Dear Author,

Please, note that changes made to the HTML content will be added to the article before publication, but are not reflected in this PDF.

Note also that this file should not be used for submitting corrections.

Computer Standards & Interfaces xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Attribute-based authorization for structured Peer-to-Peer (P2P) networks

Diego Suárez Touceda [a,*], José M. Sierra Cámara [b], Sherali Zeadally [c], Miguel Soriano [d,e]

[a] Evalues — IT Security Evaluation, Parque Leganés Tecnológico, Avda. Gregorio Peces Barba 1, 28918 Leganés, Madrid, Spain
[b] Computer Science Department, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911 Leganés, Madrid, Spain
[c] College of Communication and Information, University of Kentucky, Lexington, KY 40506, USA
[d] Department of Telematics Engineering, Universitat Politècnica de Catalunya (UPC), 08034 Barcelona, Spain
[e] Centre Tecnolgic de Telecomunicacions de Catalunya (CTTC), 08860 Castelldefels, Barcelona, Spain

ARTICLE INFO

Article history:
Received 16 July 2014
Received in revised form 2 April 2015
Accepted 25 April 2015
Available online xxxx

Keywords:
Authorization
P2P security
Attribute Certificates

ABSTRACT

We present the deficiencies, lack of flexibility and inefficiency in the assignment of privileges, of traditional identity-based authorization models in structured Peer-to-Peer (P2P) networks where user's Public Key Certificates (PKCs) represent two roles, user's authentication and user's authorization, and the access to the network resources is controlled by Access Control Lists (ACLs). With these deficiencies in mind, we propose a complete new framework for authorization in structured P2P networks based on Attribute Certificates (ACs) which links the privileges of a user within the system with its identity (represented by a Public Key Certificate (PKC)). We also present a distributed certificate revocation system that can be established within the structured P2P network and does not need the intervention of any external server. We argue that the proposed separation between authentication and authorization yields a more flexible and secure authorization scheme for structured P2P networks while improving the efficiency of the assignment of privileges in comparison to the existing identity-based approaches.

1. Introduction

Early Peer-to-Peer (P2P) systems were intended for file-sharing purposes which determine the guidelines adopted by access control solutions deployed in such networks. Their open-nature and the free availability of shared resources motivated researchers to focus more on restricting the number of malicious nodes in the network than implementing an authentication and authorization mechanism per se. However, in recent years several P2P applications (audio and video conferencing, multi-party games, content distribution, etc.) have emerged which require a more fine-grained access control.

Several alternative schemes have been proposed in the literature to try to solve the access control problem for this type of applications for decentralized architectures and the lack of centralized online infrastructures. Structured Peer-to-Peer (P2P) systems use: Internet Protocol (IP) based [1] access control, web of trust [2], shared secret [3], decentralized certification [4] or offline certification [3,5]. Regardless of the specific model used, one property is common to all of them: Public Key Certificates (PKCs) (either self-signed or by a Trusted Third Party (TTP)) are used for the authentication of users. However, these PKCs represent two roles: user's authentication (who the user is) and user's authorization (privileges of the user in the network: usernames allowing the user to join the network and to have a location in the ID space to store its resources, nodeIDs establishing its location in the network and the resources it is responsible for, storage quota limiting the amount of data it can store in the network, etc.). Also, PKCs are complemented by using Access Control Lists (ACLs) to control the access to the network resources.

However, the fact that PKCs are used for both authentication and authorization of users is not a good idea [6]. Including the identity and the privileges of a user (username, nodeID, services contracted, etc.) into the same certificate requires that both the identity and any privileges should have the same lifetime and should be issued by the same authority. In addition, every time a new privilege is added, removed or changed the certificate should be revoked and a new one should be created. This authorization approach is inefficient and does not consider scenarios where the identity of the users and their privileges are provided by different entities.

In the same way, ACLs perform well in operating systems or client–server architectures but not in structured P2P networks. In order to be usable, ACL's content has to be made public (to let the reader verify that the resource it is accessing has been created by an authorized user) revealing all the users' privileges over a resource and, therefore, affecting the privacy of users. Moreover, the fact that all the resource's replicas should be contacted in order to modify the resource's ACL for granting a new user privileges over it, makes this approach inefficient.

Finally, despite the fact that most P2P applications use short-lived PKCs, the different nature of the privileges that can be assigned in a

* Corresponding author. Tel.: +34 91 624 40 34.
E-mail addresses: diego.suarez@uc3m.es (D.S. Touceda), sierra@inf.uc3m.es (J.M.S. Cámara), szeadally@uky.edu (S. Zeadally), soriano@entel.upc.edu (M. Soriano).

http://dx.doi.org/10.1016/j.csi.2015.04.007
0920-5489/

P2P system and the existence of applications with special security requirements, would make revocation of privileges desirable in some cases. Unfortunately, existing alternatives based on centralized servers (such as the Certificate Revocation List (CRL) servers [7]) or trusted intermediary authorities (such as the Online Certificate Status Protocol (OCSP) responders [8,9]), that should be contacted each time a certificate has to be checked, are not an option for P2P systems.

One specific example of protocol using the before commented techniques (PKC + ACL), and therefore suffering from all the commented drawbacks, is the IETF P2P standard REsource Location and Discovery (RELOAD) protocol [3] and its usage for shared resources [10]. RELOAD is the only existing standard for P2P networks and, although it was initially designed with P2PSIP in mind, it can be utilized by other applications with similar requirements, such as Scribe or P2PCast [11].

Considering the above limitations of the existing authorization approaches (including the only existing standard RELOAD) for structured P2P networks, in this paper we present an new authentication framework for structured P2P networks based on the recently published Internet Attribute Certificate Profile for Authorization [6] that we adapt and extend to make it suitable for structured P2P networks.

The main contributions of this paper are:

1. We present an analysis of the deficiencies of traditional identity-based authorization models in structured P2P networks showing their lack of flexibility, efficiency and privacy in the assignment of privileges.
2. We propose and present a general framework for authorization in structured P2P networks that not only solves the identified deficiencies but homogenizes the access control under a unique authorization schema. Our framework is intended for structured P2P networks where user resources are distributed over the network, but it could be used with any P2P system that uses PKC as source of authentication.
3. We present of a distributed revocation system that can be established within the structured P2P network and does not require the intervention of any external server or trusted intermediate authority.
4. We evaluate (both theoretically and with simulations) of our framework by applying it to the RELOAD protocol and by comparing it against the RELOAD's original identity-based authorization model concluding that the proposed approach's separation between authentication and authorization supports a more flexible and secure authorization scheme while simultaneously improving the efficiency of the assignment of privileges.

The rest of the paper is organized as follows. In Section 2 we introduce structured P2P networks, Attribute Certificates and present an overview of existing access control mechanisms for structured P2P networks along with their drawbacks which are discussed in Section 3. Section 4 presents our proposed framework for authorization in structured P2P networks. In Section 5 we evaluate the proposed approach by applying it to RELOAD and comparing it with the RELOAD's identity-based model. Finally, Section 6 presents the conclusions of the research conducted in this paper.

2. Related works

In this section we present the state of the art efforts in the area of structured P2P networks, access control mechanisms for structured P2P networks and introduce Attribute Certificates.

2.1. Structured P2P networks

Structured P2P networks maintain a Distributed Hash Table (DHT) that makes each node responsible for a specific part of the content in the network. These networks employ hash functions to assign identifiers to each node and content in the network. In this way, when a node wants to access certain resources, it first determines the node responsible for them and directs its search towards it.

One of the most popular structured P2P networks is Chord [12], which is used in the RELOAD protocol [3]. Chord uses a logical ring as the underlaying structure for the routing of messages and the searching for keys. Within this ring, nodes are ordered clockwise, from 0 to $2^m - 1$ (being m the size in bits of the identifiers), according to their node ID. A hash function is used to create the Key IDs for any content (information) to be stored in the Chord network. Each node is responsible for storing all the Key IDs that are equal to or less than its own identifier but larger than the identifier of its predecessor in the ring. Also, for routing purposes, each node maintains a routing table with its predecessor and its successor in the ring, and a set of links to nodes located at different parts of the ring called fingers. A good survey of Peer-to-Peer overlay network schemes can be found in [13].

2.2. Access control in structured P2P networks

Early P2P systems were intended for file-sharing purposes. As a result access control solutions were primarily influenced by file sharing for these structured P2P systems. Early structured P2P networks' access control was based on the generation of node-IDs by hashing a 'unique' property of each node such as its IP address [1,12] or its public key [14]. The use of cryptographic puzzles, first described in [15], was also proposed in the literature to control the access of nodes to the network [5]. Besides, it is worth mentioning other decentralized access control systems based on the use of CAPTCHAs [16], the web of trust [2], social networks [17], a shared secret [3] or decentralized certification [4].

However, the paper on *The Sybil attack* [18] shows that, without a logically centralized authority, it is impossible to limit the number of identities a user can obtain to access the network except under extreme and unrealistic assumptions of resource parity and coordination among entities. Taking into account this research, [5] and [19] proposed the introduction of an offline centralized CA in the system that assigns to each user a X.509 PKC [20] binding a node-ID, chosen randomly by the server, to a public key generated by the client and its username. These proposals are the basis of the access control schemes followed by actual structured P2P networks such as RELOAD [3]. A deeper analysis of the security of these schemes can be found in [21].

With the aforementioned schemes, PKCs represent two roles: user's authentication and user's authorization (privileges of the user in the network: usernames allowing it to join the network and to have a location in the ID space to store its resources, nodeIDs establishing its location in the network and the resources it is responsible for, storage quota limiting the amount of data it can store in the network, etc.). In addition, access control over the P2P system's resources is built around these privileges declared in the user's PKC. P2P systems, like OceanStore [22] and Fairsite [23], use local Access Control Lists (ACLs) to determine the privileges of each user over an object. Each resource has an ACL associated that contains the PKs of the users authorized to access it. A user's request is digitally signed so that the node responsible for the resource can check that the user's access is authorized. A similar approach is used in other systems (such as RELOAD [3]) to enable resource sharing by delegated ACLs [10]. Unfortunately, due to the fact that in some cases the node responsible for a resource may be malicious and give free read access to all the users of the network to a private resource it is responsible for, an additional security mechanism such as the use of cryptography (resource encryption) in conjunction with ACLs should be used (as described in [24,22]).

2.3. Attribute Certificates

Attribute Certificates are used for the management of privileges. These certificates are supported by an Attribute Authority (AA). This kind of entity complements the functionalities of the Certificate Authority (CA). But, instead of establishing certification of the identities associated with a particular public key, the AA associates privileges to a PKC issued by another entity, with a different policy of certification, lifetime, etc.

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

The concept of AC is thoroughly discussed in the ITU X.509 standard [20] which establishes its definition and structure (that we present in detail in Section 4.3), and recently in the RFC5755 [6]. This idea arises from the problems of using one single certificate to establish identity and privileges of a user. PKCs include privileges into the certificate through the use of the extension 'Subject directory attributes'. However, the problem with PKCs is that they are designed to last for relatively long periods of time especially when compared with the frequency of change of rights or privileges. If a PKC is also used for this purpose, it is necessary to make a new one containing such privileges, and then revoke it whenever the privileges change.

Recent works [25,26], have already presented the advantages of using Attribute Certificates (ACs) for authorization over traditional approaches (such as Kerberos [27] and Microsoft .NET passport [28]) in distributed environments. However, to the best of our knowledge, no previous research has discussed the AC model in P2P systems and has presented a framework for authorization based on ACs to satisfy the special requirements of structured P2P networks.

3. Access control discussion

As we have noted previously, structured P2P networks' access control is based on a combination of PKCs and ACLs that presents several drawbacks. On the one hand, the use of the same PKC for both the authentication and authorization of users is not a good idea:

- The fact that the user's privileges are included within the certificate that also grants its access to the network determines that all the privileges will have the same lifetime, and the same as the certificate of identity of the user. This is not acceptable because we may want to allow a user to access different network services (voicemail, more storage, new identifiers, etc.) during different periods of time.
- Any change in any of the privileges already incorporated in the PKC forces the creation of a new PKC to update all the privileges.
- The inclusion of new privileges for a user (due to the inclusion of new resources, the acquisition of new privileges) also forces the creation of a new PKC.
- The use of short-lived certificates is usually recommended for P2P systems. However, a change in any of the privileges included in the user's PKC is enough to make the certificate invalid. The use of certificates of a very short duration would increase the load of the system while the lack of a specific revocation method for P2P systems makes it impractical to use an alternative revocation method because of the cost of including the necessary infrastructure to implement a traditional revocation system.
- From a security perspective, we found that it is necessary to separate the network infrastructure from the applications or services running over it. We therefore need a schema where different providers (users or companies) could offer different services or applications to the users of the network even if the network is managed by another provider using simple and standardized mechanisms.
- Since users' PKCs must be available to all the other users of the system to allow their authentication (including all the user's privileges in the same PKC), attacks on the privacy (need to know) of users by disclosing all their privileges are possible.

On the other hand, the existing resource's access control mechanisms do not fulfill all the security and flexibility requirements by themselves. The use of ACLs in structured P2P networks presents several issues:

- This mechanism works well with simple access control policies that are publicly accessible or private. However, the lack of a standardized format for ACLs and the fact that they were not designed with distribution systems in mind make the definition of more complicated access control policies in P2P systems difficult and application specific.
- Due to the fact that a structured P2P network replicate contents in several locations of the network, a change in the ACL of one resource forces a change in all the replicas even if the content has not been modified. It is not efficient to have to contact all the responsible nodes for a resourceID in a decentralized network each time we want to modify its access control policy and it should be avoided whenever possible.
- ACLs need to be public to permit users requesting a shared resource to check its integrity. This is a serious privacy issue.

Following the observations above, in the next section we propose a new authorization framework for structured P2P networks based on the use of Attribute Certificates to manage the privileges of users over the resources of the network.

4. Proposed authorization framework for structured P2P networks

After identifying the deficiencies of existing authorization mechanisms for structured P2P networks, we propose a new authorization framework for structured P2P networks. The proposed framework represents an improvement in the flexibility and user's privileges management of the system compared to existing proposals. Some of these improvements are:

- Separation of user's proof of identity (PKC used for authentication) from user's privileges (ACs used for authorization) that may be provided by different entities and have different lifetimes, in comparison to existing identity-based proposal where both concepts are mixed in the same PKC and should have the same lifetime and be provided by the same entity.
- Several alternatives can be used as source of authentication, either internal to the system (self-signed certificates, offline CA, etc.) or external (TTP PKCs, Electronic Identity Cards, etc.), making it suitable for both open-access and restricted networks. Existing approaches can only use internal sources of authentication.
- Homogeneity of the access control using the same authorization schema, in comparison to existing approaches where some privileges are included in the users PKC (access to the network, access to the users' reserved resources, etc.) and others specified using system specific ACLs (access to shared and delegated resources).
- More efficient and anonymous access control policy over the resources, in comparison to the use of ACLs that are neither efficient (must be replicated with each resource) nor private (must be public in order to be checked).
- Extra services may be provided by different entities (companies, users, etc.) within the network using the same authorization scheme, in comparison to existing approaches where the scheme is not extensible and in order to provide extra services ad-hoc mechanisms must be used.
- Inclusion of a distributed revocation system established within the structured P2P network that does not require the intervention of any external server or trusted intermediate authority, in comparison to existing approaches where either no revocation is available or an external server must be contacted.

In order to achieve these improvements, as start point we use the recently published Internet Attribute Certificate Profile for Authorization [6] that we adapt and extend to develop an attribute-based authentication framework suitable for structured P2P networks.

Fig. 1 shows an overview of the architecture of several possible scenarios that can be addressed with our proposal. On the left side of the figure we can see several examples of authentication methods that can be used; some of them decentralized, such as a user generated self-signed certificate, and others centralized, such as an Electronic ID Card or an Identity certificate generated by an offline CA. We can also

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

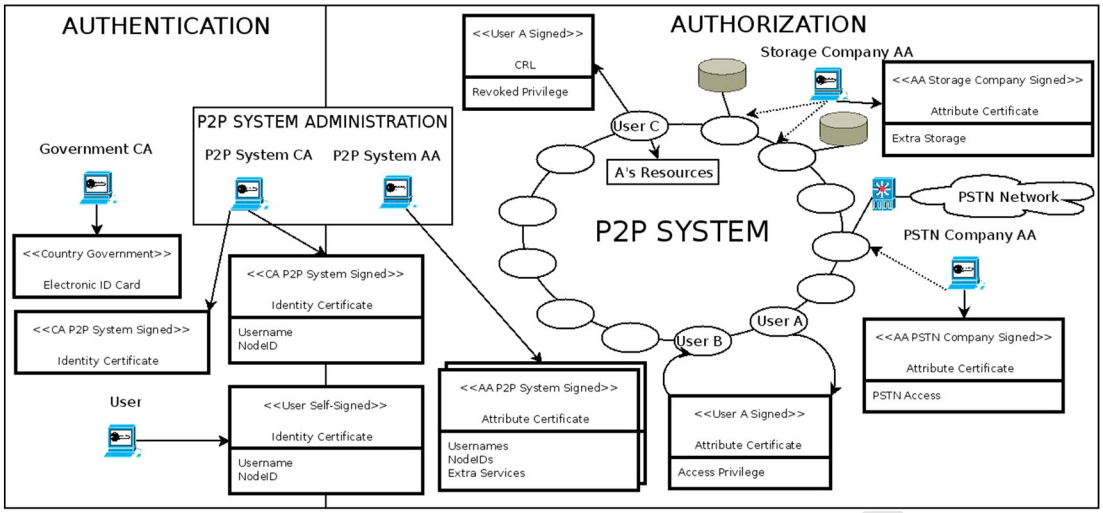

Fig. 1. Overview of possible authentication and authorization architectures for P2P scenarios.

differentiate between what we call pure certificates (situated completely on the left side of the figure, only containing authentication information) and hybrid certificates (situated in the middle of both sides of the figure, containing both authentication and authorization information). In turn, the right side of the figure represents possible authorization scenarios: a user A authorizing a user B accessing A's resources stored at a user C in a decentralized P2P system, a System AA providing the necessary credentials to access a centrally-managed structured P2P network or some companies issuing privileges allowing the access to extra services they provide through the structured P2P network infrastructure. Also, we can see how revocation information related to the issued privileges can be stored and checked within the network.

In the rest of this section we first introduce our proposal's application scenarios and the assumptions we made about the used P2P network. Then, we present separately how the user authentication and authorization is done in our proposal including the data structures and the flow of communication used.

4.1. Application scenarios and assumptions

Before describing the main components of our architecture, in this section we describe its intended scenarios and the assumptions we do about the used P2P overlay network.

As discussed in the previous section, our proposal is intended for both decentralized and centralized P2P systems with any network access policy (open-access, restricted, etc.). In relation to the structure of the P2P network, although it could be used with both, our proposal is more intended for structured than for unstructured P2P networks. For structured P2P networks, where resources are distributed along the network using a Distributed Hash Table (DHT) (users neither have direct control nor store their own resources), our application framework perfectly suits to control the access to the users' resources. However, in unstructured P2P networks users typically store and have direct control over their own resources, making the application of our proposal less practical. We assume, therefore, the use of a structured (Distributed Hash Table (DHT)-based) P2P network such as, for example, Chord [12]. This DHT overlay implements replication (being the number of replicas configurable) to prevent a malicious node from denying the existence of a resource. Also, resources can be signed and timestamped to prevent malicious resource modification and replay attacks.

Users have a nodeID and, usually, a username[1] which can be assigned using any of the methods we describe later in Section 4.2. These credentials also determine some locations of the network (resourceIDs) where they can store, modify and share their resources: resourceID = Hash(nodeID) and resourceID = Hash(username). Besides, despite the fact that its name seems to represent a single entity, each resourceID may contain several resources of different types. Since we want our framework to be application independent, we do not define the specific resourceIDs each user has privileges and the kind or the amount of data they can contain, but only assume that each user, based on the application's specifications used, has privileges over certain resourceIDs.

4.2. User authentication

We assume that all the possible users of the system have a X.509 PKC compliant with the standard described in the RFC5280 [7] that grants their identities. This certificate could be issued using any of the existing certification models, either decentralized or centralized, presented in the literature and already discussed in Section 2.2: by self-generation (users' self-signed certificates), by a decentralized certification authority formed by the members of the network themselves, by a system certification authority (either online or offline), by any external certification authority such as the government of a country that issues an electronic ID card to all its citizens, etc.

It is desirable that these PKCs are only a proof of the users' identity and do not authorize any access to the network or its resources. This approach has two main advantages with respect to identity-based authorization models: first, the user identity certificate does not necessarily have to be issued within the system and external sources of authentication can be used (e.g. Electronic ID Card). Second, and as a consequence of the first advantage, a user can have a single identity certificate (as it should be because the identity of a user is unique) and can use it as a source of authentication with any system instead of requiring one different identity certificate for any of the systems it has access to.

However, we are aware that our approach is not feasible in certain scenarios. Some already developed structured P2P networks, such as the based on the RELOAD protocol [3], use an identity-based authorization model where PKCs (either issued by the system's CA or self-signed by the users themselves) include, apart from the user's identity, a few privileges (specifically its username and nodeID). In such cases, our proposed authorization model serves to complement the privileges already defined in the user 's PKC. Another proposal exists for split certification in structured P2P networks [29] where users and devices are represented by different PKCs. Although this proposal is not specifically analyzed in this paper, the application of our authorization framework to such an

[1] Excluding some special applications and entities such as gateways that might only require a valid nodeID.

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

authentication proposal is straightforward, using ACs to assign privileges not only to users, but also to devices.

4.3. User authorization

Our proposed authorization framework is based on the recently published Internet Attribute Certificate Profile for Authorization [6]. In it, authorization is granted using X.509 ACs [20] which associate privileges with the identity of the user defined in its PKC. ACs allow the privileges to have a different policy of certification, lifetime, etc. than the user's PKC. Moreover, ACs can be issued by several entities (Authorization Authorities, users granting access to their resource to other users, etc.) different from the issuer of the user's PKC. In the rest of this section we introduce the data structures used and the processes to be followed in order to assign, revoke and use privileges with the proposed approach.

4.4. Attribute Certificates

Before describing how privileges can be assigned, revoked and used within the proposed approach, we introduce the data structure used in our proposed system to represent them, i.e. the Attribute Certificates (ACs). ACs serve to issue any possible privilege in the system: privileges issued by one user (e.g. user A allows user B accessing A's resources), privileges to access the system (e.g. username and nodeID), privileges to access services offered by others within the system (PSTN Gateway), etc.

The structure and fields of the ACs (following the profile standardized in [6]) used in our system are described in Table 1.

Table 1
Attribute Certificate structure.

Attribute Certificates	
Field name	Description
acinfo.version	Represents the version of the AC used. It should be *v2* (1)
acinfo.holder.baseCertificateID.issuer	References the PKC to which this AC applies (the user who receives the privileges). This field represents the issuer of the holder's PKC (creator of the user PKC). It must be equal to the field in the PKC of the holder (user who receives the privileges)
acinfo.holder.baseCertificateID.serial	References the PKC to which this AC applies (the user who receives the privileges). This field represents the serial number of the holder's PKC (creator of the user PKC). It must be equal to the field in the PKC of the holder (user who receives the privileges)
acinfo.issuer.v2Form.issuerName	Name (in its PKC) of the issuer (entity assigning the privileges)
acinfo.signature	Algorithm's identifier used to validate the AC. It can be any of the algorithms defined in the standard [20]
serialNumber	Serial number of the AC. The pair issuer/serialNumber must be unique
acinfo.attrCertValidityPeriod.notBeforeTime	Start of the period of validity of the certificate
acinfo.attrCertValidityPeriod.notAfterTime	End of the period of validity of the certificate
acinfo.attributes.type[].value[]	Set of privileges (not described here because they are application specific) the AC gives to the holder (user)
acinfo.extensions.authorityKeyIdentitifer.keyIdentifier	ResourceID where the PKC of the AC's issuer is stored. This field and the next two allow the PKC of the AC's issuer to be found in order to check the validity of the AC
acinfo.extensions.authorityKeyIdentitifer.authorityCertIssuer	Issuer of the PKC of the AC's issuer
acinfo.extensions.authorityKeyIdentitifer.authorityCertSerialNumber	Serial Number of the PKC of the AC's issuer
acinfo.extensions.crlDistributionPoints	This field must only be included if revocation of this certificate is possible and must point to the resourceID where the revocation information can be found
acinfo.extensions.noRevAvail	This field must only be included if revocation of this certificate is not possible. It includes no data
signatureValue	Signature of the issuer on the certificate

4.5. Assignment of privileges

We describe below how privileges are assigned in the proposed approach using the described ACs. The flow of the assignment process is graphically presented in Fig. 2. In this figure, we can see two different entities: Requester (entity willing to acquire a new privilege) and Issuer (entity granting the privilege). Requester is always a user of the network, while Issuer can be another user of the network or a third entity providing services within the network. Also, if the network is centralized, Issuer can be an administrative entity (such as an offline CA) used to acquire the privileges needed to access the network.

Despite the fact that usually the communication between both entities goes through several intermediate hops within the structured P2P network, it is shown in the figure as a direct communication for simplicity. This process has five steps:

- *Step 1*: Requester creates a request REQ (the structure of this message is application dependent) to demand some privilege and signs REQ with its private key PrKreq.
- *Step 2*: Requester attaches to REQ its credentials (PKCreq + ACreq or only PKCreq depending on the authentication model used: pure, hybrid, etc.) and sends it to Responder.
- *Step 3*: After receiving the request, Issuer:
  1. Checks requester's credentials; i.e. PKCreq and ACreq (if needed) satisfy the network access policy (self-signed or signed by a CA or …). Issuer also checks that the signature of REQ has been done with PrKreq, i.e. the PrK related to the presented PKCreq, in order to check that Requester is actually who it says it is.
  2. If the previous check was successful, Issuer checks that Requester satisfies the requirement needed to obtain the requested privilege (these requirements are application specific).
  3. If all the previous checks were successful, Issuer creates an AC (ACpriv) following the structure presented in Table 1 that is signed with the Issuers PrK (PrKiss) and grants the privilege. If any of the verification check fails, Issuer creates an answer ANS, signed with Issuers PrK (PrKiss), denying the privilege. This negative ANS is application specific and could include more information about why the privilege was denied.

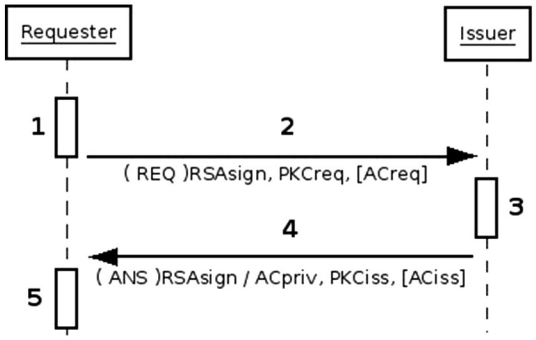

Fig. 2. Assignment of privileges.

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

- *Step 4*: Issuer attaches to ANS or ACpriv (depending on the decision) its credentials (PKCiss and, if necessary, ACiss) that justify that Issuer is the owner of the resource referenced by the privilege and sends it to Requester.
- *Step 5*: Finally, after receiving the answer, Requester:

1. Checks Issuer's credentials; i.e. PKCiss and ACiss (if needed) satisfy the network access policy (self-signed or signed by a CA).
2. If the previous check was successful, Requester checks that the answer is legitimate, i.e. the signature of ANS or ACpriv has been done with PrKiss (PrK related to the presented PKCiss). Also, if the answer was an ACpriv granting the privilege, Requester checks that ACpriv points to the PKCiss as issuer and that Issuer is actually the owner of the resource ACpriv references (e.g. Hash(Issuer Username) == Resource ID referenced by ACpriv).
3. Finally, if all the previous checks were correct and the answer was an ACpriv, Requester stores ACpriv. If all the previous checks were correct and the answer was an ANS denying the privilege, Requester can analyze the answer to find out why the privilege was denied. Otherwise, if the previous checks were not correct, Requester discards the answer.

4.6. Revocation system

It seems reasonable for us to use short-lived certificates for granting the user's privileges because this access is usually temporal: either related to the necessity of temporally accessing the resources provided by other members of the network or to some kind of temporal (daily, weekly or monthly) subscription when the network is centrally managed. Nevertheless, it is possible that these privileges have a longer duration (such as a year or more) mainly when a hybrid proposal is used and some of the user's privileges are included in the user's identity PKC. In this case, to have a revocation alternative is reasonable.

Unfortunately, the traditional client–server revocation scheme is not feasible for P2P systems because a CRL server must exist that should be contacted every time a user wants to check a PKC or AC. With this in mind, we have developed a fully distributed revocation scheme for our proposed framework.

In the rest of this section we introduce the specific CRLs used in our proposed revocation system, where to store them and the process that must be followed to revoke a privilege.

4.6.1. Certificate Revocation List

In traditional client–server systems, CRLs are related to several certificates (containing revocation information of all the users of the system). However, the specific nature of P2P systems requires the CRLs to be used in a different way. In our approach, a CRL is issued independently for each privilege (one CRL for each revoked AC) and includes only its revocation information. This way, the revocation information related to an AC can be easily found by requesting the resource specified in the AC (using, for example, the typical dictionary access mode with the issuer and the serial number of the AC as key). The revocation information related to the users' privileges is distributed among the nodes of the network to prevent the system from handling huge CRLs. Moreover, the fact that CRLs can be discarded once the lifetime of the AC has ended also reduces the amount of data related to revocation that has to be stored.

Similarly, in client–server approaches CRLs are issued periodically which is unsuitable for P2P systems. In our approach, a CRL is only issued when the privilege it references has been revoked. Therefore, while an AC is valid, its issuer does not have to contact the responsible node for the CRL to update it. This prevents the issuer of the privilege from having to periodically contact the node responsible for the privilege (particularly critical when the issuer of the privilege is an external or centralized entity, such as an AA issuing the user's usernames), and has to do it only once when the AC is revoked. A CRL can be discarded when the lifetime of the AC associated with it ends.

The structure and fields of the CRLs (following the profile standardized in [7]) used in our system for the revocation of privileges are presented in Table 2.

4.6.2. Location of revocation information

To take advantage of the structured P2P network facilities and to avoid the inclusion of extra entities in the system, we propose to store the revocation information relative to the ACs issued by a member of the network in the same resourceIDs where the resources these ACs reference are stored. We have made this decision because these are the only resourceIDs where the issuer can store information in the network. This specific location is referenced in all the ACs using the extension field crlDistributionPoints.

However, there are cases (such as when a hybrid proposal is used) where revocation information is not only required about the ACs but also about the used PKCs. For such cases, where PKCs do not have the crlDistributionPoints extension, the resourceID can be calculated by a function of the specific network properties such as, for example, resourceID = Hash(user's username) or resourceID = Hash(user's nodeID).

Table 2
Certificate Revocation List.

Certificate Revocation List	
Field name	Description
tbsCertList.version	Represents the version of the CRL used. It should be *v2* (1)
tbsCertList.signature	Algorithm's identifier used to validate the CRL. It can be any of the defined in the standard [20]
tbsCertList.issuer	Name (in its PKC) of the issuer of the CRL (entity revoking privileges)
tbsCertList.thisUpdate	Date of the CRL
tbsCertList.nextUpdate	Since in the proposed approach a CRL is related to only one privilege (AC) once it has been revoked no further updates are needed for it because the revocation is final. Therefore, to be consistent with the standard, this field (usually used for the date of the next CRL update) has to be a later date than the expiration date (*notAfterTime* field) of the AC to which the CRL is related
tbsCertList.revokedCertificates.userCertificate	Contains the serial number of the revoked certificate
tbsCertList.revokedCertificates.revocationDate	Contains the date of revocation of the revoked certificate
tbsCertList.crlExtensions.AuthorityKeyIdentifier.keyIdentifier	It points to the resourceID where the PKC of the CRL's issuer is stored, therefore identifying the public key to be used to verify the signature of this CRL. This field and the next two allow the PKC of the CRL's issuer to be found in order to check the validity of the CRL
tbsCertList.crlExtensions.AuthorityKeyIdentifier.authorityCertIssuer	Issuer of the PKC of the CRL's issuer
tbsCertList.crlExtensions.AuthorityKeyIdentifier.authorityCertSerialNumber	Points to the serial number of the PKC of the CRL's issuer
tbsCertList.crlExtensions.crlNumber	Sequence number of the CRL
signatureAlgorithm	Algorithm's identifier used to validate the CRL. It can be any of the defined in the standard [20]
signatureValue	Signature of the issuer on the CRL

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

4.6.3. Revocation of privileges

To revoke a privilege of any kind, the Issuer (entity that previously granted a privilege and now wants to revoke it) has to create a CRL and has to send it to one of the nodes Responsible for the resourceID the privilege references. The flow of the process is presented in Fig. 3:

- *Step 1*: Issuer creates a CRL (following the structure presented in Table 2 that includes all the revocation information and it is signed with Issuer's PrK, PrKiss) to request the revocation of a privilege.
- *Step 2*: Issuer attaches to CRL its credentials (PKCiss and, if necessary, ACiss proving it as the issuer of the privilege it wants to revoke). Then localizes the ResourceID where the information should be stored (following Section 4.6.2) and sends it to one of the nodes Responsible for this resourceID (the one chosen depends on the network topology plugin used).
- *Step 3*: After receiving the message with the CRL, Responsible:
  1. Checks Issuer's credentials; i.e. PKCiss and ACiss (if needed) satisfy the network access policy (self-signed or signed by a CA).
  2. If the previous check was successful, Responsible checks that the CRL is legitimate, i.e. it points to the presented PKCiss as issuer and its signature has been done with PrKiss (PrK related to the presented PKCisss).
  3. Then, Responsible checks that Issuer is actually the owner of the resourceID (e.g. Hash(Issuer Username) == ResourceID).
  4. If all the previous check were successful, Responsible stores the CRL and creates an answer ANS (signed with its PrK, PrKres) that confirms the requested operation. Otherwise, if any of the previous checks were unsuccessful, creates an answer ANS (signed with its PrK, PrKres) denying the operation.
- *Step 4*: Responsible attaches to ANS its credentials (PKCres and, if necessary, ACres) that justify that Responsible is one of the nodes responsible for storing the resource referenced by the ACpriv (and, therefore, for storing the CRL) and sends it to Issuer.
- *Step 5*: After receiving the answer, Issuer:
  1. Checks Responsible's credentials; i.e. PKCres and ACres (if needed) satisfy the network access policy (self-signed or signed by a CA or …).
  2. If the previous check was successful, Issuer checks that the answer is legitimate, i.e. the signature of ANS has been done with PrKres (PrK related to the presented PKCres), and that Responsible is actually one of the responsible nodes for storing the CRL (this last check depends on the specific topology plugin used, e.g. in Chord rate of closeness between the Responsible's NodeID and the ResourceID referenced by ACpriv).
- *Step 6*: If the operation was accepted, the topology plugin of the network replicates the new CRL to all the nodes responsible for the replicas of this resourceID. However, this replication can also be done directly by the Issuer by communicating with all the replicas. This second alternative has the advantage of preventing a single malicious node (the contacted responsible for the resource) from preventing the revocation and ensuring immediate consistency (revocation information is updated at the same time in all the replicas without having to wait for the topology plugin for updating the information). As drawbacks, this second approach requires more overhead with the Issuer which needs to contact all the replicas instead of relaying in the topology plugin stabilization.

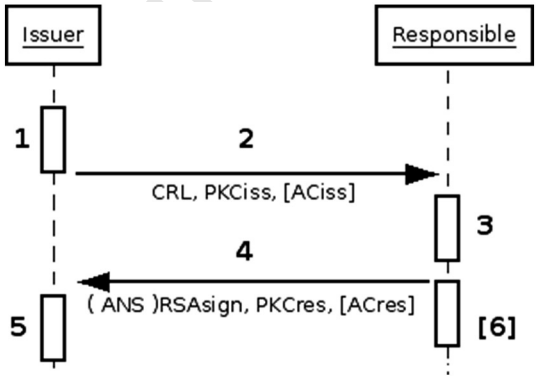

Fig. 3. Revocation of privileges.

4.7. Use of privileges

We describe below how users can make use of the privileges they are assigned using the described ACs. The flow of the process is presented in Fig. 4. In this figure, we can see two different entities: Accessor (entity willing to access a resource) and Responsible (entity responsible for the resource); and five steps:

- *Step 1*: Accessor creates a request REQ (the structure of this message is application dependent) to request access to a resource. Unlike the previous cases (assignment and revocation of privileges), authorization may not be needed to access to a resource (e.g. access a public resource, such as a CRL). In such cases it is not necessary to sign REQ. For the cases when authorization is needed (e.g. access a user's private resource), REQ must be signed with Accessor's private key (PrKacc).
- *Step 2*: If authorization is not needed, Accessor creates a message only including REQ. Otherwise, if some kind of authorization is needed, Accessor attaches to REQ its credentials (PKCacc + ACacc or only PKCacc depending on the authentication model used: pure, hybrid, etc.) and, if the privilege needed to perform the access is not included in Accessors credentials (e.g. privilege to access other user's resources), an AC containing the necessary privilege (ACpriv). To end this step, Accessor sends the message to Responsible.
- *Step 3*: After receiving the access request and if authorization is not needed, Responsible creates an answer ANS including either the requested resource (if it is available) or a negative answer (the resource does not exist). Otherwise, if authorization is needed, Responsible:
  1. Checks Accessor's credentials, i.e. PKCacc and ACacc (if needed) satisfy the network access policy (self-signed or signed by a CA or …). Also, if revocation is enabled in the system, it must check that the credentials have not been revoked. Two cases can happen here: Accessor is the owner of the resource and, therefore, its revocation information would be available locally, or Accessor is another user and the revocation information must be requested (using a non-authorized REQ) to the node of the network responsible for storing it.
  2. If the previous check was successful and the needed privileges to perform the access are included in Accessor's credentials (e.g. Accessor is trying to access a resource with resourceID = Hash(Username included in Accessors ACacc)) the access is granted. However, if the previous check was successful but the needed privileges to perform the access are not included in Accessor's credentials, Responsible has to check ACpriv:
     (a) ACpriv must reference the requested resource.
     (b) ACpriv must be legitimate, i.e. it points to the resourceID's owner PKC (PKCown) as Issuer and its signature has been done with the PrKown (PrK related to the resources owner PKC).

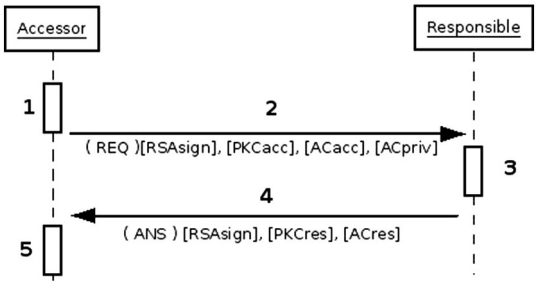

Fig. 4. Use of privileges.

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

(c) ACpriv must be valid, Responsible must check (locally) that no revocation information is available neither for the resource owner's credentials nor for the ACpriv itself.

3. If all the previous checks were successful, Responsible creates an answer ANS (signed with its PrK, PrKres) including either the requested resource (if it is available) or a negative answer (the resource does not exist).

- *Step 4*: Responsible sends ANS to Accessor and, if the ANS has been signed, also includes in the message its credentials (PKCres and, if necessary, ACres to prove it is in fact the responsible for the resource).
- *Step 5*: After receiving the answer and if authorization is not needed, Accessor can use the received information. Otherwise, if authorization is needed Accessor:

1. Checks Responsible's credentials, i.e. PKCres and ACres (if needed) satisfy the network access policy (self-signed or signed by a CA or …).
2. If the previous check was successful, Accessor checks that the answer is legitimate, i.e. the signature of ANS has been done with PrKres (PrK related to the presented PKCres), and that Responsible is actually one of the responsible for storing the resource (this last check depends on the specific topology plugin used, e.g. in Chord rate of closeness between the Responsible's NodeID and the ResourceID).
3. If all the previous checks were correct, it can use the received information.

5. Evaluation

In this section, we present an evaluation of our proposed authorization framework. In order to do this evaluation not only in absolute terms but also in relative terms, we have applied our proposed authorization framework to the IETF P2P standard, the RELOAD protocol [3], and we have evaluated it against the original identity-based authorization approach used by the protocol (previously described in Section 2).

We first analyze the performance of both models in terms of their communication and computational costs. This analysis has been carried out both theoretically and with simulations using the P2P overlay simulation framework OverSim [30] and shows that, in addition to its flexibility improvements (described in Section 4), our proposed approach reduces the cost of the assignment of privileges, has only a slight overhead associated with its verification and uses a very competitive distributed revocation mechanism. After the performance analysis, we highlight the advantages of the proposed approach while using standard methods. Finally, a study about the security of both schemes shows that the proposed approach not only maintains the security of identity-based authorization models but also offers additional security functionalities.

5.1. Performance of proposed authorization approach

We present an analysis of the cost of the main operations of our authorization proposal in the context of RELOAD [3] and we compare them with the cost of the RELOAD original identity-based approach. Communication cost is measured in terms of the number of messages required to establish communication and the number of cryptographic operations that should be performed to carry out an action. The computational cost to establish each communication is not included in our analysis because it depends on the protocol used: TCP/IP (Transmission Control Protocol/Internet Protocol) communication, TLS (Transport Layer Security), DTLS (Datagram Transport Layer Security), IPSec (Internet Protocol Security), etc. However, since this cost is constant and is independent of the authorization proposal used, this simplification does not affect our comparison results.

This analysis has been carried out both theoretically and with simulations using the P2P overlay simulation framework OverSim [30]. Before starting the analysis, we present the notations and the system configuration used in the tests:

- In our analysis we compare the RELOAD protocol [3] using our proposed authorization framework against its original identity-based authorization based on the usage for shared resources using delegation ACLs [10] with the following configuration:
  - The authentication model used for the test is a hybrid one with an open-access access control policy. In order to access to the network, users create a self-signed certificate that includes a nodeID = Hash(PK) and a username freely chosen by the user.
  - User resources are stored at ResourceID = Hash(NodeID).
- We define the parameter *Comm* to represent the cost required to establish a communication between two entities of the system.
- The operational cost is measured in terms of the main RSA-1024 operation whose performance[2] is RSA_{verify} 110,000 verify/s and RSA_{sign} 6000 sign/s.

5.1.1. Theoretical performance analysis

In this section we analyze theoretically the cost of assignment, revocation and use of privileges in both authorization proposals.

- Assignment of privileges: The cost of issuing privileges.
  1. Communication cost: In the proposed authorization approach a communication should be established between Requester and the Issuer of Privileges. In the identity-based authorization approach a communication should be established between Requester and Issuer, another one between Issuer and Responsible for the resource to modify the ACL of the resource affected and one extra communication between Responsible and each resource's replica to update them.[3] So the cost is One *Comm* in the proposed approach and 2 + *NumReplicas Comm* in the identity-based approach.
  2. Computational cost: In the proposed approach:

Requester		**Issuer**
Creates REQ, RSA_{sign}	REQ + PKCreq ⟶	Checks PKCreq, RSA_{verify} Checks REQ, RSA_{verify}
Checks PKCiss, RSA_{verify} Checks AC/ANS, RSA_{verify}	⟵ ANS/AC + PKCiss	Creates ANS or AC, RSA_{sign}
Total		
$RSA_{sign} + 2RSA_{verify}$		$RSA_{sign} + 2RSA_{verify}$

In the identity-based approach:

Requester		**Issuer**		**Responsible**
Creates REQ, RSA_{sign}	REQ + PKCreq →	Checks PKCreq, RSA_{verify} Checks REQ, RSA_{verify}		
		Creates ACL, RSA_{sign}	ACL + PKCiss →	Checks PKCiss, RSA_{verify} Checks ACL, RSA_{verify}
		Checks PKCres, RSA_{verify} Checks ANSres, RSA_{verify}	← ANSres + PKCres	Creates ANSres, RSA_{sign}
Checks PKCiss, RSA_{verify} Checks ANSiss, RSA_{verify}	← ANSiss+ PKCiss	Creates ANSiss, RSA_{sign}		
		Replica 1		
		Checks PKiss, RSA_{verify} Checks ACL, RSA_{verify}	← ACL + PKCiss	
		Creates ANSrep1, RSA_{sign}	ANSrep1 + PKCrep1 →	Checks PKCrep1, RSA_{verify} Checks ANSrep1, RSA_{verify}
		• • •		
		Replica N		
		Checks PKiss, RSA_{verify} Checks ACL, RSA_{verify}	← ACL + PKCiss	
		Creates ANSrepN, RSA_{sign}	ANSrepN + PKCrepN →	Checks PKCrepN, RSA_{verify} Checks ANSrepN, RSA_{verify}

Total	
Requester:	$RSA_{sign} + 2RSA_{verify}$
Issuer:	$2RSA_{sign} + 4RSA_{verify}$
Responsible&Replicas:	$(1 + NumReplicas)RSA_{sign} + (2 + 4 \times NumReplicas)RSA_{verify}$

[2] Using the OpenSSL (version 0.9.8 g) speed test in an Ubuntu 10.04 (lucid) 64-bits with kernel Linux 2.6.32-25 running over an Intel(R) Core(TM)2 Quad CPU Q8200 @ 2.33GHz with 4GB of RAM.

[3] As for the case of our proposal's revocation system, these communications could be also performed by Issuer. Anyway, the global overhead of the system is the same.

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

- *Revocation of privileges*: As we have pointed out earlier, we recommend the use of short-lived AC for most scenarios. However, there might be some scenarios or special privileges that may need revocation. Below, we compare the cost of revocation in our proposed authorization approach with the cost of revoking privileges in the identity-based one. It is important to note that while our approach allows revoking any kind of privilege, in the identity-based approach only the privileges included in the ACLs can be revoked and for any privilege included in the user's PKC an external and centralized revocation server is needed.

1. Communication cost: In the proposed authorization approach a communication should be established between the Issuer willing to revoke the privilege and the Responsible of the resource that the privileges references and one extra communication between Responsible and each resource's replica to update the new revocation information.[4] In the identity-based authorization approach a communication should be established between the Issuer willing to change the ACL and the Responsible of the resource that the ACL references and one extra communication between Responsible and each resource's replica to update the new ACL[4]. So the cost is $1 + NumReplicas\ Comm$ for both approaches.
2. Computational cost: In the proposed approach:

Issuer		Responsible		Replicas
Creates CRL, RSA_{sign}	CRL + PKCiss →	Checks PKCiss, RSA_{verify} Checks CRL RSA_{verify}		
Checks PKCres, RSA_{verify} Checks ANS, RSA_{verify}	← ANS + PKCres	Creates ANS, RSA_{sign}		
				Replica 1
			CRL + PKCiss →	Checks PKCiss, RSA_{verify} Checks CRL, RSA_{verify}
		Checks PKCrep1, RSA_{verify} Checks ANSrep1, RSA_{verify}	← ANSrep1 + PKCrep1	Creates ANSrep1, RSA_{sign}
				• • •
				Replica N
			CRL + PKCiss →	Checks PKCiss, RSA_{verify} Checks CRL, RSA_{verify}
		Checks PKCrepN, RSA_{verify} Checks ANSrepN, RSA_{verify}	← ANSrepN + PKCrepN	Creates ANSrepN, RSA_{sign}
Total				
Issuer:	$RSA_{sign} + 2RSA_{verify}$			
Responsible&Replicas:	$(1 + NumReplicas)RSA_{sign} + (2 + 4 \times NumReplicas)RSA_{verify}$			

In the identity-based approach:

Issuer		Responsible		Replicas
Creates ACL, RSA_{sign}	ACL + PKCiss →	Checks PKCiss, RSA_{verify} Checks ACL RSA_{verify}		
Checks PKCres, RSA_{verify} Checks ANS, RSA_{verify}	← ANS + PKCres	Creates ANS, RSA_{sign}		
				Replica 1
			ACL + PKCiss →	Checks PKCiss, RSA_{verify} Checks ACL, RSA_{verify}
		Checks PKCrep1, RSA_{verify} Checks ANSrep1, RSA_{verify}	← ANSrep1 + PKCrep1	Creates ANSrep1, RSA_{sign}
				• • •
				Replica N
			ACL + PKCiss →	Checks PKCiss, RSA_{verify} Checks ACL, RSA_{verify}
		Checks PKCrepN, RSA_{verify} Checks ANSrepN, RSA_{verify}	← ANSrepN + PKCrepN	Creates ANSrepN, RSA_{sign}
Total				
Issuer:	$RSA_{sign} + 2RSA_{verify}$			
Responsible&Replicas:	$(1 + NumReplicas)RSA_{sign} + (2 + 4 \times NumReplicas)RSA_{verify}$			

- *Use of privileges*: Cost of using the existing privileges in both proposals:

1. Communication cost: In both proposals, when a user wants to use a privilege the user has to establish a communication with the node responsible for it. So the cost is One *Comm* in both approaches.
2. Computational cost: In both approaches, if the access does not need to be authorized, there is no computational cost (in terms of cryptographic operations).
   When the authorization is needed in our approach:

Accessor		Responsible
Creates REQ, RSA_{sign}	REQ + PKCacc + ACpriv ⟶	Checks PKCacc, RSA_{verify} Checks REQ, RSA_{verify} Checks ACpriv, RSA_{verify}
Checks PKCres, RSA_{verify} Checks ANS, RSA_{verify}	⟵ ANS + PKCres	Creates ANS, RSA_{sign}
Total		
$RSA_{sign} + 2RSA_{verify}$		$RSA_{sign} + 3RSA_{verify}$

When authorization is needed in the identity-based approach:

Accessor		Responsible
Creates REQ, RSA_{sign}	REQ + PKCacc ⟶	Checks PKCacc, RSA_{verify} Checks REQ, RSA_{verify}
Checks PKCres, RSA_{verify} Checks ANS, RSA_{verify}	⟵ ANS + PKCres	Creates ANS, RSA_{sign}
Total		
$RSA_{sign} + 2RSA_{verify}$		$RSA_{sign} + 2RSA_{verify}$

Table 3 summarizes the theoretical performance cost analysis associated with the assignment, revocation and use of privileges in both approaches.

5.1.2. Simulation

To verify the theoretical performance analysis presented earlier, we have also simulated both authorization proposals with OverSim [30]. For these simulations we have used the same configuration as for the theoretical analysis plus some additional parameters:

- Simulation time: 5 days.
- Number of nodes of the network: 100, 1000 and 10,000.
- Interval of assignment of privileges: Every 8 h each node has a probability of 0.3 of assigning a new privilege.[5]
- Interval of revocation of privileges: Every 8 h each node has a probability of 0.2 of revoking a privilege[5].
- Interval of use of privileges: We adopt the model presented in [31] (80% of inactive nodes — nodes that do not issue queries, the time spent until a node issues its first query is modeled by a Weibull distribution of parameters $\alpha = 0.9821$ and $\lambda = 0.02662$, while the time spent between each of the following queries issued by a node is modeled by a Weibull distribution with a log-normal distribution of parameters $\delta = 1.625$ and $\mu = 3.353$).
- Number of replicas: 4.

As for the assignment of privileges, Fig. 4a and b shows the number of messages and cryptographic operations performed per node during the simulation respectively. The simulation supports the theoretical analysis results presented earlier showing that the proposed approach significantly lowers the cost of the assignment of privileges in structured P2P networks. The reason of this improvement is because the privileges in the proposed approach are sent directly to the user in an AC while in the identity-based approach they have to be granted by modifying the ACLs of the responsible node for the resource and all its replicas. The figures also demonstrate the good scalability of both our proposed approach and the identity-based approach: the overhead

[4] Again, these communications could be also performed by Issuer incurring the same global overhead.

[5] In contrast to the interval of use of privileges that has been chosen using existing researches, the intervals of assignment and revocation have been randomly chosen because no research was found related to the intervals of assignment and revocation of privileges in P2P systems.

Table 3
Theoretical performance analysis.

Performance analysis			
		Our proposal	Identity-based
*Assign	Communication cost	*One*	$(2 + NumReplicas)$
	Computational cost	$2RSA_{sign} + 4RSA_{verify}$	$(4 + NumReplicas)RSA_{sign} + (8 + 4NumReplicas)RSA_{verify}$
*Revoke	Communication cost	$(1 + NumReplicas)$	$(1 + NumReplicas)$
	Computational cost	$(2 + NumReplicas)RSA_{sign} + (4 + 4NumReplicas)RSA_{verify}$	$(2 + NumReplicas)RSA_{sign} + (4 + 4NumReplicas)RSA_{verify}$
*Use	Communication cost	*One*	*One*
	Computational cost	$2RSA_{sign} + 5RSA_{verify}$	$2RSA_{sign} + 4RSA_{verify}$

incurred by the nodes of the network during the assignment of privileges is independent of the network size.

In relation to the revocation of privileges, Fig. 5a and b shows the number of messages and cryptographic operations performed per node during the simulation respectively. The results show that the extra functionalities (revocation of any privilege including ACs and PKCs) provided by our decentralized revocation scheme does not incur any extra cost in comparison to the limited (it can only be used to control the access to the network resources) ACL system used by identity-based approach. Finally, in terms of scalability, with both approaches the overhead of the nodes of the network for the revocation of privileges is independent of its size.

Finally, in relation to the cost of using privileges, simulations (Fig. 6a and b) show similar results with both approaches in terms of messages and a slightly overhead in terms of cryptographic operations with our proposed approach (Fig. 7). This overhead, as we have seen in the theoretical analysis, is caused by the extra operation needed to check the ACpriv containing the user's privileges.

5.2. Standardization

All the mechanisms used in the proposed approach are based on well-known standards: PKCs, ACs and CRLs. This is a major advantage in comparison with identity-based approaches that usually use non-standardized and application specific ACLs that could present several issues such as interoperability problems, security threats or additional performance overhead due to an unclear definition.

5.3. Security

From a security perspective, our certification model maintains the security of the identity-based approach related to the authentication of users because all users of the system still hold a PKC. How these PKCs are obtained (self-signed, issued by an offline CA, etc.) will determine how resilient the system is to attacks such as *Sybil Attacks* [18], ID Mapping Attacks [32] and so on; being the security of the authentication independent from the authorization alternative used.

In relation to the authorization framework itself, as described in Section 4, all the requests and answers that need to be authorized are digitally signed. In the same way, all the PKCs, ACs and CRLs used are digitally signed. This ensures their authenticity and integrity. Also, in order to use a privilege it is necessary to prove the possession of the PKC the AC references ensuring, therefore, that only the intended user can use a privilege.

If additional security services are needed (e.g. confidentiality) we rely on the specific P2P routing algorithm used for the application. All

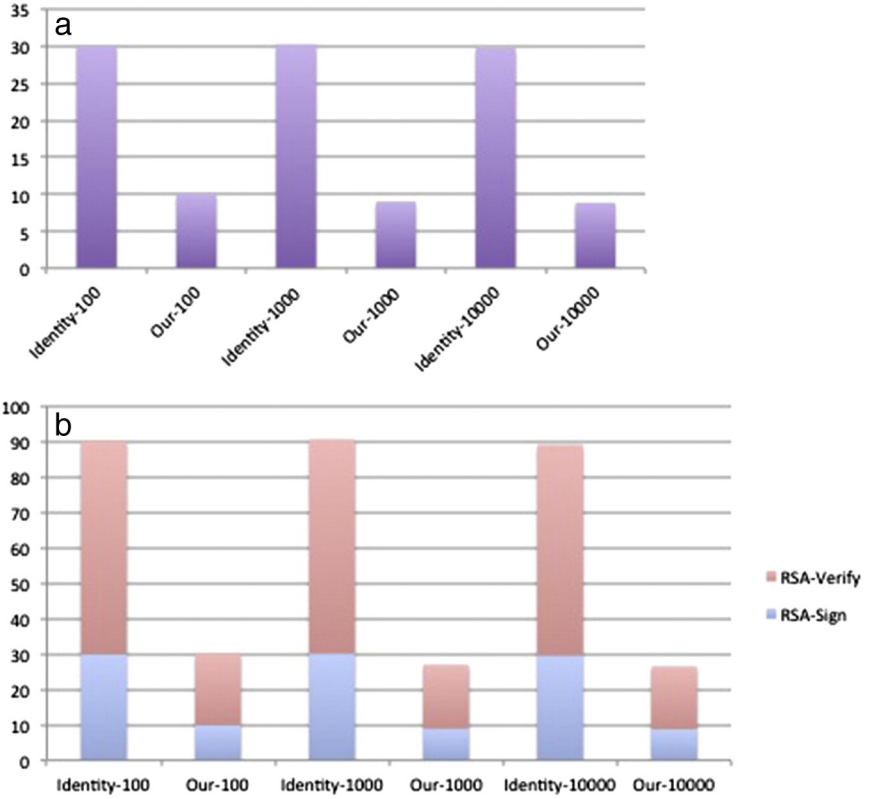

Fig. 5. Simulation results for the assignment of privileges.

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

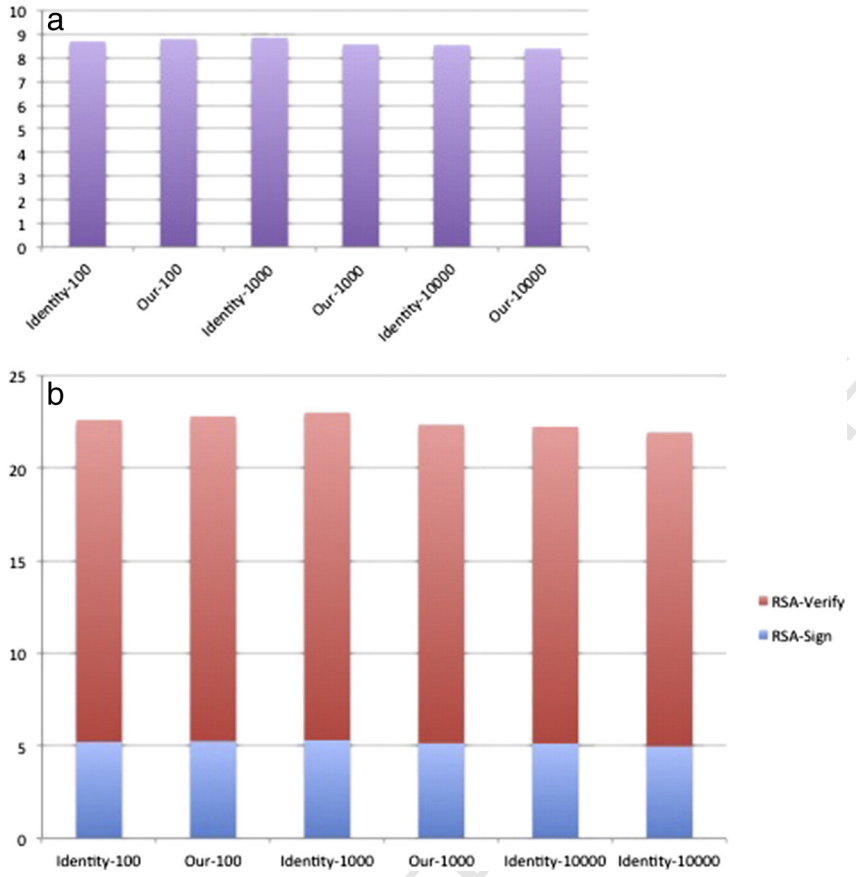

Fig. 6. Simulation results for the revocation of privileges.

the users of the network holding a PKC can be used to grant the authentication, integrity and confidentiality of the communications using the pair PK/PrK. Protocols such as TLS, DTLS or IPSec can be used to secure these communications hop-by-hop[6], while application specific solutions (encryption, unique identifiers) can provide complementary end-to-end protection. Again, choosing one mechanism or the other and its security is independent of the authorization proposal used. However, special attention should be given to replay attacks (attacker capturing a user's request and resending it later to get unauthorized access to a resource). As we commented before, typical end-to-end solutions (TLS, DTLS, etc.) cannot be used in P2P systems because most of the peers are not directly connected. Therefore, it is crucial that the application specific request contains some mechanism (timestamp, unique identifier, etc.) to prevent this kind of attack.

In relation to the distributed revocation, a reasonable decision (see Section 4.6.3) should be made about relying on the replication of the network topology plugin (more efficient) or making Issuer communicate with all the replicas (less efficient but quicker and more secure).

Finally, applications can implement resource encryption to avoid a malicious node responsible for a resource from revealing its content to unauthorized users, signature over the resources to grant their integrity and use replication to prevent a single malicious node from denying access to the resources it is responsible for (availability).

In addition, our proposed approach supports several security improvements compared to the identity-based approaches:

- The possibility of taking advantage of external trusted sources of authentication, such as Electronic Identity Cards.
- The inclusion of a distributed revocation mechanism that permits the easy revocation of both PKCs and ACs to prevent identity theft in case an attacker had compromised them. In our proposed revocation system, CRLs are stored in several locations that can be checked by a user. This prevents a malicious node responsible for a CRL from denying its existence (availability) while the CRL's signature prevents a malicious node responsible for a CRL from answering with a fake or modified CRL (authenticity and integrity).
- Support for privacy of the user's privileges. Privileges are provided using a different AC for each privilege. ACs are private to users and should only be presented to the node responsible for the resources. This is in contrast to identity-based approaches which must maintain public ACLs (that reveal to every node in the system which users have privileges over a resource).

6. Conclusion

In this research we have presented a new authorization scheme for structured P2P networks based on a clear differentiation between the concepts of authentication and authorization. This differentiation is built on the use of Attribute Certificates that link the privileges of a user within the system with the user's identity represented by a public key certificate. We have also presented a distributed revocation system that can be established within the structured P2P network and does not need the intervention of any external server.

Our proposed approach solves the limitations of identity-based approaches by allowing the definition of a finer-grain access control system over the systems' resources and the establishment of different durations for the user's privileges. The evaluation conducted on our proposed framework shows that it is not only more flexible than identity-

[6] Although these are typically end-to-end security solutions, they cannot usually provide such a service in a structured P2P network because not all peers are directly connected among them, but to a few peers known as neighbors.

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007

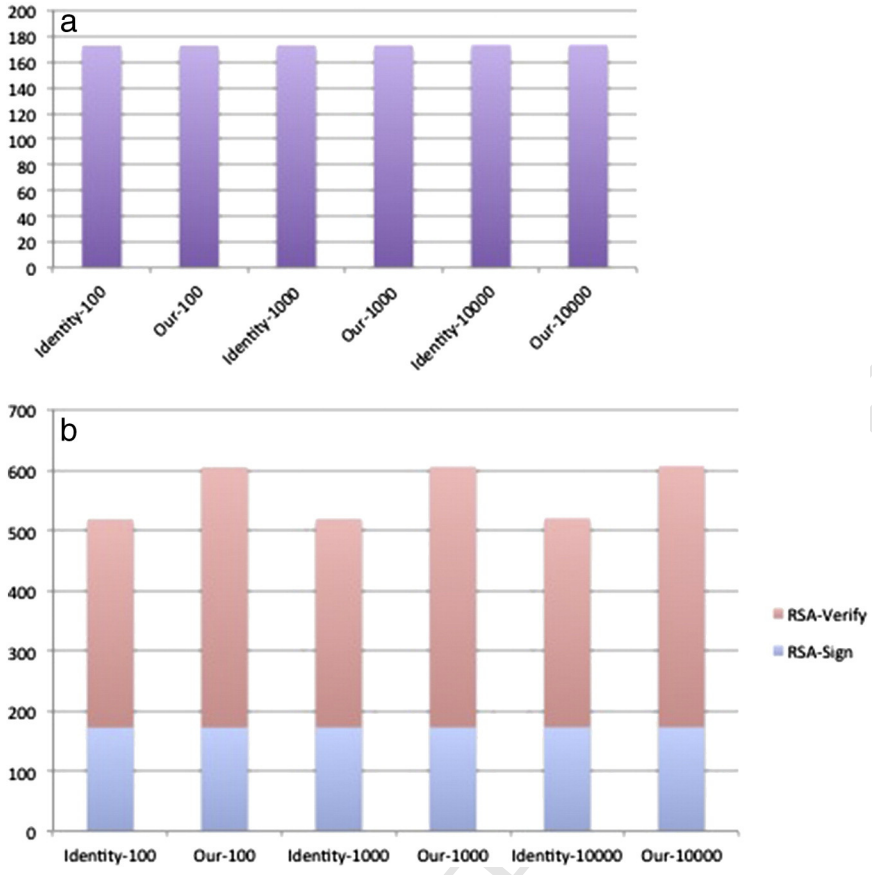

Fig. 7. Simulation results for the use of privileges.

based approaches but also more secure and efficient in the assignment of privileges while preserving its simple infrastructure. In addition, our proposed approach minimizes overheads involved when certificates are revoked.

Acknowledgments

This work was supported by the Ministry of Economy and Competitiveness of Spain through the Project INNPACTO IPT-2011-1328-390000 SMOTY: A Security System Based on Emergent Intelligence on the Internet of Things. We thank the anonymous reviewers for their feedback and useful comments, which have helped us improve the quality and presentation of this article.

References

[1] D.A. Bryan, B.B. Lowekamp, C. Jennings, SOSIMPLE: a serverless, standards-based, P2P SIP communication system, Proceedings of the First International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications, IEEE Computer Society, Washington, DC, USA 2005, pp. 42–49.
[2] P.R. Zimmermann, The Official PGP User's Guide, MIT Press, Cambridge, MA, 1995.
[3] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, H. Schulzrinne, RFC 6940: REsource LOcation And Discovery (RELOAD) Base Protocol, -draft-ietf-p2psip-base-26, Jan. 2014. (work in progress).
[4] D.S. Touceda, J.M.S. Cámara, M. Soriano, Decentralized certification scheme for secure admission in on-the-fly peer-to-peer systems, Peer-to-Peer Netw. Appl. 5 (2) (2012) 105–124.
[5] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, D.S. Wallach, Secure routing for structured peer-to-peer overlay networks, OSDI '02, Proceedings of the 5th Symposium on Operating systems Design and Implementation, ACM, New York, NY, USA 2002, pp. 299–314.
[6] S. Farrell, R. Housley, S. Turner, An Internet Attribute Certificate Profile for Authorization, RFC 5755, Jan. 2010. (Standard).
[7] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 5280, May 2008. (Proposed Standard).
[8] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol — OCSP, RFC 2560, Jun. 1999. (Proposed Standard).
[9] S. Santesson, P. Hallam-Baker, Online Certificate Status Protocol Algorithm Agility, RFC 6277, Jun. 2011. (Proposed Standard).
[10] A. Knauf, T.C.S. abd, G. Hege, M. Waehlisch, Internet-Draft: A Usage for Shared Resources in RELOAD (ShaRe), draft-knauf-p2psip-share-05 (Work in Progress), Mar. 2015.
[11] J. Buford, M. Kolberg, Application-layer Multicast Extensions to REsource LOcation And Discovery (RELOAD), RFC 7019 (Experimental), Sep. 2013.
[12] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup service for internet applications, Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '01, ACM, New York, NY, USA 2001, pp. 149–160.
[13] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, A survey and comparison of peer-to-peer overlay network schemes, IEEE Commun. Surv. Tutorials 7 (2) (2005) 72–93, http://dx.doi.org/10.1109/COMST.2005.1610546.
[14] E. Sit, R. Morris, Security considerations for peer-to-peer distributed hash tables, Revised Papers from the First International Workshop on Peer-to-Peer Systems, IPTPS '02, Springer-Verlag, London, UK 2002, pp. 261–269.
[15] R.C. Merkle, Secure communications over insecure channels, Commun. ACM 21 (1978) 294–299.
[16] L. von Ahn, M. Blum, N. Hopper, J. Langford, The Official CAPTCHA Site, http://www.captcha.net/2000 (URL http://www.captcha.net/).
[17] H. Yu, M. Kaminsky, P.B. Gibbons, A. Flaxman, SybilGuard: Defending Against Sybil Attacks via Social Networks, SIGCOMM Comput. Commun. Rev. 36 (2006) 267–278.
[18] J.R. Douceur, The Sybil Attack, Revised Papers from the First International Workshop on Peer-to-Peer Systems, IPTPS'02, Springer-Verlag, London, UK 2002, pp. 251–260.
[19] D. Bryan, B. Lowekamp, M. Zangrilli, The Design of a Versatile, Secure P2PSIP Communications Architecture for the Public Internet, IEEE International Symposium on Parallel and Distributed Processing, IPDPS, IEEE Computer Society, Washington, DC, USA 2008, pp. 1–8.
[20] ITU, ITU-T Recommendation X.509: The Directory: Public key and attribute certificate frameworks, Tech. rep., ITU, 2005.
[21] D. Touceda, J. Sierra, A. Izquierdo, H. Schulzrinne, Survey of Attacks and Defenses on P2PSIP Communications, IEEE Commun. Surv. Tutorials 14 (3) (2012) 750–783.
[22] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B. Zhao, OceanStore: An Architecture for Global-Scale Persistent Storage, ACM SIGPLAN Not. 35 (11) (2000) 190–201.
[23] A. A., et al., FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment, Proceedings of the 5th symposium on Operating systems design and implementation, OSDI'02, ACM, New York, NY, USA 2002, pp. 1–14. Q10

[24] D.A. Bryan, B. Lowekamp, Innovations in Peer-to-Peer Communications, Proceedings of the 2006 Virginia Space Grant Consortium Research Conference, 2006.
[25] J. Lopez, R. Oppliger, G. Pernul, Authentication and Authorization Infrastructures (AAIs): A Comparative Survey, J. Comput. Secur. 23 (7) (2004) 578–590.
[26] M.F. Hinarejos, J.L. Muñoz, J. Forné, O. Esparza, Preon: An efficient cascade revocation mechanism for delegation paths, J. Comput. Secur. 29 (6) (2010) 697–711.
[27] C. Neuman, T. Yu, S. Hartman, K. Raeburn, The Kerberos Network Authentication Service (V5), RFC 4120 (Proposed Standard), updated by RFCs 4537, 5021, Jul. 2005.
[28] Microsoft, .net passport: balances authentication solutions, Tech. rep., Microsoft, 2002.
[29] D. Touceda, J. Camara, L. Villalba, J. Marquez, Advantages of identity certificate segregation in P2PSIP systems, Commun. IET 5 (6) (2011) 879–889.
[30] K. I. f. T., Institut für Telematik, The oversim p2p simulatorURL http://http://www.oversim.org2013.
[31] D. Stutzbach, R. Rejaie, Understanding churn in peer-to-peer networks, Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, IMC'06, ACM, New York, NY, USA 2006, pp. 189–202.
[32] D. Cerri, A. Ghioni, S. Paraboschi, S. Tiraboschi, ID Mapping Attacks in P2P Networks, IEEE Global Telecommunications Conference, GLOBECOM'05, vol. 3, 2005.

Q11

Please cite this article as: D.S. Touceda, et al., Attribute-based authorization for structured Peer-to-Peer (P2P) networks, Comput. Stand. Interfaces (2015), http://dx.doi.org/10.1016/j.csi.2015.04.007