot decrypt it.

4.2 | Performance evaluation

We have implemented the first IBE-PRE construction of Green and Ateniese[10] using the Charm cryptographic library.[12] In our implementation, protected items are encrypted using AES and a 128-bit key. As discussed in Section 3, symmetric encryption keys are encrypted using IBE. The scheme of Green and Ateniese is based on bilinear pairings, and it considers that the *msg* used as input in the *Encrypt* algorithm is an element of a pairing group G of prime order q. For this reason, the symmetric encryption keys of our scheme are generated using the following steps:

- Let G be the group used by the IBE-PRE construction
- select a random element $r \in G$
- using a hash function $H : G \rightarrow \{0,1\}^n$ calculate a symmetric encryption key $k = H(r)$.

The group G used by our implementation is composed of the elements of a symmetric elliptic curve with a 512-bit base field. Moreover, all identities used in our implementation are of equal size, ie, 256 bits[†]. The size of $C_{policy}(k)$ is 2048 bits, and the size of a re-encryption key $RK_{ID_{policy} \rightarrow ID_{subscriber}}$ is 3072 bits.

Table 1 presents the computation overhead of the cryptographic operations of our scheme. Measurements were obtained in an Ubuntu 16.04 machine, running in a single core of an Intel i7-4440 3.1 GHz processor with 16 GB of RAM, using Charm v0.43.

[†]In a real-world application, this can be achieved by hashing identities.

An RV should perform an encryption per session (to authenticate subscribers), as well as a re-encryption per authorized request. Content popularity has no effect on the total computational overhead, since re-encryption outputs cannot be re-used.

For each advertised item and in addition to the the PSI-related state, an RV maintains a list of identities, ie, the identities of the authorized subscribers, and a list of the corresponding re-encryption keys. Hence, for *each item*, an RV maintains $U \times (256 + 3072)$ bits of additional state, where U is the number of authorized subscribers. An optimization that can be considered for reducing this state is to optimize how records for items protected by the same policy are maintained. In particular, in each content advertisement, a trusted publisher can include the ID_{policy}; then for 2 or more items protected by the same ID_{policy}, a single list of identities and re-encryption keys can be maintained. It should be noted that this state is not related to the global number of subscribers, but only to the number of authorized subscribers.

Our scheme adds 3 additional messages to the PSI architecture; these are messages 4, 5, and 7a of Figure 3. Messages 4 and 5 are related to the subscriber authentication process, and message 7a is for delivering the content decryption key. Message 7a can be sent simultaneously with message 7b, ie, the publisher notification; therefore, it can be assumed that is adds negligible delay (since 7b is also sent in vanilla PSI). However, this is not the case for messages 4 and 5. An optimization that can be considered for omitting these messages in subsequent transactions, with the cost of some additional state in the RV, is the use of *hash chains*. In particular, when a subscriber requests access to a protected item for the first time, the RV generates a random number r and the hash chain $h^l(r)$ of length l; it stores $h^l(r)$ and sends to the subscriber l and $C_{ID_{subsciber}}(r)$. The subscriber calculates and sends back to the RV $h^{l-1}(r)$ and the RV checks if $h(h^{l-1}(r))$ is equal to the stored value of the chain. If this true, the subscriber is considered authenticated, and the stored value of the chain is replaced with the value that the subscriber sent. In all $m < l$ subsequent subscriptions, the subscriber includes $h^{l-1-m}(r)$ in the subscription message, and therefore, the RV is able to tell if the subscriber is authorized, without any further exchange of messages.

4.2.1 | Policy update

Every time a subscriber is joined to, or removed from an access control policy, the RV has to be updated with the new/removed subscriber identity and re-encryption key. Therefore, this introduces a communication overhead of 256 + 3072 bits, plus the size of the related control messages. This overhead is achieved by assuming the *lazy re-encryption model*,[13] ie, every time a subscriber is revoked, the protected item is not re-encrypted.

4.2.2 | Key update

We consider the following cases of key compromise: (1) the symmetric encryption key k is compromised, (2) the secret key $SK_{ID_{policy}}$ is compromised, and (3) the secret key of an authorized subscriber $SK_{ID_{subscriber}}$ is compromised. In the first case, the protected item has to be encrypted again using a new symmetric encryption key k'. Then, the RV has to be updated with $C_{ID_{policy}}(k')$ and the trusted publisher with $Enc_{k'}(item)$. Finally, the RV should remove all entries from non-trusted publishers, since they concern $Enc_k(item)$, which is now obsoleted. In the second case of key compromise, a new secret key $SK'_{ID_{policy}}$ has to be generated. Consequently all $C_{ID_{policy}}(k)$ ciphertexts and all $RK_{ID_{policy} \rightarrow ID_{subscriber}}$ re-encryption keys have to be updated. In the third case of key compromise, the subscriber will generate a new $SK_{ID_{subscriber}}$; therefore, a new $RK_{ID_{policy} \rightarrow ID_{subscriber}}$ has to be created; the RV has to be updated with this new key.

5 | RELATED WORK

Access control solutions for ICN can be divided into 2 categories: cryptography-based access control solutions and infrastructure-based access control solutions.

5.1 | Cryptography-based access control

Cryptography-based access control solutions rely on content encryption so as to protect content items from unauthorized access. A cryptographic scheme that is commonly used for this purpose is attribute-based encryption (ABE). Attribute-based encryption allows the specification of "attributes" that a user must hold to decipher a ciphertext.

This can be accomplished by linking attributes with private keys. Attribute-based encryption can also be used to define simple access control policies using logical operators (eg, it is possible to decrypt a ciphertext by the users who have the "cs" AND "student" OR "director" attributes). In ABE-based access control systems, content owners use attributes to group trusted subscribers (eg, students and professors), then they encrypt their content items using a symmetric encryption key, and encrypt this key using ABE. Many research efforts have studied cryptography-based access control in the context of ICN (eg, previous studies[14-20]).

All these solutions encrypt protected content items based on the access control policies that protect them, ie, access control policies determine the encryption keys. In our solution content items are encrypted using a random key, which is generated independently of the access control policy. This has a number of advantages:

- With our solution, an access control modification does not require content re-encryption.
- When a user key is revoked, content re-encryption is not required.
- Cryptographic operations are faster and ciphertexts are smaller.

An interesting category of encryption-based access control schemes are schemes that obfuscate content names such as only authorized subscribers can request them. An example of such a scheme is presented in Ghali et al.[21] Such solutions can be combined with our approach, and they can be used as an alternative to our authentication protocol.

5.2 | Infrastructure-based access control

In infrastructure-based access control systems, access control decisions are delegated to (semi-)trusted entities. A common example of this case is policy-based admission control.[22] Policy-based admission control is an RFC designed to allow network managers and service providers to delegate access control decisions for network resources to centralized trusted parties. These centralized parties are referred to as a policy decision points (PDPs), and they based their assessments on data collected by actors deployed in network resources. These actors, which are referred to as policy enforcement point (PEP), are responsible for collecting user related data, as well as for enforcing PDPs decisions. In this paper, we improved the infrastructure-based access control mechanism that we developed for the PSI architecture in Fotiou et al[6] (which shares similar goals with RFC 2753). In particular, the work in Fotiou et al[6] defines a mechanism that allows a trusted, centralized entity, known as the access control provider (ACP) to evaluate subscribers against an access control policy. Access control policies are created and stored in ACPs by the respective content owners. In order for a content owner to protect an item using a particular access control policy, he has to attach to that item a URI pointing to the ACP where the policy is stored. This URI is public information and can be used by any publisher to request from subscribers "proofs of authorization." These proofs are digital signatures generated by the ACP, pointed by the policy URI, over a publisher-generated token.

Compared to the solution presented in Fotiou et al,[6] this work has the following advantages:

- An ACP should be fully trusted, since it stores user credentials. In the solution presented in this work, access control is incorporated in the rendezvous process, and RVs have no access to user sensitive information.
- In Fotiou et al,[6] an RV should be trusted to respect the decision of an ACP. Indeed, a malicious RV may as well ignore the verdict of the ACP and notify a publisher to forward an item to the subscriber. In that case, unauthorized subscribers can have access to the protected items. In the solution presented in this work, only authorized subscribers can decrypt a protected item.
- The use of ACP introduces additional latency: In order for a subscriber to obtain a content item, he should communicate with both the ACP and the RV.

The work in Wood and Uzun[23] is also a form of infrastructure-based access control that uses PRE. The main differences of this work compared to ours are the following:

- The solution in Wood and Uzun[23] is built for CCN: an ICN architecture that does not have an entity similar to PSI's RV. For this reason, it requires either the content owner or a third trusted party to distribute re-encryption keys.
- In Wood and Uzun,[23] symmetric encryption keys are encrypted using IBE and the content item name as key. This has the drawback that the number of the re-encryption keys is proportional to the number of the (protected) content item identifiers, whereas in our solution, the number of re-encryption keys is proportional to the number of the access control policies.

6 | DISCUSSION AND CONCLUSIONS

In this paper, we proposed a rendezvous-based access control mechanism for ICN architectures. Our solution is built around the PSI architecture, nevertheless, it can be generalized to any rendezvous-based ICN architecture. The proposed system leverages identity-based PRE and achieves significant security properties with realistic overhead. By decoupling content-encryption process from access control enforcement, our system allows for efficient access control policy management, since the modification of a policy does not require content re-encryption. Moreover, by enforcing access control policies at the rendezvous points, unreliable publishers are allowed to store copies of the protected item. Finally, by using as a content encryption key a policy specific identifier, our solution is protected against misbehaving rendezvous points, ie, rendezvous points that satisfy the subscriptions of unauthorized subscribers.

Further work in this domain includes improvements that will make our solution suitable for as many ICN baseline scenarios as possible. For instance, our solution relies on IBE, which is computational intensive and cannot be used with constrained devices, such as the devices that will compose the IoT. For this scenario, gateway-based optimizations should be considered. Moreover, the use of IBE implies that a private key loss will result in the modification of an identity. This requires a carefully designed identity-management system for scenarios where identities are long lasting and semantic-rich, eg, as in a smart-city architecture.

ACKNOWLEDGEMENTS

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant no. (G-80-611-38). The authors, therefore, acknowledge with thanks DSR technical and financial support.

ORCID

Nikos Fotiou http://orcid.org/0000-0001-9100-1081

REFERENCES

1. Trossen D, Sarela M, Sollins K. Arguments for an information-centric internetworking architecture. *SIGCOMM Comput Commun Rev*. April 2010;40(2):26-33.
2. Content Centric Networking project. http://www.ccnx.org/. Accessed August 31, 2017.
3. Xylomenos G, Vasilakos X, Tsilopoulos C, Siris VA, Polyzos GC. Caching and mobility support in a publish-subscribe internet architecture. *IEEE Commun Magazine*. 2012;50(7):52-58.
4. FP7 PSIRP project. http://www.psirp.org/. Accessed August 31, 2017.
5. FP7 PURSUIT project. http://www.fp7-pursuit.eu/PursuitWeb/. Accessed August 31, 2017.
6. Fotiou N, Marias GF, Polyzos GC. Access control enforcement delegation for information-centric networking architectures. *SIGCOMM Comput Commun Rev*. 2012;42(4):497-502.
7. Fotiou N, Polyzos GC. Securing content sharing over ICN. In: Proceedings of the 3rd ACM Conference on Information-Centric Networking, ACM-ICN '16. ACM; 2016; New York, NY, USA:176-185.
8. Xylomenos G, Ververidis CN, Siris VA, et al. A survey of information-centric networking research. *IEEE Commun Surv Tutor*. 2014Second;16(2):1024-1049.
9. Trossen D, Parisis G. Designing and realizing an information-centric internet. *IEEE Commun*. 2012;50(7):60-67.
10. Green M, Ateniese G. Identity-based proxy re-encryption. *Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007:288-306.
11. Fotiou N, Polyzos GC. Enabling NAME-based security and trust. In: Damsgaard Jensen Christian, Marsh Stephen, Dimitrakos Theo, Murayama Yuko, eds. *Trust Management IX: 9th IFIP WG 11.11 International Conference, IFIPTM 2015, Hamburg, Germany, May 26-28, 2015, Proceedings*: Springer International Publishing; 2015:47-59.
12. Akinyele J, Garman C, Miers I, et al. Charm: a framework for rapidly prototyping cryptosystems. *J Cryptographic Eng*. 2013;3(2):111-128.
13. Garrison WC, Shull A, Myers S, Lee AJ. On the practicality of cryptographically enforcing dynamic access control policies in the cloud. In: 2016 IEEE Symposium on Security and Privacy (SP). San Jose, CA, USA; 2016:819-838.
14. Chen T, Lei K, Xu K. An encryption and probability based access control model for named data networking. In: 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC). Austin, Texas, USA; 2014:1-8.
15. Ion M, Zhang J, Schooler EM. Toward content-centric privacy in ICN: attribute-based encryption and routing. In: Proceedings of the 3rd ACM Sigcomm Workshop on Information-Centric Networking, ICN '13. ACM; 2013; New York, NY, USA:39-40.
16. Kuriharay J, Uzun E, Wood CA. An encryption-based access control framework for content-centric networking. In: 2015 IFIP Networking Conference (IFIP Networking). Toulouse, France; 2015:1-9.

17. Li B, Huang D, Wang Z, Zhu Y. Attribute-based access control for ICN naming scheme. *IEEE Trans Dependable Sec Comput*. 2016;PP(99):1-1.
18. Misra S, Tourani R, Natividad F, Mick T, Majd NE, Huang H. AccConF: An access control framework for leveraging in-network cached data in ICNs. *CoRR* 2016; abs/1603.03501. http://arxiv.org/abs/1603.03501.
19. da Silva RS, Zorzo SD. An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges. In: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). Las Vegas, NV, USA; 2015:128-133.
20. Wang J, Lang B. An efficient KP-ABE scheme for content protection in Information-Centric Networking. In: 2016 IEEE Symposium on Computers and Communication (ISCC). Messina, Italy; 2016:830-837.
21. Ghali C, Schlosberg MA, Tsudik G, Wood CA. Interest-based access control for content centric networks. In: Proceedings of the 2nd acm conference on information-centric networking, ACM-ICN '15. ACM; 2015; New York, NY, USA:147-156.
22. Yavatkar R, Pendarakis D, Guerin R. A framework for policy-based admission control. RFC 2753, RFC Editor; January 2000.
23. Wood CA, Uzun E. Flexible end-to-end content security in CCN. In: Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th. Las Vegas, NV, USA: IEEE; 2014:858-865.

Nikos Fotiou is a researcher at the Mobile Multimedia Laboratory, Department of Informatics, Athens University of Economics and Business (AUEB). He has received his diploma in Information and Communication Systems Engineering from the University of the Aegean in Samos, Greece (2005), his MSc in Internetworking from the Royal Institute of Technology (KTH) in Stockholm, Sweden (2007), and his PhD in Computer Science from AUEB. His current research interests include publish/subscribe architectures and information-oriented security mechanisms.

Bander A. Alzahrani is an assistance professor in Information Systems Department at King Abdulaziz University. He completed his MSc in Computer Security (2010) and his PhD in Computer Science (2015) from Essex University, Colchester, UK, in 2009. His research interests include network security, information centric networks, bloom filter data structure and its applications, secure content routing, and IoT.

How to cite this article: Fotiou N, Alzahrani BA. Rendezvous-based access control for information-centric architectures. *Int J Network Mgmt*. 2018;28:e2007. https://doi.org/10.1002/nem.2007