Access Control Enforcement Delegation for Information-Centric Networking Architectures

Nikos Fotiou, Giannis F. Marias, George C. Polyzos
Mobile Multimedia Laboratory, Department of Informatics
Athens University of Economics and Business, Greece
{fotiou, marias, polyzos}@aueb.gr

ABSTRACT

Information is the building block of *Information Centric Networks* (ICNs). Access control policies limit information dissemination to authorized entities only. Defining access control policies in an ICN is a non-trivial task as an information item may exist in multiple copies dispersed in various network locations, including caches and content replication servers. In this paper we propose an access control enforcement delegation scheme which enables the purveyor of an information item to evaluate a request against an access control policy, without having access to the requestor credentials nor to the actual definition of the policy. Such an approach has multiple merits: it enables the interoperability of various stakeholders, it protects user identity and it can set the basis for a privacy preserving mechanism. An implementation of our scheme supports its feasibility.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

Keywords

Access control policies, Privacy preservation, Stakeholder interoperability

General Terms

Security, Design, Experimentation

1. INTRODUCTION

Information-Centric Networking (ICN) is an emerging paradigm envisaged by a growing body of researchers. ICN architectures leverage the role of information as the building block of the Internet architecture, in contrast to the current end-host oriented paradigm. ICN architectures have better support for multicast, mobility, and security.

In ICN architectures efficient information dissemination is expected to be supported by dispersing an information item in many network locations using, for example, caches, CDN-like networks and bittorrent-like systems. Nevertheless this information dispersion raises severe security concerns, as it will make difficult the enforcement of access control policies. It is unrealistic to expect that each information item transmission will be accompanied by an access control policy that each purveyor should implement; not only that requires the existence of a complex access control mechanism in each purveyor, but also implies that everybody should have access to the user management system of the information owner as well as that everybody will have an insight in the attributes of a user that requests an information item protected by access control. Access control policies computations should not be a task performed by any network entity that potentially hosts an information item, but instead hosting entities should delegate access control decisions to Access Control Providers (ACPs) that are considered reliable by the information owner; hosting entities then only have to respect the decision of these ACPs and enforce them.

In this paper we propose an access control enforcement delegation system for ICN architectures. Our system operates simply by exploiting the functions of the underlay architecture and provides user credential protection, privacy preservation, and facilitates stakeholders interoperability. Furthermore, in our system, hosting entities can evaluate a request for an item against an access control policy, without having access to the policy itself.

The basic principle of the proposed system is that an information owner attaches to every information item a pointer to a function that implements the access control policy that protects that item, rather than the policy itself. Any purveyor can challenge an item requestor to invoke that function, and based on the function's output, the purveyor can decide whether or not the requestor is eligible to access the protected item.

The remainder of this paper is organized as follows. Section 2 presents the underlay ICN architecture. Section 3 gives a high level view of our scheme, which is then detailed in Section 4. We evaluate the security properties and the communication overhead of our scheme in Section 5, we compare it with related work in this area in Section 6 and we present our conclusions in section 7

2. UNDERLAY ARCHITECTURE

Our reference architecture is the PURSUIT[1] architecture, but we believe that our scheme can be adapted to any

[1]http://www.fp7-pursuit.eu/

ICN'12, August 17, 2012, Helsinki, Finland.

rendezvous-based ICN architecture. The rendezvous is essentially a lookup service that maps information *requests* to information *advertisements*. This map results in information being forwarded from an information purveyor (from hereafter will be referred to as the *Publisher*) to the interested parties (from hereafter they will be referred to as the *Subscribers*). The underlay architecture abides by the publish-subscribe paradigm [5]

Publishers advertise an information item they possesses by *publishing information* about the item's identity to a *Scope*. The Scope can be regarded as a topological hint and it can be for example the URL of a web server, a network path or a path in a social graph. A Scope may as well implement a specific dissemination strategy (e.g., every information item advertised in a social network can be accessed by the clients of this social network). Every publisher can also *publish a new scope*, this way hierarchies of scopes can be created. Every scope is managed by a *rendezvous node*.

A Subscriber requests access to an information item by sending a *subscription* message to the rendezvous node that manages the item's scope. This specific rendezvous node is referred to as the *rendezvous point* of the item. In the rendezvous point, a subscription message is matched with an advertised item. Upon such a match the rendezvous point will *notify* the publisher to *publish information data* to the subscriber. In case there is not any information advertisement that matches the subscription, the subscription will be kept for a period of time in case a matching advertisement appears in the future. Therefore a subscriber can subscribe to information items that have not yet been advertised.

In order to better understand these notions consider the following example. Suppose a hospital that maintains a rendezvous node, named RN-H, which manages a scope called "Hospital". A doctor, would like to create a new scope under which he will advertise medical prescriptions for "Patient X" (therefore he chooses to name this new scope "Patient X prescriptions"). He achieves that by sending a `publish_scope("Hospital/Patient X prescriptions",...)` message to RN-H. This message may contain additional parameters such as, an access control policy for that scope that specifies the doctors who can advertise prescriptions as well as the authorized subscribers. In order for a doctor to advertise a prescription identified by "PRE-0110" and stored in his office server, he has to send a `advertise("Hospital/Patient X prescriptions","PRE-0110",...)` message to RN-H from his office server. This message may also contain additional arguments–such as doctor's credentials. A patient wishing to access "PRE-0110" has simply to send a `subscribe_info("Hospital/Patient X prescriptions","PRE-0110",...)` message from his pc to RN-H, providing his credentials as an additional argument. This subscription message will result in RN-H `notifying` doctor's office server to send the data of "PRE-0110" to patient's pc.

It should be noted here that this example gives only a high level overview of the PURSUIT architecture hiding many of its internal functions. Interested readers are referred to the PURSUIT project deliverables[2]. Moreover there are other ICN architectures that abide by the same principles; these architectures can potentially be used as underlays for our scheme.

[2]http://www.fp7-pursuit.eu/PursuitWeb/?page_id=158

3. SYSTEM OVERVIEW

3.1 System entities

Our system is composed by the following parties: the Information Owner (Owner), many Information Consumers (Consumers), the Relaying Party (RP) and the Access Control Provider (ACP). The Owner is the principal of an information item, he has full control over it and specifies the access control policy that governs its dissemination. Consumers on the other hand are entities interested in accessing a specific piece of information. An information item can be stored in multiple RPs–such as content replication servers, caches and web servers. An RP is responsible for the efficient distribution of information items from their storage point to authorized Consumers. The access control policy of an information item is stored in the ACP. An ACP provides the means to Owners for creating access control policies and is responsible for evaluating Consumers against these policies. An ACP can be for example an LDAP or a social network. Access control policies define the attributes that a Consumer should have in order to access an information item, without specifying the identity of that item. They can be regarded as a function that accepts as input a Consumer ID and a set of attributes and outputs *True* if the Consumer satisfies the input attributes or *False* on the contrary.

3.2 Security model

In the general case it is assumed that RPs and ACPs belong to different administrative domains. It is also assumed that on any domain, the information Owner is capable to control his owned items and to define the access control policies that apply to them. Of course an Owner can implement his own RP or ACP, over which he will have full control. It is assumed that every Consumer can authenticate herself to the ACP. Both RPs and ACPs are considered to be honest but curious, i.e., they operate as expected but try to obtain as much information regarding Consumers as possible. There should not be any trust relationship between an RP and an ACP, they should only agree on a (trivial) communication protocol. On the contrary an Owner should trust an ACP to properly apply an access control policy and an RP to operate according to the ACP decision. An information item can be encrypted. Encryption can act as a counter-incentive for an RP to misbehave, as unauthorized Consumers will not be able to read an encrypted item, therefore the RP will just waste resources for sending it. Finally it is assumed that all entities have abundant resources and the underlay architecture assures data integrity, confidentiality and provenance (we will further elaborate this assumption in Section 4.1).

3.3 Design Goal

Our main goal is to create a system in which a Consumer's credentials are protected and privacy is preserved (e.g., as opposed to current approaches where user names and passwords remain hidden but other user's attributes–such as age, sex–become known). The RP should be oblivious about the access control policy that protects an information item, and therefore, about any information associated with the Consumer. Similarly, details about the information item for which a Consumer requests access should remain hidden

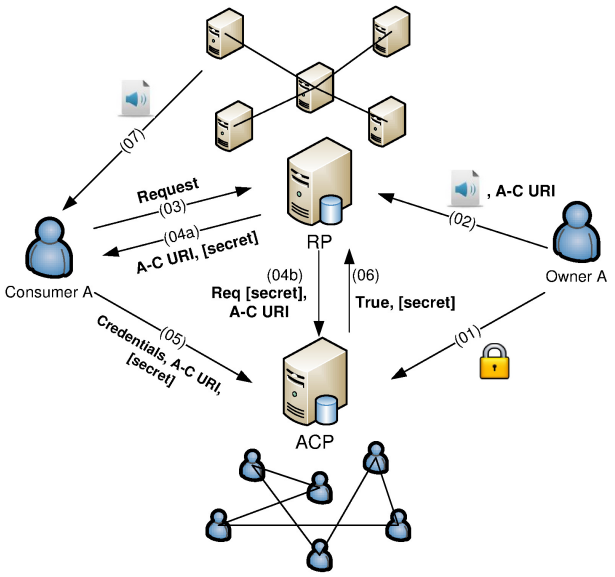

Figure 1: System design overview

from the ACP. Moreover our system design aims at separating the functions of each stakeholder: each stakeholder should have a clear, distinct role in the system.

Another design goal is to eliminate as much as possible the Consumer intervention in the communication between the ACP and the RP. ACP and RP should communicate directly without relying on the Consumer to properly relay their messages. Consumer intervention introduces significant security risks that make the implementation of such systems complicated (as we will see in the related work section many similar systems suffer from security attacks due to the poor implementation of the Consumer relay).

3.4 System design

Figure 1 illustrates our system entities and their interactions by an example. This example shows how our system would operate in the current end-host oriented Internet. In the next section we will see how ICN simplifies this design. In the illustrated example a user (Onwer A) is the Owner of an audio file which he wants to share with his friends in a social network (ACP). For efficiency reasons the user wants to use a content distribution network (RP). Initially Owner A creates a new access control policy which outputs *True* for all his friends in the ACP and stores this policy in the ACP (message 01). The ACP creates a unique URI (A-C URI) for this access control policy (which becomes known to the content owner). As a next step Owner A sends the audio file to the RP, indicating at the same time that this file is protected by the A-C URI access control policy (message 02). A friend of Owner A, named Consumer A, requests access to this file (message 03). The RP responds with the A-C URI and a session secret (message 04a). At the same time the RP notifies the ACP that it expects somebody that holds the secret to invoke the A-C URI access control policy (04b). Consumer A is a client of the ACP, she authenticates herself and she invokes A-C URI providing the same time the secret (message 05). The ACP notifies the RP that a user that knows the secret, authenticated herself correctly (06) and RP sends the audio file to Consumer A through the closest server (message 07).

It can be seen that this system satisfies our design goals: the RP is completely oblivious about the content of the access control policy and the identity of Consumer A. It has only access to the access control URI and to the end-point address of Consumer A. Similarly the ACP learns no information about the file in which Consumer A is interested in. The RP knows nothing about how the ACP implements access control and the ACP does not know how the RP makes its forwarding decisions;the RP's and the ACP's functionalities are completely differentiated and independent from each other. Finally the RP and the ACP have direct communication without the intervention of Consumer A.

It should be noted here that since there is not an 1-to-1 relationship between an access control policy and an item, an access control policy is reusable, i.e., the same access control policy can be used for controlling the dissemination of multiple information items, stored in different RPs. Moreover an access control policy is not specific to an RP, so in our example if the A-C URI was embedded in the audio file any RP that would receive that file–no matter from whom–would be able to protect this file using the exact same access control policy. Therefore any cache, or bittorrent tracker, or mirror site would have been able to follow the same process.

4. IMPLEMENTATION

An implementation of our scheme has been developed by extending a prototype of the PURSUIT architecture, codenamed Blackadder [9]. This extended prototype[3], supports all the primitives described in Section 2.

In an ICN architecture RPs and ACPs can be regarded as rendezvous nodes. Owners and Consumers interact with these nodes by sending the appropriate architecture-specific messages. Initially, an Owner creates an access control policy, denoted as F_s to describe the attributes that a Consumer should have in order to access an item protected by F_s. Let SId_{ACP} be a scope managed by an ACP. The Owner creates(publishes) a new scope, named SId_{fs}, under SId_{ACP}, including F_s in the publication message, i.e., he sends to the ACP a: `publish_scope(`SId_{ACP}/SId_{fs}`,`F_s`)` message[4]. When the ACP receives this message, it creates a new scope ("SId_{ACP}/SId_{fs}") in which everybody can advertise items, but only users(Consumers) that abide by F_s can subscribe to items that are advertised under that scope. An Owner can create numerous policies using the same process. As a next step, the Owner incorporates a meta-data field in the information items that he wants to be protected by F_s. This field denotes that *this item is protected by "*SId_{ACP}/SId_{fs}*"*, i.e., the newly created scope. Therefore, the Owner incorporates a "pointer" to F_s rather than F_s itself. If any Owner wants to protect an information item with the same access control policy (i.e., F_s) he has simply to attach to that item the same pointer; the item will be protected even if the Owner does not know the actual content of F_s. As a final step the Owner has to send(forward) this item to an RP; after

[3]The latest open-source version of Blackadder can be found at https://github.com/fp7-pursuit/blackadder whereas our implementation can be found at http://mm.aueb.gr/research/icn-access.zip

[4]Some of the implementation specific parameters of the message are omitted for clarity reasons.

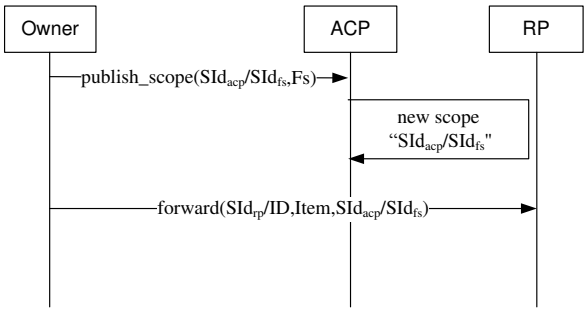

Figure 2: Policy creation and item publication

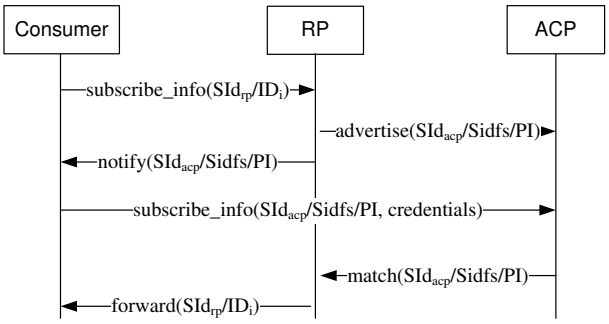

Figure 3: Subcription to an item

this step the item will be available under the RP scope, i.e., SID_{rp}. Figure 2 illustrates the above process.

A Consumer interested in accessing ID_i should send a `subscribe_info`(SID_{rp}/ID_i) message to the RP. The RP knows that ID_i is protected by "SId_{ACP}/SId_{fs}". It is also known that, by design, a Consumer that is able to subscribe to an item published under "SId_{ACP}/SId_{fs}" is also eligible to subscribe to any item protected by "SId_{ACP}/SId_{fs}". That happens because–as previously mentioned–subscriptions to "SId_{ACP}/SId_{fs}" are limited to the users that satisfy F_s, which is the access control policy that protects ID_i. Therefore, if a Consumer proves to the RP that she is allowed to subscribe for an information item published in "SId_{ACP}/SId_{fs}" then this Consumer gains access to ID_i. Based on this, the RP upon receiving the subscription message generates a pseudo-item, identified by PI, `advertises` it at the ACP under the scope "SId_{ACP}/SId_{fs}" and `notifies` the interested Consumer that, in order for her subscription to be completed she has firstly to successfully subscribe to "$SId_{ACP}/SId_{fs}/PI$". Since everybody is allowed to advertise items under "SId_{ACP}/SId_{fs}" the advertisement of PI by RP is achieved simply by sending an `advertise`($SId_{ACP}/SId_{fs}/PI$) message to the ACP. Finally the RP notifies the Consumer, using the standard notification procedure of the underlay architecture. Upon receiving this notification, the Consumer, subscribes to "$SId_{ACP}/SId_{fs}/PI$" by sending a `subscribe_info` message to the ACP, which includes her credentials. The ACP evaluates the Consumer's credentials against F_s, since all subscriptions to items stored under "SId_{ACP}/SId_{fs}" should abide by F_s. If the credentials are in compliance with F_s, the ACP will send a notification message to the RP informing it that a Consumer successfully subscribed to PI. However, since PI is only known to the Consumer, the RP will understand that the Consumer can access ID_i. The subscription process is illustrated in Figure 3.

In a nutshell, for each access control policy a scope is created in an ACP. Everybody can advertise items in that scope but only users that match the access control policy can subscribe to items advertised in that scope. Everytime a Consumer requests an access control protected object from an RP, this RP publishes a pseudo-item in the scope that corresponds to that access control policy and challenges Consumer to subscribe to that item. If the RP receives a notification that somebody subscribed to that item then it understands that the Consumer matches the access control policy. All the publication, subscription and notification operations are functions provided by the underlay architecture.

4.1 Secure communication

On purpose, we have not described so far the form of the scope and information item identifiers. The reason for this is that the naming format is left to the underlay architecture; the underlay is required to provide self-certifying names that guarantee the integrity, the relevance, the provenance and the confidentiality of the transmitted messages–as discussed in [14, 6]. Due to the cryptographic primitives associated with the names, content is secured even if it is transmitted over an unsecured channel. Moreover we assume that the underlay architecture provides mechanisms that prevent replay attacks on the notification messages. Although no concrete solution has been proposed on this problem yet, we believe that it can be easily solved (e.g., by adding an encrypted nonce in the publication/subscription messages which is repeated in the respective notifications)

5. EVALUATION

5.1 Security Evaluation

The proposed system has the following security-related properties:

1. *Consumer credentials are protected:* The only entity that has access to Consumer credentials is her credentials' provider, i.e., the ACP. Moreover due to the security primitives of the underlay architecture–as described in Section 4.1–an attacker cannot eavesdrop Consumer credentials, neither can he pretend to be an ACP.

2. *Consumer privacy is preserved:* The only information an RP learns about a Consumer is an end-point address as well as that she is a client of an ACP. Similarly what an ACP learns about a consumer is that she is interested in an item which an RP "knows". An attacker eavesdropping all communication channels can learn that an "endpoint" interacts with a particular RP and a particular ACP. Only an ACP colluding with an RP can obtain full information about a Consumer.

We also examine the following attack scenarios:
Man in the middle:
In this attack an attacker pretends to be an RP, trying to hijack a session between a Consumer and a legitimate RP, in order to obtain the information item that the Consumer requests. In order for this attack to be successful the attacker should persuade the Consumer that he is the legitimate RP.

However the security primitives described in Section 4.1 assure data provenance, therefore the Consumer will understand that the messages received from the attacker did not originate from the legitimate RP.

Malicious Consumer colluding with fake ACP:
In this attack a fake ACP tries to persuade an RP that a Consumer successfully subscribed to the pseudo-item by sending a fake notification message. This attack cannot be achieved due to the provenance assurance: the RP will understand that the notification did not originate from the real ACP. An attacker may circumvent this security mechanism by "replaying" a captured legitimate notification. Providing that the RP uses a different id for each pseudo-item he creates, the replay attack will also be unsuccessful, as the replayed notification will concern another pseudo-item id.

Malicious Consumer colluding with a valid Consumer:
In this attack a malicious Consumer asks a valid Consumer to subscribe on her behalf to the pseudo-item. RP will receive the notification and it will publish the protected item to the malicious consumer. Therefore this is a successful attack. Nevertheless this attack is equivalent of having a valid consumer "giving" the protected item to an unauthorized Consumer using out of band mechanisms. We leave this attack as an open issue, that currently should be handled by application-layer solutions (e.g., encrypt items using attribute-based encryption, therefore the valid consumer should reveal his private keys to the malicious user)

5.2 Communication Overhead

Our scheme introduces a small communication overhead when it comes to the information advertisement and subscription. Every time an information item is advertised to an RP the corresponding access control policy has to be communicated to an ACP; this introduces an extra message which, however, can be omitted when access control policies are re-used. Moreover, the advertisement message that is sent from an Owner to the RP contains a pointer to the corresponding access control policy, therefore, its size is increased by the size of that pointer. The size of the pointer is application specific and in general it is expected to be as big as an object identifier.

For the information subscription operation four new messages are introduced: the message for the pseudo-item advertisement, the notification sent from the RP to the Consumer, the subscription to the pseudo-item and the notification sent from the ACP to the RP. Various optimizations can be considered in order to decrease the communication overhead introduced in the subscription operation. E.g., the Consumer can decide what the identifier of the pseudo-item will be, therefore, the notification sent from the RP to the Consumer can be omitted and the subscriber can send simultaneously the two subscription messages (one for the information item and one for the pseudo-item). Nevertheless, in all cases some extra messages are unavoidable.

6. RELATED WORK

To our knowledge this is the first research attempt to address the problem of access control enforcement delegation in the context of ICN. Access control issues have been surmounted so far using cryptographic solutions in information naming or at the packet level [4]. Nevertheless these solutions simply transfer the problem of access control to the endpoints or to the rendezvous point [3], whereas our system leverages the role of in-network mechanisms.

Our work is inspired by single sign-on (SSO) systems–such as OpenID [13] and Shibboleth [11]. Nevertheless our system differs from SSO in a significant way: SSO systems are based on the so-called *proof-by-possession* primitive, i.e., users(Consumers) authenticate themselves to an RP by providing a token issued by the identity provider. This token can be in the form of a web cookie, a HTTP field or a security "ticket". This token however may constitute a security [16] or privacy [15] threat. The secure implementation of this token is complicated and even popular SSO providers–including Facebook and Google–have been proved prone to severe security attacks for this reason [16]. In our system the user does not intervene in the communication between the ACP and the RP, eliminating this way those attacks.

Access control using anonymous credentials–such as in [1, 2]–as well as schemes for delegating user private resources–such as OAuth [8]–are also close to our work. In the former systems the RP is responsible for evaluating an access control policy and is granted access to the user(Consumer) attributes that are required in order to achieve this task, whereas in the OAuth case the RP and the ACP are the same entity. In our solution the RP neither gets access to any user information nor does it evaluate any access control policy; the only entity that has access to both user attributes and access control policies is the ACP. This approach has many merits: it safeguards user credentials, it preserves user privacy and it releases RP from the burden of evaluating access control policies. Moreover, in our approach the RP and the ACP are two distinct entities with separated roles.

Privacy preserving access control schemes–e.g., [7, 12]–and decentralized access control mechanisms for cloud services and distributed systems–e.g., [17, 10]–are orthogonal to our work. Those schemes provide cryptographic primitives that enable outsourcing data storage as well structures that enable the co-operation of various access control mechanisms. Those tools can be used by RPs for securely storing data and by ACPs for creating chains of trust. In any case these mechanisms are transparent to our system, which operates on a higher layer.

7. CONCLUSIONS

In this paper we designed and implemented an access control enforcement delegation scheme for ICN architectures. The proposed scheme tackles the thorny problem of access control in ICN architectures in an efficient and radically new way. Our system protects user credentials and preserves user privacy. Moreover, it clearly separates the roles and the functions of each stakeholder. Access control decisions are made in our system by in-network mechanisms and they are not left as afterthoughts to be handled by endpoints.

An implementation of our scheme shows its feasibility and exhibits the strengths of an information-oriented architecture: the basic functions that are used for information organization and dissemination can be combined in a straightforward way in order to achieve the design goals of our system.

By embedding a pointer to an access control policy–and not the access control policy itself–to an information item, any intermediate node that handles this item can protect it using this policy, without having access to its definition. Therefore, caches, replication servers, mirrors, and many

other stakeholders can protect information items without having to implement any access control computation mechanism and without having access to sensitive information of the users.

Future work in this area includes support for ACP federations and implementation of our scheme for other ICN architectures. Moreover we pursue to evaluate the scalability of ACPs through simulation and large scale deployment.

8. ACKNOWLEDGMENTS

The work reported in this paper was supported by the FP7 ICT project PURSUIT, under contract ICT-2010-257217

9. REFERENCES

[1] C.A. Ardagna et al. Enabling privacy-preserving credential-based access control with XACML and SAML. In *Proceedings of the 2010 IEEE CIT*, pages 1090–1095, Washington, DC, USA, 2010.

[2] J. Camenisch, S. Mödersheim, G. Neven, F.-S. Preiss, and D. Sommer. A card requirements language enabling privacy-preserving access control. In *Proceedings of the 15th ACM symposium on Access control models and technologies*, SACMAT '10, pages 119–128, New York, NY, USA, 2010.

[3] N. Fotiou, G. F. Marias, and G. C. Polyzos. Towards a secure rendezvous network for future publish/subscribe architectures. In A. e. a. Berre, editor, *Future Internet - FIS 2010*, volume 6369 of *Lecture Notes in Computer Science*, pages 49–56. Springer Berlin / Heidelberg, 2010.

[4] N. Fotiou, G. F. Marias, and G. C. Polyzos. Publish-Subscribe internetworking security aspects. In L. Salgarelli et al., editor, *Trustworthy Internet*, pages 3–15. Springer Milan, 2011.

[5] N. Fotiou, D. Trossen, and G. C. Polyzos. Illustrating a publish-subscribe internet architecture. *Telecommunication Systems, Springer*, pages 1–13, 2010.

[6] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker. Naming in content-oriented architectures. In *Proceedings of the ACM SIGCOMM ICN workshop*, pages 1–6, New York, NY, USA, 2011.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. ACM, 2006.

[8] E. Hammer-Lahav. The oauth 1.0 protocol, 2012. `http://art.tools.ietf.org/html/rfc5849`.

[9] Kjällman, J., ed. PURSUIT deliverable 3.2, first lifecycle prototype implementation (d3.2), September 2011. `http://www.fp7-pursuit.eu/`.

[10] S. Miltchev, J. M. Smith, V. Prevelakis, A. Keromytis, and S. Ioannidis. Decentralized access control in distributed file systems. *ACM Comput. Surv.*, 40(3):10:1–10:30, Aug. 2008.

[11] R. Morgan, S. Cantor, S. Carmody, W. Hoehn, and K. Klingenstein. Federated security: The Shibboleth approach. *Educause Quarterly*, 27(4):6, 2004.

[12] Q. Ni, E. Bertino, J. Lobo, and S. Calo. Privacy-aware role-based access control. *Security Privacy, IEEE*, 7(4):35 –43, july-aug. 2009.

[13] D. Recordon and D. Reed. Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital Identity Management*, DIM '06, pages 11–16, New York, NY, USA, 2006.

[14] D. Smetters and V. Jacobson. Securing Network Content. Technical report, PARC, 2009.

[15] M. Uruena and C. Busquiel. Analysis of a privacy vulnerability in the openid authentication protocol. *IEEE Multimedia Communications, Services and Security (MCSS2010)*, 2010.

[16] R. Wang, S. Chen, and X. Wang. Signing me onto your accounts through Facebook and Google: a traffic-guided security study of commercially deployed single-sign-on web services. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2012. to appear.

[17] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, 2010.