Solving identity delegation problem in the e-government environment

Sergio Sánchez García · Ana Gómez Oliva · Emilia Pérez Belleboni · Iván Pau de la Cruz

Abstract At present, many countries allow citizens or entities to interact with the government outside the telematic environment through a legal representative who is granted powers of representation. However, if the interaction takes place through the Internet, only primitive mechanisms of representation are available, and these are mainly based on non-dynamic offline processes that do not enable quick and easy identity delegation. This paper proposes a system of dynamic delegation of identity between two generic entities that can solve the problem of delegated access to the telematic services provided by public authorities. The solution herein is based on the generation of a delegation token created from a proxy certificate that allows the delegating entity to delegate identity to another on the basis of a subset of its attributes as delegator, while also establishing in the delegation token itself restrictions on the services accessible to the delegated entity and the validity period of delegation. Further, the paper presents the mechanisms needed to either revoke a delegation token or to check whether a delegation token has been revoked. Implications for theory and practice and suggestions for future research are discussed.

Keywords Digital identity · Identity delegation · Proxy certificate · X.509 · SAML

S. Sánchez García (✉) · A. Gómez Oliva · E. Pérez Belleboni · I. Pau de la Cruz
T>SIC Group, DIATEL, Polytechnic University of Madrid, Ctra. Valencia Km. 7, 28031 Madrid, Spain
e-mail: sergio@diatel.upm.es

A. Gómez Oliva
e-mail: agomez@diatel.upm.es

E. Pérez Belleboni
e-mail: belleboni@diatel.upm.es

I. Pau de la Cruz
e-mail: ipau@diatel.upm.es

1 Introduction

In the present phase of development of the information society, governments are playing a very important role. In many European countries, the public authorities are acting as the drivers of this development by implementing a number of e-government services that enable citizens to engage in different administrative processes in a manner that is both quick and effective. This development and its promotion by the authorities are a consequence of the efforts undertaken by the European Union in this field. As the European Commission sees e-government as a factor of integration and cohesion that can help achieve a single European area, a number of directives have been enacted, such as 2004/387/CE of the European Parliament and Council of 21 April 2004 on interoperable delivery of pan-European eGovernment services to public administrations, businesses and citizens [1] or action plans like the eEurope 2005 Action Plan for the delivery of e-government services [2] and the more recent i2010 Action Plan [3], which obligates administrations to gradually offer all administrative acts to citizens through the Internet.

As a first step toward the secure use of telematic services offered by public institutions, citizens must be provided with a digital identity that will identify them to Service Providers or other citizens in an unequivocal manner. For this reason, a majority of European countries—and others throughout the world—are supporting reliable electronic Identity Management Systems (eIDM) based mostly on X.509 certificates that allow citizens, businesses and government bodies—even in different member States—to identify and certify their operations in a way that is accurate, quick and simple. However, the solutions adopted to date for the telematic delivery of services by the Public Administration are insufficient, as they do not enable all possible types of user interaction

with institutions. Specifically, they do not include an important capacity that is offered in many national legal systems: namely, identity delegation, through which one citizen can authorize another to act on his or her behalf to access certain services provided by public institutions. Moreover, identity delegation is one of the main priorities of the European Union, as shown in the paper Signposts toward eGovernment 2010 [4], where it is expressly envisaged that one current restriction regarding online identity concerns the related issues of "delegation", "intermediary", and "roles" management. Also, that document states that by 2010, eID compliant systems will support mechanisms to identify and authenticate natural persons together with their varying roles (principal, delegate, intermediary, authorized agent, etc), including roles on behalf of legal persons (administrations or businesses). In like manner, the European Commission's eID Roadmap [5], according to the report on the state of pan-European initiatives for electronic identity management by the ENISA (European Network and Information Security Agency) [6] considers it to be a fundamental objective that "citizens should be able to designate persons to represent them in transactions, and the eIDM solution they use should support this".

Even though identity delegation is a process that is regulated in the legislation of many European countries and it is a priority of the European Union that is broadly accepted in society, it is not being addressed as a problem and, therefore, it is unsupported telematically. That is, where electronic identity management systems in European e-government environments support identity delegation in citizens' telematic access to services, they do so in only a rudimentary form. This problem has received little attention to date, and no clear solution has been found. The main objective of this paper is to offer a solution to the problem of identity delegation in e-government. Among the features of the proposed delegation system are the following: it is sufficiently secure to be used in this environment and it does not require the use of offline mechanisms to generate a delegation and delegation can be executed instantaneously without complex steps or the participation of a large number of entities. Further, the solution proposed can be readily integrated in present public key infrastructures; thus, given that a large number of European Public Administrations base their digital identity systems on the use of PKI and X.509 certificates, the solution can be implemented in a real environment without major changes or behavior modifications.

The following section provides the theoretical background of the proposal, including a review of the formal concepts of identity delegation and its underlying technologies. Section 3 defines a conceptual model of delegation in the e-government environment and the restrictions applicable to this environment. Section 4 presents the delegation solution and an explanation of its fulfillment of requirements. Section 5 discusses the technical viability of the solution and its applicability in a real environment. Finally, Sect. 6 presents our conclusions and future directions for research.

2 Theoretical background

2.1 Identity delegation

The concept of identity delegation is defined by the Modinis Study Team in their paper Modinis Study on Identity Management in eGovernment: Common Terminological Framework for Interoperable Electronic Identity Management [7] as "The process in which an identified entity issues a mandate to another identified entity." This definition yields the idea that the act of delegating consists of one person assigning a part of his or her rights to another person, thus allowing the latter person to act on behalf of the former person with third parties. If we focus on citizens and their interaction with Public Administrations, delegation basically involves one citizen granting another an authorization or mandate the latter can use to access, in the name of the former, services provided by Public Administrations.

According to Peeters et al. [8], there are at least three parties involved in the process of delegating electronic identity:

- *Delegator*: The person or entity who shares one or more privileges in accessing a service with another person or entity by means of what is usually called a delegation assertion/token.
- *Delegatee*: A person who receives the privileges of the delegator, namely the delegation assertion, for access to a service.
- *Service Provider*: An entity that provides services to the delegatee following presentation of the delegation assertion.

In addition to these generic entities, and depending on the delegation method used, additional entities may appear, like Identity Providers or delegation authorities, which are responsible for support activities like generating delegation assertions or validating the identities of users and entities.

With this set of basic entities, [9] presents a classification of delegation in two different models:

- *Direct delegation model*: The delegator delegates a subset of privileges to the delegatee, who uses them to access a service (see Fig. 1).
- *Indirect delegation model*: This is the same as the direct delegation model, but it is executed through a set of intermediate delegates. That is, the delegatee in turn delegates to a third party, thus yielding an indeterminate number of delegation leaps (see Fig. 2).

Fig. 1 Direct delegation model

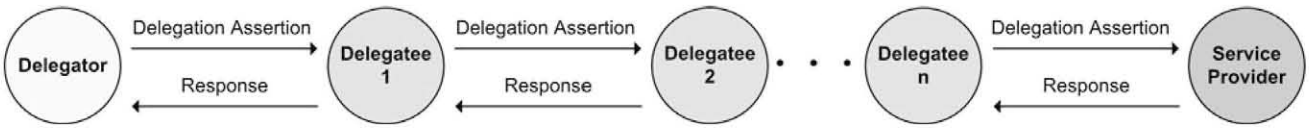

Fig. 2 Indirect delegation model

Any delegation process must take into consideration three important points.

The first is that delegation does not imply authorization. That is, even if delegation is accepted, the Service Provider is not obligated to allow the privileges requested by the delegatee, and the delegatee may not possess the privileges required for that service. The Service Provider shall always retain the decision-making power to accept or deny the request made by the delegatee.

Second, the delegation assertion must always demonstrate the consent to the delegation on the part of the delegator, as the latter may set further conditions on the act of delegation. For example, the delegator may establish a validity period for the delegation by defining a time frame or restrict the capacity for indirect delegation and, if indirect delegation is allowed, specify a limit.

Finally, the process must always seek to preserve the privacy of the delegator by not allowing the delegatee to disclose more information on the delegator than that authorized by the delegator in the delegation assertion or in the prevailing delegation policy.

2.2 Delegation token requirements

One of the key elements in any identity delegation system is what is called a delegation token, an item that allows an entity to show that it has authorization to act on behalf of another. There are multiple alternatives in handling identity information (SAML, X.509 certificates, etc.), each with its own advantages and drawbacks. However, the field of options narrows when the point is to propose a system that will enable dynamic identity delegation and support the establishment of restrictions in delegated access to services based on user attributes, that is, based on the identity attributes of the delegator.

Dynamic identity delegation entails a series of requirements that must be fulfilled for a solution to be valid.

First, users who wish to delegate must be capable of generating either on their own, or with minimal involvement of third parties, a token that enables delegation. Further, if delegation is truly meant to be dynamic, the process of generating a delegation token must be as immediate and independent as possible: it must minimize the cost in both processing capacity and necessary equipment as much as possible.

Second, the delegation token must be sufficiently secure. That is, a Service Provider receiving it must be able to unmistakably discern whether the delegation process has been executed correctly or not and retain the capacity to determine whether the token presented by a delegatee is correct, whether it has been altered since it was generated, and whether it was truly generated by the entity identified as the delegator.

If the objective is to enable dynamic and secure delegation and delegate access to services in a way that can determine, based on the identity attributes of the delegator, whether the delegator is granted access to the service or not, we will find that the token itself must carry any delegator attributes necessary, either directly or by means of a reference to them that will allow the Service Provider to retrieve them or query them.

The delegation token must be a self-contained information item that fulfills the following set of requirements:

1. Quick and easy generation, with minimal involvement of third parties.
2. Integrity. It must be possible to detect whether or not the token has been altered since it was generated.
3. Generator identification. The token must enable identification of the person who generated it: that is, the delegator.
4. User identification. The token must allow for determining whether the entity using it is the same for which it was generated, that is, the delegatee.
5. Retrieval of identity attributes. The token must enable retrieval of identity information required to access the service.
6. Service identification. The token must contain clear information on the services offered by the Service Provider for which delegation is enabled.

2.3 Review of literature on delegation

Traditionally, delegation has been studied in an environment called RBAC, Role Based Access Control [10]. In the RBAC model, assignment of permissions to users is not direct, but instead uses roles. That is, permissions are assigned to roles and users are assigned certain roles according to the tasks they are to perform, and users automatically acquire permissions associated with said roles. The separation between users, roles, and permissions allows for management of access control by systems administrators.

Even though a study that can be considered in-depth has been made on delegation in RBAC systems, with proposals such as RDM200 by Zang et al. [11], the ideas and solutions they propose are not applicable to the field of work we are addressing herein. The fundamental reason for this lies in the fact that the mechanism for managing users' identities and the environment for applying solutions are utterly different in both theoretical and practical terms.

In RBAC environments, users are assigned a limited set of roles, and services are provided to users with a determinate role. The service provision environment of public authorities must be more flexible and granular. The main reason is that services are to be provided to users who possess certain attributes. However, as a single user may have a large number of attributes, the set of combinations used as the basis for establishing restrictions for access to a service would also be quite large, giving rise to an unmanageable number of roles. In addition, although this paper discusses a specific environment that may appear closed, the solution aims to be flexible, intended for possible application in the future to broader, more global environments, as we shall see below.

With respect to delegation between persons based on user attributes and on generation of a delegation item that can grant a user or entity access to a service on behalf of another user or entity, we will find that no globally accepted solutions exist, let alone implementations of such solutions.

Even though this is an old problem in telematic security, the issue of identity delegation as it is posed in this paper has not been broadly studied until relatively recently. Nonetheless, academic literature offers a number of delegation systems conceived for different purposes and using different technologies. The following examples are notable because of their proximity to the objectives of this paper.

X.509 Proxy Certificates for Dynamic Delegation [12]. One of the first studies on identity delegation dates from 2004, by Von Welch and other authors. Even though it is not focused on identity delegation in identity management systems, an interesting solution for delegation is presented for a completely different environment, namely Grid networks.

A Delegation Framework for Federated Identity Management [13]. This delegation system was presented by Gomi et al. and it proposes a delegation framework for Web Services and federated identity management systems based on an RBAC mechanism.

Cross-Context Delegation through Identity Federation [8]. Proposal for delegation system by Roel Peeters et al. The authors describe a basic delegation scheme in a federated environment that is quite similar to the preceding one, but which is more generic and user-oriented. Delegation is achieved through a token that presents certain similarities to attribute certificates defined in RFC 3281 [14].

A Delegation Framework for Liberty [9]. In 2008, Alrodhan et al. presented a delegation framework for the Liberty technology. This proposal integrates perfectly in the proposals by Liberty [15] in identity federation. The framework described in the article uses the advantages provided by the relations of trust that exist by definition in the circles of trust as conceived by Liberty and is based on the extension of the attribute statements in SAML assertions.

Proposal of Delegation Using Electronic Certificates on Single Sign-On System with SAML-Protocol [16]. A recent proposal by Komura et al. presents a system of delegation for web applications in a scenario that assumes that the applications are accessible in a single sign-on system based on SAML specifications, where users are authenticated and authorized by the Identity Provider with X.509 certificates stored on individual smart cards.

A fundamental aspect of the identity delegation solution proposed in the present paper is its orientation toward identity delegation between persons in a dynamic and secure manner for telematic access to services. One of the missing ingredients in most solutions proposed is precisely such an orientation toward this type of delegation, as they have been conceived mainly for processes of delegation between machines or services. Another key aspect of our proposal lies in delegation and delegated access to services on the basis of a subset of delegator attributes, with the use of a delegation token that must be self-contained and secure. As discussed previously, if the need for dynamism and security in identity delegation is combined with the need to establish restrictions in delegated access to services based on the identity attributes of the delegator, a number of requirements for the delegation token arise. These requirements, considered essential by the authors, are presented at the end of Sect. 2.2. Although the majority of papers analyzed fulfill one or more of these requirements, none cover all of them, unlike the solution presented herein.

2.4 Outline of underlying technologies

Taking into consideration the foregoing, this paper presents a solution for the delegation token that has not been previously suggested for the delegation of identity between persons and is, hence, entirely novel in this field. It is based on a combination of the two technologies: X.509 proxy certificates

[17] and SAML 2.0 language (Security Assertion Markup Language) [18]. Our summary explanation for these choices is as follows:

- Proxy certificates: owing mainly to their easy integration in most identity systems presently being used in European countries—most use X.509 certificates[1] to authenticate users. Also by their potential for dynamic generation without the involvement of third parties, their ability to quickly and easily identify the generating entity and the entity for which it was generated. Finally, because they are fully verifiable from the point of view of integrity.
- SAML for the transport of user attributes, specifically SAML assertions with attribute statements: it is a relatively simple method for retrieving user attributes and because it is now the dominant trend in both the standardization and use of electronic identity management systems or eIDMs.

The use and combining of these technologies benefits from the advantages of both. First, proxy certificates, in addition to their ease of integration in PKI-based environments, offer the well-known advantages of X.509 public key certificates. For their part, SAML assertions lend the solution flexibility by allowing for a simple exchange of identity attributes between different security domains. Although the solution proposed is focused on a closed environment, we have sought to leave it as open as possible for future application in more complex environments.

The following section provides a brief summary of each of the technologies and the result of their combined use.

2.4.1 Proxy certificates

X.509 proxy certificates are defined in RFC 3820 [17], having emerged as a result of different needs that are not sufficiently met by X.509 certificates [19], the clearest case perhaps being dynamic delegation, that is, the granting of a set of privileges by one entity to another for a highly specific period of time. It is true that such delegations could be achieved with other elements from the X.509 world. For example, X.509 attribute certificates [14] can be used to delegate certain rights to other holders of X.509 certificates. However, the generation of attribute certificates implies the involvement of new entities such as attribute authorities and a large amount of processing and, hence, a large amount of time, making the process unsuitable for generating delegation in a dynamic manner. These processing and time costs could be accepted in environments that do not require dynamism in delegation, but this is not the case herein. Another problem with using attribute certificates relates to interoperability: more specifically, the need to adapt pre-existing environments for their use. As we will see, the use of proxy certificates does not imply the modification of the protocols or systems of authentication and authorization that make use of certificates, because they are common X.509 certificates with certain extensions. This supposes that the interchange and handling of this type of certificates imply minimum modifications in the present applications. X.509 certificates and proxy certificates have the same format, as both link a public key to a subject name, allowing proxy certificates to be used easily and without new implementations involving libraries and protocols initially prepared to work with X.509 certificates.

The main difference between X.509 certificates and proxy certificates lies in how they are generated. Unlike X.509 certificates, the entity that generates a proxy certificate is not a certification authority (CA), but rather an entity identified by an X.509 certificate or another proxy certificate. This immensely simplifies the process of generating certificates, as it dispenses with the process of interacting with certification authorities. Without this interaction, management time is shorter and dynamic, as intended by the solution herein.

One notable feature of proxy certificates is the fact that their public key is different than the public key of the entity generating the certificate, and they may even have different properties. For example, an entity may have a X.509 certificate that uses a 1024-bit RSA key but generate proxy certificates with 2048-bit RSA keys. A proxy certificate is an X.509 certificate with the following properties:

1. It can be signed by the end entity of an X.509 certificate and by an entity possessing another proxy certificate.
2. It can be used only to sign other proxy certificates, but never X.509 certificates for end entities.
3. It has its own public-private key pair.
4. The identity present in the proxy certificate derives from the identity of the X.509 certificate of the entity that signed it. When the proxy certificate is used for authentication, it inherits the rights of the X.509 certificate that signed it, yet subject to the restrictions specified in the proxy certificate by means of mechanisms discussed below.
5. The identity of the proxy certificate derives from the identity of the X.509 certificate used to generate it and is unique at the generator level.
6. Proxy certificates contain an extension that identify them as such and allow for establishing usage policies. This extension, along with other fields and X.509 extensions, allows for it to be properly validated.

[1] Throughout this paper "X.509 certificate" refers to X.509 public key certificates that demonstrate user identity according to RFC 5280 [20], "X.509 proxy certificate" refers to certificates defined in RFC 3820 [17] and "X.509 public key certificate" includes both of them.

```
id-pkix OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)
      dod(6) internet(1) security(5) mechanisms(5) pkix(7) }
  id-pe OBJECT IDENTIFIER ::= { id-pkix 1 }
  id-pe-proxyCertInfo OBJECT IDENTIFIER ::= { id-pe 14 }
  ProxyCertInfo ::= SEQUENCE {
     pCPathLenConstraint   INTEGER (0..MAX) OPTIONAL,
     proxyPolicy           ProxyPolicy }
  ProxyPolicy ::= SEQUENCE {
     policyLanguage        OBJECT IDENTIFIER,
     policy                OCTET STRING OPTIONAL }
```

Fig. 3 ASN.1 definition of the PCI extension (Proxy Certificate Information extension)

2.4.1.1 Proxy Certificate Information extension X.509 certificates and proxy certificates have the same format but, in accordance with RFC 3820, all proxy certificates must contain a critical extension called Proxy Certificate Information extension (PCI), whose ASN.1 definition is shown in Fig. 3. This extension has different objectives or functionalities. First, it serves to identify the certificate as a proxy certificate. Second, it enables the certificate generator to express its desires with regard to the delegation of rights and to limit the number of proxy certificates that can be generated from it.

To support the preferences of the generating entity with respect to the delegation, the PCI extension has a framework for transporting the delegation policies expressed in any policy language, with the sole restriction being that both parties must be able to interpret said language and, therefore, the policy defined. To achieve this, the `proxyPolicy` field in the PCI extension is used. As we can see in the ASN.1 definition, the `proxyPolicy` field consists, in turn, of another two fields: `policyLanguage` and the `policy`. The `policyLanguage` field indicates the language in which the policy is expressed and `policy` is an optional field that contains a statement of the policy in the language indicated by the former. RFC 3820 on proxy certificates defines two values for the `policyLanguage` field that are of particular importance, as they must be understood by all implementations of proxy certificates: `id-ppl-inheritAll` (Proxying) and `id-ppl-independent` (Independent).

The value of `id-ppl-inheritAll` (Proxying) indicates that the generator of the proxy certificate delegates all privileges to the holder of the proxy certificate. The value of `id-ppl-independent` (Independent) indicates the contrary: that is, the generator assigns no privileges to the proxy certificate, meaning that the proxy certificate can only provide its holder a unique identifier that can be used along with other solutions to grant privileges.

In these two cases, the `policy` field must be empty, as the delegation policy is implicit.

When other values are used for the `policyLanguage` field, the policy field will contain the policy that establishes the use restrictions of the proxy certificate, except in cases in which the policy is implicit in the value itself, as in the two preceding cases.

To limit the number of proxy certificates that can be generated from one proxy certificate, the `pCPathLenConstraint` field is used. This field uses an integer to establish a restriction on the maximum length of the proxy certification path that can be generated from the proxy certificate. If the value of `pCPathLenConstraint` is zero, this indicates that the proxy certificate cannot be used to generate other proxy certificates. If the value is n, where n is an integer within the permitted range, this indicates that the proxy certificate can be used to generate n proxy certificates under it: that is, in which the initial proxy certificate is the root (see Fig. 4). It should be recalled that this field may not be shown, in which case it is considered an infinite value.

2.4.1.2 Process of generating a proxy certificate for delegation The process of generating proxy certificates as part of the delegation between a delegator and a delegatee implies communication between them through a secure channel and the series of steps shown below (Fig. 5):

1. The delegatee generates a key pair, one public and the other private, and forms a request for a proxy certificate.
2. The delegatee sends the request generated through an authenticated channel with an integrity guarantee.
3. The delegator checks whether the request is correct, and if it is, it generates proxy certificate. The certificate will be signed with either the private key of the delegator or the private key of another proxy certificate.
4. The delegator sends the delegatee the proxy certificate generated through an authenticated channel with an integrity guarantee.

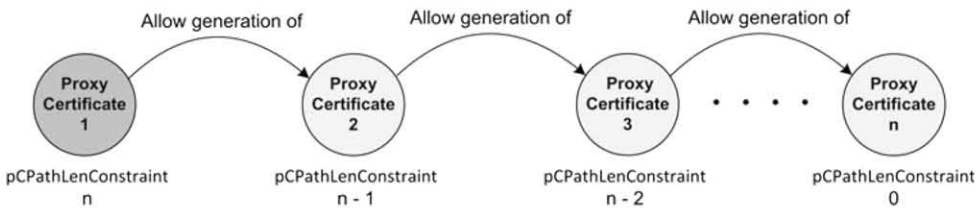

Fig. 4 Chain of proxy certificates rooted in a proxy certificate with *pCPathLenConstraint* at n

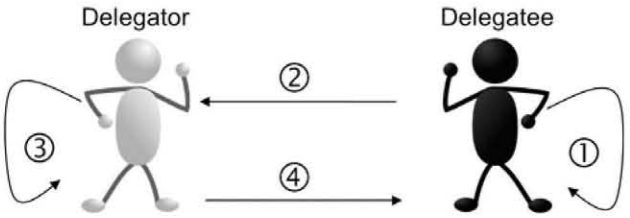

Fig. 5 X.509 proxy certificate generation process for delegation

The process of generating proxy certificates is much quicker and easier than for X.509 certificates; its main advantage being that it does not require the intervention of a CA.

2.4.1.3 Validating and revoking proxy certificates RFC 5280 [20] defines the idea of validating the certification path as a process of finding a trusted path from an end entity's X.509 certificate to the root certificate of a certification authority. In a manner similar, proxy certificates, verification of the certification path of a specific proxy certificate consists of finding a trusted path through a chain of proxy certificates to the X.509 certificate of the entity that generated the first proxy certificate in the chain.

Below is a possible verification algorithm of the certification path as detailed in RFC 3820. It is important to recall that the process of validating proxy certificates presented involves validating the time and date upon validation, but it would be possible to envisage validation of a proxy certificate for a specific moment in the past, given that no mechanisms exist to validate a proxy certificate for a moment outside its validity period.

A valid certification path always begins in a X.509 certificate for an end entity or end entity certificate (EEC) correctly validated according to the procedures indicated in RFC 5280 [20]. Validation of a proxy certificate requires a distinguished name and the public key of the EEC that generated it, and the certification path, that is, a chain of n certificates, must fulfill the following conditions:

1. For any `x` in the set between `1` and `n-1`, the value of the subject field of the `x` certificate must match the value of the issuer field of the proxy certificate where `x + 1` is a legal distinguished name subject because it was generated by certificate `x`.
2. Certificate `1` is a valid proxy certificate generated by the end entity certificate (EEC) provided as an input to the process of validating the proxy certification path.
3. The certificate `n` is not the proxy certificate to be validated.
4. For any `x` in the range between `1` and `n`, the certificate was valid at the time in question.
5. For certificates in the path with a `pCPathLenConstraint` field, the number of certificates following it in the certification path does not exceed the length specified in said field.

One aspect that is notable for its importance in the certificate environment and, particularly when used in identity delegation, is revocation. As the RFC on proxy certificates [17] does not define a mechanism for revocation, no method is presently implemented for the revocation of these certificates. Traditionally, proxy certificates used for delegation have been used in environments where delegation periods are quite short, so proxy certificates' lifetimes are very short. This sharply limits the danger that a proxy certificate might be compromised and misused and, thus, reduces the pressure to find a solution to this problem. However, in delegation of identity between citizens for access to services provided by Public Administrations, delegation periods can be long, and the risk that a certificate may be compromised is very high. Hence, given that proxy certificates can be identified uniquely if we associate them with their generator (the issuer-serial number binomial is unique), a mechanism will later be defined for revoking and checking the revocation status of a proxy certificate with the use of mechanisms similar to CRLs (Certificate Revocation Lists) [20] or OCSP (Online Certificate Status Protocol) [21] used in the X.509 certificates.

2.4.2 SAML

This section will provide a brief introduction to the SAML 2.0 protocol [22,23]. This is a standard that has been developed and maintained by the Security Services Technical Committee [24] of the Organization for the Advancement of Structured Information Standards (OASIS) [25]. In summary, it can be said that the SAML defines an XML-based platform for the description or exchange of security information; that is, authentication and authorization information exchanged by different security domains over the Internet. For such an exchange to occur, the information is packaged in messages of a pre-established structure called assertions (SAML Assertions). The standard defines a precise syntax and a set of rules for creating, requesting, communicating and using SAML assertions: that is, it defines what data is to be exchanged and how that data is structured in messages, while offering a standard single sign-on solution.

2.4.2.1 SAML Assertions Assertions are SAML components that contain statements about an entity, with a structure and content defined as an XML scheme of SAML assertions. Assertions with attribute statements are of particular interest for us, as they enable creation of a delegation token along with a proxy certificate.

As we have seen, and in accordance with [18], we might say that SAML allows an entity to assert, through assertions,

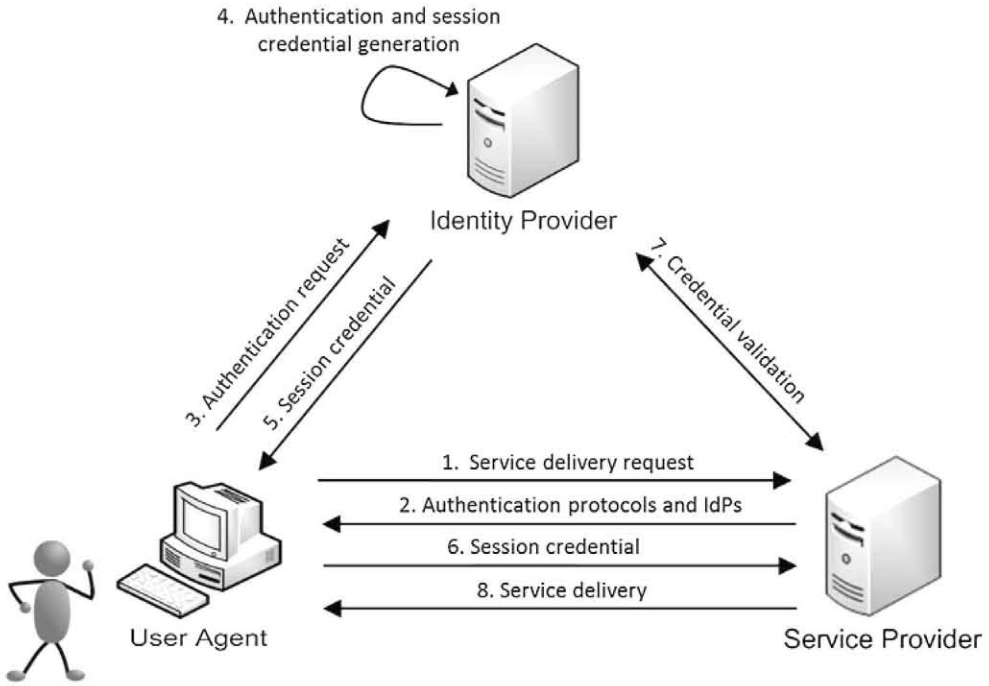

Fig. 8 Interactions in abstract model of identity management system

3.1 Participating entities

Generically, we might say that four entities or agents are involved in a scenario of identity federation:[2]

- *User* (U): A person who seeks to access the services offered by a provider and whose identity is federated. It is assumed that the user has some form of credential that is accepted by the Identity Provider and, optionally, by the Service Provider.
- *User Agent* (UA): An agent employed by the user to communicate or interact with the Service Provider and the Identity Provider.
- *Service Provider* (SP): Offers a service to the user. Before access is allowed to the service, the user must be authenticated. If the Service Provider accepts the user's original credentials, it may carry out the authentication directly; otherwise, authentication is delegated to an Identity Provider that does possess this capacity and, further, is an entity trusted by the Service Provider.
- *Identity Provider* (IdP): The entity responsible for validating the user's credentials and delivering session credentials that can be validated by the Service Provider when authenticating a user prior to providing a service.

[2] This generic model of identity management does not include PKI because it does not participate in the token generation process. Nevertheless, as explained below, it plays a key role in validating signatures, certificates and so on.

3.2 Interaction for service access and delivery

Figure 8 shows the interactions for access to a service by a user according to the generic model and their order. Then, each interaction is described in detail.

1. Through the UA, the user requests from delivery of a service from the SP.
2. The SP answers the service request by indicating to the UA what authentication protocols it supports and what IdPs it trusts.
3. The UA establishes communication with an IdP and presents the user's credentials with a view to achieving authentication and obtaining a session credential that will be valid for accessing the service provided by the SP.
4. The IdP authenticates the user based on the credentials and, if everything is in order, issues a session credential.
5. The IdP sends the session credential to the UA.
6. The UA sends the session credential to the SP.
7. The SP validates the credential received from the UA and verifies the user's access rights.
8. If everything is in order, the SP delivers the service requested to the user.

Further, the above process may have a series of steps in which the user selects a digital identity. In this case, the user, the user agent, and the Service Provider interact for the purpose of selecting the identity to be used and, hence, the Identity

Provider to be contacted in step 3. It should be recalled that the interactions depicted herein reflect a high-level, generic model that does not address the capabilities or mechanisms necessary to carry them out. Hence, for example, no indication is given of the method for redirections, which may be executed similarly to Single Sign-On redirection profiles in SAML.

3.3 Applicable domain and restrictions

Before presenting the proposed delegation solution, it would be useful to clearly define what domain is applicable to the different solutions and what set of restrictions is to be applied in accordance with the specific conditions of the domain.

The scenario for application in which work has been conducted and for which the solution has been conceived is Public Administration, with delivery of services to citizens in environments involving use of identity management systems with a model that is similar to that presented previously.

Three basic entities are involved in these scenarios: the user, the Identity Provider (IdP), and the Service Provider (SP). If we analyze these entities in the environment of Public Administrations, we will find that, unlike the generic model, both the Identity Provider and the Service Provider are controlled by the same organization, namely the state, which implies a series of aspects that should be considered in assessing the solution. Because it is a "closed" environment in which only the Public Administration acts as the Service Provider and Identity Provider, it can be assumed that relations of trust exist between the different IdPs and SPs that might participate in a transaction between citizens and Public Administration: that is, we can presume that all entities in the Public Administration involved in a transaction are trusted third parties (TTPs) to each other. In spite of the closed nature of the environment, the Public Administration itself contains multiple configurations with multiple security domains, even in a single country. Therefore, the solution proposed addresses this problem and strives to be flexible in order to adapt to possible existing configurations.

One may pre-suppose that the behavior of entities will be proper, in the sense that their treatment of information on citizens is correct and as expected according to their status and functionality, as they are considered to be properly audited and monitored. Moreover, they are, by definition, reliable sources of information, especially the IdPs, which have the basic role of authenticating citizens and providing reliable information on their identity.

Another of the assumptions one can make is that mechanisms will exist so that citizens need not concern themselves about issues such as usurpation in either the IdP or the SP: that is, it is assumed that both Identity Providers and Service Providers reside in environments that are free of problems such as phishing. Finally, it is assumed that citizens participating in the transactions will use an electronic identity based on X.509 certificates and on an associated PKI (Public Key Infraestructure) that is similar to that implemented today in many member States of the European Union. Figure 9 contains a graphic depiction of the proposed application scenario. All entities in the Public Administration would have a relationship of trust between them and shall be auditable: they can be checked at any time to ensure that their behavior is as expected. Moreover, the Identity Provider constitutes a reliable source of data. That is, there are guarantees that the information provided is authentic and valid at the moment of the query. Users access the Identity Provider and the Service Provider through secure and authenticated connections to ensure that exchanges of information occur between the right entities. Both users and entities in the Public Administration have X.509 certificates to attest to their electronic identity. They also have a set of capacities necessary for interacting with a PKI to engage in all processes associated with the query, revocation, and validation of X.509 certificates.

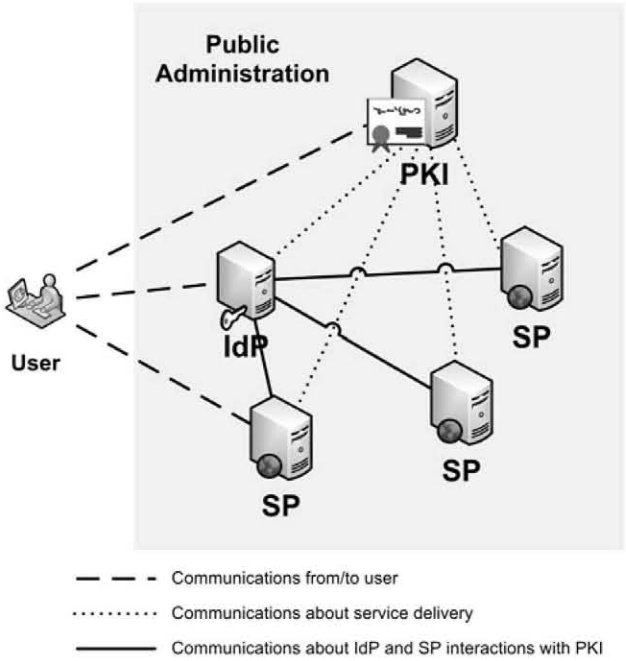

Fig. 9 Proposed application scenario

4 Delegation solution

4.1 Working methodology

The methodology used in undertaking the work presented herein is based on the successive approximations to a solutions that fulfills the requirements of the application environment, as set forth in Sect. 2.2. Hence, as we shall see, a combination of the two basic technologies discussed above is the starting point: proxy certificates that enable delegation

and their integration with SAML attribute assertions for the correct identification and transfer of attributes in the process. Additional elements are added, depending on the specific needs of the application environment.

4.2 Preliminary notes

As mentioned, the integration of SAML assertions in X.509 certificates as proposed in the GridShib constitutes the basis of the identity delegation token proposed in this paper, and it will be used in an environment of identity delegation between citizens or entities for interaction with Service Providers in Public Administrations. This application environment poses a series of requirements that prevents the system presented in the Gridshib project from being directly applicable, as we will see. Nonetheless, it is valid as an initial basis for development and this is how it has been used.

The first problem of the token as presented is that it is conceived for use in Grid environments: that is, for delegation of identity between processes with very specific and clearly defined tasks and which, hence, need not be specified in the delegation token because they are known a priori. The functionality of the process receiving the delegation is limited by the code itself. For example, as an FTP process can perform only those tasks for which it has been encoded from among the tasks specified in the FTP standard, the limitation is established a priori.

Delegation of identity in Public Administration environments as addressed in this paper involves delegation between citizens or between entities and citizens. That is, it is delegation between natural persons or between legal persons and natural persons, and this entails a series of conditions that must be taken into account. First, delegation between processes in a Grid environment usually has a quite limited duration—typically minutes or a few hours—with a few highly specific tasks. This explains why the solution adopted by Gridshib project does not include the possibility of revoking the delegation, namely the delegation token. This would not be viable in the environment herein. Delegation occurs between persons and for periods of time that can range from a few hours to several days or years. In periods this long, with the involvement of persons, countless situations may arise that might require revocation of a delegation. A solution based on such short delegation periods is not valid in such environments because it would lead to continuous revocation of delegation for each interaction, with the resulting drawbacks for users. Due to poor planning, a delegation token might expire before the end of the interaction with which it is associated. This is an obvious problem and, hence, a system of delegation is needed that supports revocation.

Finally, the solution proposed by GridShib involves use of SAML v1.1. This version is now obsolete, having been updated to SAML 2.0. The delegation token proposed herein uses the latest version. Therefore, from this point onwards, unless otherwise specified, all references to SAML in this paper can be understood as corresponding to SAML 2.0.

4.3 Problems found and our contributions

4.3.1 Services identification

One of the key aspects in access to online services, whether these are provided by the Public Administrations or private companies, is the correct identification of said services. While multiple solutions exist for univocally identifying online objects and services, the most widespread of these is the one specified in RFC 3986 [28], which defines URIs (Uniform Resource Identifiers). The W3C (World Wide Web Consortium) specifies in its recommendation on addressing of these services, called Web Services Addressing 1.0—Core [29], that the element for univocally identifying a service is the IRI (Internationalized Resource Identifier). IRIs are defined in RFC 3987 [30], where it is clearly specified that they are a complement to URIs. An IRI is a sequence of characters from a universal character set called Unicode/ISO 10646 that can univocally identify an object or resource on the Internet, thus avoiding certain limitations in URIs with regard to the characters that can be used and making it more international than URI. Nonetheless, IRIs are designed to be compatible with URIs. Even though they are more mainstream, use of IRIs is not as widespread as URIs, mainly because certain commonly-used protocols, like HTTP, do not support the use of IRIs, though they do support URIs. In view of the foregoing and, although it is not in widespread use, we propose to use IRIs as the mechanism for identifying services in the identity delegation token. The reasons for this choice are as follows:

- The mechanism defined should be as standard as possible, and IRIs are defined by the W3C in their recommendation on addressing of web services [29].
- IRIs are an extension of URIs and, hence, the definitions for IRIS are applicable to URIs, providing coverage to those used at present.
- The mechanism defined should be as flexible as possible and IRIs encompass a much wider addressing range than URIs.

IRIs correspond to a chain of characters that can univocally identify a resource. They also have a hirerarchical structure that provides a clear idea of the logical location of a service on a provider. Service providers generally make a logical grouping of services according to different criteria that usually yields a hierarchical distribution in the form of a tree, resulting in a structure similar to that shown in Fig. 10.

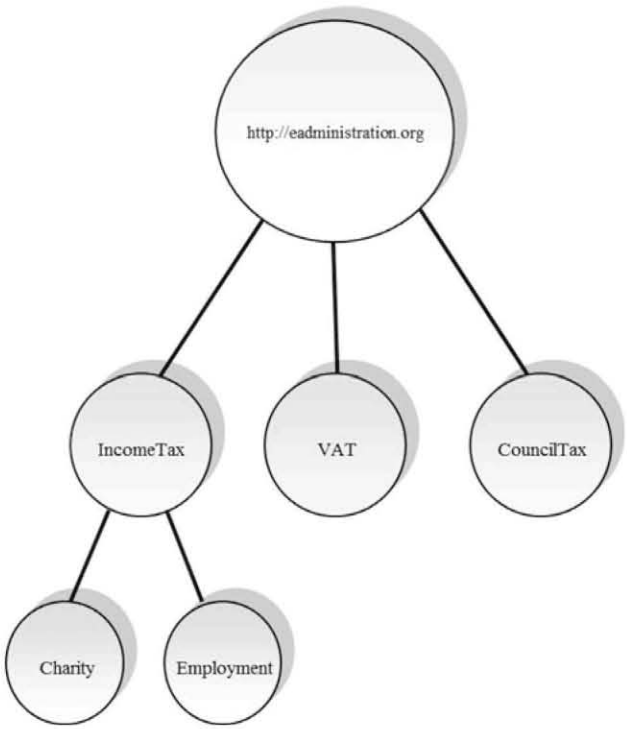

Fig. 10 Possible hierarchical structure of provider services

4.3.2 Restricted token use in the services tree

In delegation of identity, one important aspect of accessing a provider's services is the capability to clearly and concisely identify service(s) to which access is delegated. As we have seen, this is solved with the use of IRIs.[3]

A task that is not as simple, and for which no solution exists to date, is specification in a delegation of not only one service but a specific set of services throughout the logical distribution of the provider's services. The ultimate goal is to make use of IRIs' hierarchical structure to enable the delegator to specify in the identity delegation token itself which services will be subject to delegation and which will not: that is, to establish a logical pruning of the service tree offered by the provider. Generically, any one of the following three cases is possible:

a Delegation applies to only one of the services offered by the provider. For example, the one identified by the IRI http://eadministration.org/IncomeTax/Charity in Fig. 10.

b Delegation applies to a set of services dependent on the same logical root of a provider: that is, a branch of the logical tree. For example, the branch identified by IRI http://eadministration.org/IncomeTax/ in Fig. 10.

c Delegation applies to a set of limited subset out of the total services offered by the provider. For example, those identified by the IRIs http://eadministration.org/VAT and all those under the one identified by http://eadministration.org/IncomeTax, except for employment, which is identified by IRI http://eadministration.org/IncomeTax/Employment in Fig. 10.

A solution to this problem requires the capacity to add to the delegation token a mechanism that can clearly and optimally define which services can be accessed by the delegate and which cannot. At first glance, it would seem that the problem is not quite so serious, as it is generally assumed that the solution consists of including in the delegation token a list of all the service IRIs to which access is granted. This solution is both simple and valid, but it is limited to a simple operating environment in which the number of services per provider is relatively small; otherwise, the service list and, hence, the delegation token, could become unmanageable. Given that the identity delegation token proposed herein is composed of a proxy certificate with a set of non-critical extensions, the solution must be capable of fully integrating into it. This is why we have decided to define a new non-critical extension called serviceIRIConstraints to be included in the delegation token, with the capacity to optimally specify the set of services within the logical structure of a provider to which delegated access is granted.

An extension of the X.509 v3 certificate consists of a unique OID, a Boolean value that indicates whether or not it is critical and a chain of octets corresponding to the DER encoding of the extension value. Obviously, because this is a proposal, we cannot now specify a unique OID for the extension. The proposed extension is not critical, as it is designed to be used in delegation environments along with proxy certificates that are designed to be easy to use, with no new implementations or with minimal updates by libraries and protocols initially prepared to work only with X.509 certificates. If the extension were critical, we would be forcing all libraries and protocols to be able to interpret and manage it and, therefore, the identity delegation token would not be as flexible as it is meant to be.

Clearly, in an environment where delegation is based on the model proposed in this paper, with use of the identity delegation token as described, all entities that have to deal with the token must be able to understand and manage the extension serviceIRIConstraints. Figure 11 presents the ASN.1 definition of the extension. Provisionally, and pending study and approval by the standards authorities and assignment of an official OID, it has been assigned the object ID 99. Basically,

[3] A priori, the services offered by public administrations do not require different permissions according to types of citizens, but rather enable all citizens to carry out a limited a specific set of operations. If a citizen decides to delegate access to a service, the delegatee will have the same privileges as the delegator. Hence, this paper does not seek to define with greater granularity the privileges or permissions of delegatees in access to services. Nevertheless, the authors believe that this possibility could be of interest in lending the system greater flexibility and it will be included in future work.

```
id-ce- serviceIRIConstraints OBJECT IDENTIFIER ::= { id-ce 99 }

serviceIRIConstraints ::= SEQUENCE {
      permittedSubtrees  [0] ServiceSubtrees OPTIONAL,
      excludedSubtrees   [1] ServiceSubtrees OPTIONAL }

ServiceSubtrees ::= SEQUENCE SIZE (1..MAX) OF ServiceSubtree

ServiceSubtree ::= SEQUENCE {
      base                internationalizedResourceIdentifier,
      minimum    [0] BaseDistance DEFAULT 0,
      maximum    [1] BaseDistance OPTIONAL }

internationalizedResourceIdentifier ::= UniversalString
          BaseDistance ::= INTEGER (0..MAX)
```

Fig. 11 ASN.1 definition of proposed non-critical extension *serviceIRIConstraints*

the extension allows for establishing limitations in the delegated access to services: that is, it can specify which services can be used with the identity delegation token and which cannot. It indicates these services through its IRI, defining two lists: the permitted IRIs and the excluded IRIs, and it permits logical grouping through root nodes. We shall now discuss in detail the extension's fields and their possible values.

As shown in Fig. 11, the proposed extension consists of two fields, `permittedSubtrees` and `excludedSubtrees`, both of which are optional. The former of these specifies the IRI(s) of the services for which the delegator allows the delegatee to use the delegation token in which the extension is embedded. The latter indicates the opposite, namely the IRI(s) for which the delegator does not allow the delegatee to access and use the delegation token in which the extension is embedded. In both cases, the data type is `ServiceSubtrees`, which, as seen in Fig. 11, is defined as a sequence of `ServiceSubtree` elements in which at least one element and a maximum of `MAX` elements must appear. `ServiceSubtree` is in turn composed of a sequence of three elements: `base`, `minimum`, and `maximum`. The first two are mandatory, and the third is optional. Element `base` is defined as an element of the same type as that used in defining IRIs under RFC 3987, so it may contain any IRI.

The fields `minimum` and `maximum` are of type `BaseDistance`, that is, are elements defined as an integer between `0` and `MAX`. By default, field `minimum`, which is mandatory, takes the value zero, while `maximum` is optional. The field is defined in exactly the same way as the previous one but it is optional: that is, it need not appear and therefore has no default value.

As seen from the definition of its components, a `ServiceSubtree` element can define the base IRI of a Service Provider and, using that base, the minimum and maximum logical depth at which services can be found when permitting or excluding their use by the delegatee. That is, each `ServiceSubtree` element can clearly define a service or set of services within a provider's logical organization of services.

We have noted that the extension is composed of two elements: `permittedSubtrees` and `excludedSubtrees`.

Both are `ServiceSubtrees` and, hence, are sequences of elements that concisely define a service or set of services within a provider.

The element `permittedSubtrees` is used to specify the services to which the delegatee can access with the identity delegation token. On the other hand, the `excludedSubtrees` element is used to specify the services, on a given Service Provider, to which access with the identity delegation token by the delegatee is specifically denied. The combination of `permittedSubtrees` and `excludedSubtrees` enables the extension serviceIRIConstraints to clearly define the services accessible by the delegatee with the delegation token containing the extension.

With a view to further clarifying the use of this extension, what follows is an example of the values that would be used in the fields of a Service Provider with a hierarchical service structure like that shown in Fig. 10. The example corresponds to case c in Sect. 4.3.2 in which delegation is applied to a series of dispersed services out of the total offered by the provider. For example, those identified by the IRIs http://eadministration.org/VAT and all those under that identified by http://eadministration.org/IncomeTax, except employment, which is identified by the IRI http://eadministration.org/IncomeTax/Employment in Fig. 10. This case presents as a special feature that allows for two different subsets within the service logical structure and, second, it excludes a specific service from one of the permitted subsets. This means that the prohibition must be explicitly reflected in `excludedSubtrees` in the list of excluded IRIs.

```
case-c serviceIRIConstraints ::= {
    permittedSubtrees  permitted-c,
    excludedSubtrees   excluded-c
    }

permitted-c ServiceSubtrees ::= {
    ServiceSubtree permitted-c1, permitted-c2
    }

excluded-c ServiceSubtrees ::= {
    ServiceSubtree excluded-c1
    }

permitted-c1 ServiceSubtree ::={
    base        http://eadministration.org/VAT,
    minimum     0,
    maximum     0
    }

permitted-c2 ServiceSubtree ::={
    base        http://eadministration.org/IncomeTax/,
    minimum     0
    }

excluded-c1 ServiceSubtree ::={
    base        http://eadministration.org/IncomeTax/
    Employment,
    minimum     0,
    maximum     0
    }
```

4.3.3 *Fulfillment of requirements of the final delegation token*

The preceding sections presented different ideas and possibilities for fulfilling the requirements in Sect. 2.2. The result is that the correct combination of all these pieces of information gives rise to a proposed delegation token offered as a solution and graphically depicted in Fig. 12.

The basis of the delegation token is a X.509 certificate with the extension PCI: that is, a proxy certificate in which the field `policyLanguage` takes the value `id-ppl-independent` (Independent), according to which the delegator grants none of its privileges to the delegatee. Hence, the delegatee receives from the delegator through an assertion only those privileges deriving from a set of SAML attributes contained in the token. Another important field in the proxy certificate to which a default value must be assigned is `pCPathLenConstraint`. For the delegation token proposed herein, it must have the value zero by default: that is, by default, only direct delegation is allowed; thus, the delegatee cannot use the delegation token to delegate to a third party. The reason for this constraint is that the environment of Public Administrations, for which this solution is conceived, normally allows only one delegation leap. In addition, the more limited the delegation—here, the number of delegators and delegatees involved—it is more controllable from the perspective of security and auditing, where the latter is highly important in preventing possible misuse of delegation. Nonetheless, the fact that only direct delegation is allowed by default does not mean that—in certain environments and under certain conditions—that indirect delegation, with a larger number of leaps in the delegation, is permitted. Besides extension PCI, obligatory in the Proxy Certificates, proposed delegation token includes other two extensions. The first one is defined by Globus for the inclusion of SAML assertions but modified by authors for use SAML v2 instead of v1.1. The second one called serviceIRIConstraints is suggested in this work to determine what services can be used with the delegation token.

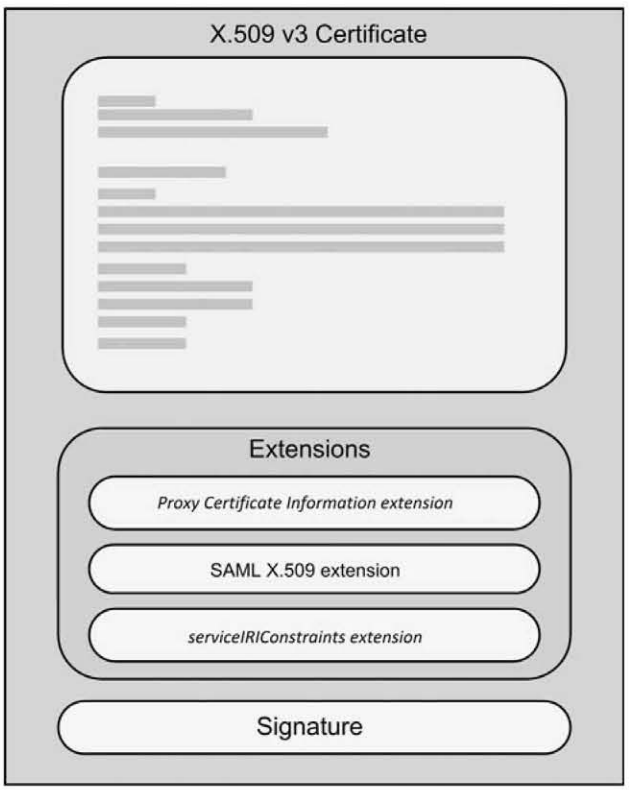

Fig. 12 Graphic depiction of the proposed delegation token

It is worth emphasizing that all the information necessary for access control and delegation is included in the token that is controlled by the delegator who decides what of their identity attributes delegates and what services of Service Provider grants delegated access, not leaving any aspect of the delegation at the mercy of the delegatee. The delegator decides what attributes to include in the token and, on the basis of these attributes, certain services are provided or not provided in the Service Provider. Further, the delegator also specifies which services available in a Service Provider are accessible with the token generated.

It is also important to emphasize that, even though they are closely linked, delegation of identity and control of access to a service are two different things. The process of delegation involves issuance of a valid delegation token, without which the delegatee cannot access services. For its part, control of access to a service in a delegation environment involves the existence of a valid delegation token but, in turn, a valid delegation token that has been correctly formed does not imply immediate access to services by the delegatee. In addition to a delegation token that is valid, a series of access requirements must be fulfilled that may not be satisfied by the attributes and restrictions established by the delegator in the delegation token.

If we analyze the proposed delegation token in terms of fulfillment of the six requirements specified in Sect. 2.2, we will see that it satisfies all of them. Generation of a proxy certificate is quick and easy and, initially, it can be transparent to the user. Token integrity is fulfilled by the digital signature that must be included in the proxy certificate. The signature also fulfills the need to identify the issuer of the proxy certificate, that is, the delegator. The requirement of identifying the user, that is, the delegatee, is simple if we recall that a proxy certificate certifies a public key for which the user must possess the private key. A simple challenge and response mechanism allows for verifying whether the delegation token has been issued for the user actually using it. The requirements of retrieval of the identity and service identification attributes for which delegation is enabled are satisfied, as we have seen, by extensions.

4.4 Revoking a delegation token

This section presents a solution to the particularly important problem of revoking delegation tokens.

The validity period of the delegation token is set during the process of its issuance and is reflected in the token. Because the token is based on a proxy certificate, fields related to the validity period of the certificate are used to set the validity period of the token, namely `Validity` (`Not Before` and `Not After`). By default, and as a rule in the delegation process, the validity period should match the estimated time of use of the token as closely as possible by seeking to reduce as much as possible the period in which the token can be compromised and thus minimize possible security problems arising from improper or unauthorized use of the token. Nonetheless, even if the validity period of the identity delegation token is reduced to the minimum essential level, the possibility that the token may be compromised must be addressed. A proxy certificate is, in general, less secure than a X.509 certificate. The main reason lies in the fact that the private key associated with a proxy certificate is not protected with the same diligence and rigor as a private key associated to a X.509 certificate. Hence, and given that the proposed delegation token is based on the use of a proxy certificate, a revocation mechanism must be enabled. That is, there must be a mechanism to enable either revocation of a delegation and a public statement of its invalidity or to check whether a delegation token has been declared invalid or not.

The problem of revocation, which has clearly been solved in X.509 certificates with the mechanisms of CRLs [20] or the OCSP protocol [21], does not have a standard solution in the case of proxy certificates, as the RFC 3820 itself [17] has indicated that no mechanism has been defined, as of yet, for revoking such certificates. Given that the delegation token proposed in this paper is basically a proxy certificate, there is no directly applicable revocation mechanism. We will now present a solution to this problem: namely, a mechanism for revoking the proposed delegation token.

A series of problems arise in establishing a mechanism for revoking proxy certificates. Unlike X.509 certificates, proxy certificates are not issued by a limited number of certification authorities; the entities that issue proxy certificates need not be online and, a priori, proxy certificates issued by an entity are unique from the point of view of identification within the entity, but they need not be outside it.

The fact that the uniqueness in identifying proxy certificates is relative and that, theoretically, there are as many certification authorities (CAs) as end entities that wish to delegate—in proxy certificates, the end entity signing the proxy certificate issued acts as the CA—means that the use of mechanisms based on a limited number of hierarchical CAs like CRLs or OCSPs, is unfeasible, as such mechanisms are not designed for environments in which the number of CAs may be massive and where, in theory, they need not be online to provide revocation lists. The same conclusions apply to the OCSP protocol, which is also designed for a limited number of hierarchical CAs. Hence, the problem of revoking proxy certificates and, thus, the proposed delegation tokens, arise from two issues: the lack of univocal identification and the lack of a centralized and permanently online entity that might enable requests to revoke a token and provide answers to possible queries on revocation status.

We will now present proposed solutions to these problems.

4.4.1 Univocal delegation token identification

A solution for revocation of delegation tokens requires, as a first step, univocal identification of tokens. In identification of a proxy certificate—and thus a delegation token—there are a number of fields whose values are set in RFC 3820, as follows:

- *Issuer*: A proxy certificate can be generated through use of either an X.509 certificate or another proxy certificate. The `issuer` field in the proxy certificate must contain the same value as the `subject` field of the certificate used to generate it.
- *Issuer alternative name*: Use of the extension `issuerAltName` in the proxy certificate is prohibited.
- *Serial number*: As the serial number of a proxy certificate must be unique among all the proxy certificates issued by the same issuer, the issuance policy must ensure a minimal likelihood of collision. Possible assignment policies involve consecutive whole numbers or serial numbers generated from the public key hash of the proxy certificate.
- *Subject*: Shown in the `subject` field of the proxy certificate. It is formed by adding to the `subject` field of the certificate used to issue proxy certificate a `Common Name` component. The value of `Common Name` must be unique for every bearer of a proxy certificate from the same proxy certificate issuer. If an issuer issues two proxy certificates to the same bearer, it can choose to use the same `Common Name` for both. As in the previous case, the issuer of the proxy certificate can choose a system that will ensure a high probability of uniqueness, with use of the same value as for the serial number field. The result of this approach is that all subject names of proxy certificates derive from the name of the end entity certificate that issued it and which are unique for every bearer.
- *Subject alternative name*: Use of the extension `subjectAltName` on the proxy certificate is prohibited.

As we can see, there is not field which must necessarily be unique at a global level. The aim is to guarantee uniqueness within a given issuer of proxy certificates, while providing recommendations to achieve a certain degree of uniqueness, though without establishing a single criterion as a standard for preventing collisions at a global level. Of the fields discussed above, `subject` on proxy certificates comes closest to achieving an identification system of the kind desired. If the value of `Common Name` is chosen correctly, we can achieve an identification system that is unique and which associates the proxy certificate to its issuer. One simple way of yielding a `Common Name` with a high degree of uniqueness is proposed in the standard itself, which involves matching the value with the result of applying an SHA hash algorithm to the proxy certificate's public key.

This paper proposes as solution the following steps for issuing a delegation token: the `subject` field of the proxy certificate, which acts as base, must be formed by adding the `subject` of the X.509 certificate of the entity issuing the delegation token; that is, the delegator, and a unique element resulting from the application of the SHA-256 hash function [31] to the public key of the delegation token. This solution not only verifies the criteria set forth in RFC 3820 but also achieves a high degree of uniqueness in the identification of delegation tokens, as the probability of a collision in identifiers issued in this way is quite small if we assume that different proxy certificates in a single generator have different public keys, which is a fair assumption.

4.4.2 Delegation token revocation authority

Having solved the problem of univocal identification of delegation tokens, we must address the lack of a centralized, permanently online entity that could handle revocations and responses to queries about the revocation status of tokens. We propose a solution consisting of the creation of an entity called the Delegation Token Revocation Authority (DTRA), which will manage, in a centralized manner, a register of delegation tokens that have been revoked. The fundamental principle is the same as that of CAs with their revocation lists, but adapted to the particular conditions of identity delegation, namely:

- *No single, unique CA*: As the number of CAs is, theoretically, unlimited, a revocation system based on a query of the CA issuing the certificate does not apply. Hence, the Delegation Token Revocation Authority will be a centralized entity managing a register of delegation tokens that have been revoked for its application environment.
- *Certificate identification*: In PKIs, certificates are identified univocally by their serial number and CA. Thus, when revocation queries are made in relation to a CA, only the serial number of the certificate is required to check its status. When identifying delegation tokens, the Revocation Authority must use the identification solution discussed in the preceding section to ensure that each delegation token is identified in a univocal manner. This means that the format of the lists of revoked delegation tokens differs slightly from that of revoked X.509 certificates, as identification by the `serial` associated to a CA must be replaced by the `subject` in the delegation token.

When checking the revocation status of a delegation token, a query with the Delegation Token Revocation Authority will suffice. As with revocation lists of identity certificates or CRLs [20], the Delegation Token Revocation Authority will have revocation lists of delegation tokens, thus providing the list to an entity in the event of a query. The list identifies the delegation tokens that have been revoked by means of the `subject` of each of these. Once the list has been received, the requesting entity will process it and check the revocation status of the token.

Although we have chosen to use lists to check the revocation status of delegation, we might easily have decided to use a query protocol similar to OCSP [32]. Here, the Delegation Token Revocation Authority would respond in an individualized manner to queries on the revocation status of a specific token identified univocally by its `subject` field.

Both solutions are valid, although each has its own advantages and drawbacks. Using lists offers the advantage of being able to make queries on revocation status even when the Delegation Revocation Authority is not available, as the entity may have a prior list. It also poses a problem of privacy concerning the information contained in the list arising from updating of the list of revoked tokens. Generally, entities will download revocation lists periodically, so they will not be aware of revocations that have taken place in the interim between updates. Using the OCSP protocol involves the opposite advantages and drawbacks: namely, it suffers from the drawback that if the Delegation Token Revocation Authority is not available, no query can be made of the revocation status of a token, with the advantage that any reply to a revocation query will reflect the latest status of the token in question, as no update intervals exist.

Bearing in mind the foregoing, the process of revoking a delegation consists of a notification by the delegator of the DTRA of the subject field of the token to be revoked and verification by the DTRA that the token has been generated by the delegator requesting revocation. In response to a service request, the Service Provider may use the DTRA to verify whether the delegation token has been revoked or not. This revocation solution is valid for delegation tokens generated by the delegator, as presented in Sect. 4.5. Proxy certificates giving rise to a delegation token generated by an entity other than the delegator will require a separate revocation process,

and the revocation process discussed herein would apply only to the delegation token, not the proxy certificate on which it is based. This problem of revoking proxy certificates is not addressed in this paper, as it is assumed that all tokens are of the first type.

Because delegation tokens are based on proxy certificates and that the delegation solution as a whole relies on the use of PKIs, we propose integrating the Delegation Token Revocation Authority as a Trusted Third Party (TTP) in the PKI scheme in which the delegation solution is being applied. DTRA is a TTP that, as explained in previous paragraphs, cannot revoke a delegation token without the delegator. The inclusion of the DTRA as TTP in the PKI reduces operational risks, as it is assumed that DTRA is forced by law to implement stringent audit measures in order to guarantee proper operation (identification and authentication of authorized users, access controls, audit, inspections conducted to provide confidence that appropriate controls are in place, etc.). Since the proposed solution applied to an e-government environment, we can also assume that, in many cases, DTRA will be managed by government.

4.5 Identity delegation interactions

The delegation process presented in this section is based on the delegation token discussed in the preceding section, while also using as its starting point the generic identity management scenario presented earlier which reflects the reality of most identity management and service provision systems being used at present in Public Administrations in Europe. Initially, the participating entities are as we have discussed, namely: the delegator, delegatee, Service Provider, and the Identity Provider.

The communication model and the interactions in the provision of a service with delegation would be as depicted in the figures below. The sequence in which the information is exchanged is as follows: The steps shown in Fig. 13 represent the interaction of a citizen or entity acting as a delegator with the Identity Provider: that is, the phase of authentication of the delegator. The details are as follows:

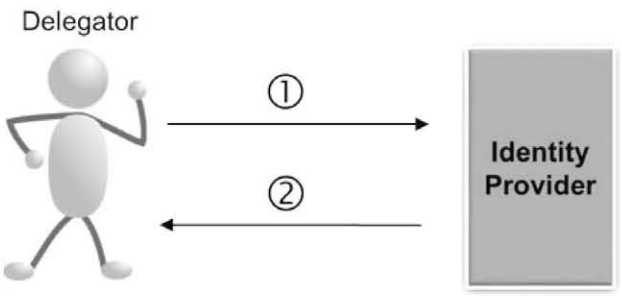

Fig. 13 Model of service provision with delegation: authentication

1. First, the delegator presents credentials in order to be authenticated. As in most countries of the European Union, there are different types of authentication: from simple ones like a user login with a password to the most complex and secure X.509 certificates. Given that the mechanism of authentication with X.509 certificates reflects the present trend in Europe as a whole, it will be adopted herein from this point onward. In addition to the authentication process, the delegator requests from the Identity Provider an attribute statement that includes attributes necessary for accessing the service(s) to be delegated. That is, the Identity Provider is asked for an SAML v2 assertion that contains user attributes related to the delegator as required by the Service Provider in allowing access to the service(s). For example, if the delegator wishes to delegate access to the service of applying for unemployment benefits, the Identity Provider can deliver an SAML assertion indicating that the delegator is of legal age and unemployed.
2. After checking the credentials of the delegator and verifying that everything is in order, the Identity Provider returns a signed SAML assertion that includes the requested attributes. The delegator has the task of checking the integrity of the reply, the validity of the Identity Provider's signature and the accuracy of the set of attributes about its identity contained in the SAML assertion.

Following authentication and acquisition of the attributes statement in the form of an SAML assertion necessary for access to the service(s) of the Service Provider, the process of delegation takes place, in which the delegator issues a delegation token in accordance with the conditions set forth above and delivers it to the delegatee. In short, the delegator will build, on its own, a delegation token that will include, through a set of non-critical extensions, the SAML v2 assertion with the attribute statement of the delegator issued and signed by the Identity Provider and the indication of the set of services offered by the Service Provider which the delegatee may access. It will now be shown as step three in graphics.

This process may appear complex, but the degree of complexity is lessened; if we recall that the majority of European public administrations and many administrations outside Europe, citizens have smart cards with encryption capacities that are now used in secure telematic transactions with administrations in which they can generate keys or digital signatures.

Now in possession of the delegation token, the delegatee may access services provided by the Service Provider, but in accordance with the limitations established by the delegator in the delegation token.

Figure 14 depicts access to the service by the delegatee and delivery of results to the delegator.

3. The delegator issues the identity delegation token and sends it to the delegatee through a secure channel.
4. Now in possession of the delegation token, the delegatee has an element that allows it to request the service from the Service Provider in the name of the delegator. The Service Provider checks the validity of the delegation token and the validation path. As we have seen, the delegation token is linked to the delegator by the signature, so the Service Provider knows on whose behalf the delegatee is accessing the service. Hence, on the basis of this link, as well as the SAML assertion with the attribute statement and the access constraints specified in the delegation token through the extension `serviceIRIConstraints`, the Service Provider can make authentication and authorization decisions pertinent to allowing or denying access to the service as requested by the delegatee.
5. Assuming that everything is in order, the Service Provider delivers information on the service requested to the delegatee.
6. Once the interaction with the Service Provider is over, the delegatee sends the delegator the results of the service requested.

As is clear, step 2 (Fig. 13) consists of verifying the signature. This involves interaction between the delegator with the PKI to check if the Identity Provider's certificate has been revoked or not. Further, the Service Provider must verify the delegator's signature in step 4, thus requiring it also to interact with the PKI. This type of validation between citizens or entities and the PKI are now common in telematic access to services provided by public administrations, and thus pose no special difficulties. This same step 4 also contains another validation process that must be prior to the signature validation process: the process of validating the delegation token. The delegation token validation process consists mainly of verifying two items: that the token delegation path is correct and that the token has not been revoked.

Given that the delegation token is based on proxy certificates, its delegation path validation process is similar to that of the latter. Validating the delegation token requires the distinguished name and the public key of the X.509 certificate of the delegator. If we use the more complex delegation model: that is, if we take into account the possibility of indirect delegation, where the delegation path may be comprised of n delegation tokens, we must verify that the delegation path—that is, the chain of n delegation tokens—meets the following conditions:

a For any x in the set between 1 and n−1, the value of `subject` in the delegation token x matches the value of the issuer field of the delegation token x + 1 and the subject distinguished name of the delegation token x + 1 is a legal subject distinguished name to have been issued by the delegation token x.
b Delegation token 1 is a valid delegation token issued by the end entity certificate (EEC) of the delegator. This certificate is provided as an input to the process of validating the delegation path.
c Delegation token n is the delegation token to be validated.
d For any x in the range between 1 and n, the delegation token x is valid at the time in question.
e For all delegation tokens in the path with a certain value in `pCPathLenConstraint`, the number of succeeding delegation tokens in the delegation path does not exceed the length specified in said field.

Of special interest in validating the delegation path is section d, which says that the delegation token must be valid at the time in question. This involves a recurring process in which each of the delegation tokens involved in the delegation path must verify that the delegation path leading up to it is correct and, further, that the delegation token has not been revoked. In our scenario, the entity that must verify both the delegation token and the revocation status of delegation tokens is the Service Provider. Hence, a breakdown of step 4 shows what might be considered an interaction between the Service Provider and the Delegation Token Revocation Authority. Depicted in Fig. 15, it includes the Delegation Token Revocation Authority and shows steps 4a and 4b of the process of querying the revocation status of the delegation token on the part of the Service Provider. Integrating the three preceding figures yields our definitive proposed model of service provision with delegation, as shown in Fig. 16.

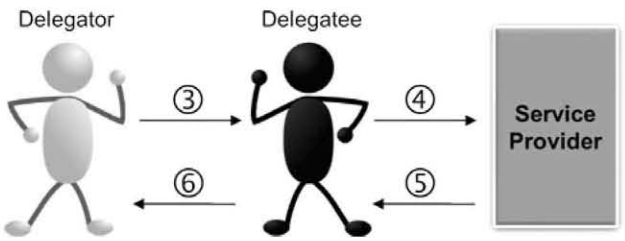

Fig. 14 Model of service provision with delegation: the delegator receives the service through the delegatee

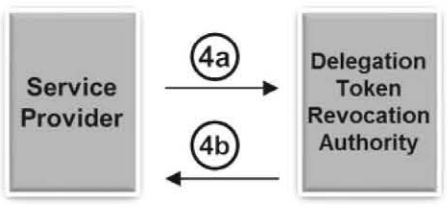

Fig. 15 Model of service provision with delegation: process of querying revocation status of delegation token

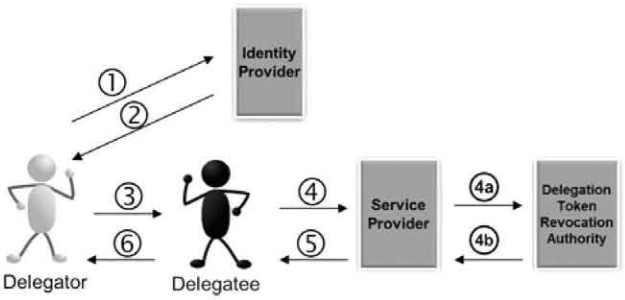

Fig. 16 Model of service provision with delegation and revocation query

5 Technical feasibility

With a view to demonstrating the technical feasibility of the delegation token proposed, it has been implemented in a limited and simplified scenario that reflects, on a small scale, a real application scenario.

Four entities are involved in the small scenario used for the tests—delegator, delegatee, Service Provider and the Identity Provider—and these engage in the interactions discussed in the section providing details on the solution and shown in Fig. 16.

All development was done in Java™ [33] by means of the JDK 6 Update 18 [34] and the Netbeans 6.8 integrated development environment [35]. Both the Identity Provider and the Service Provider are constituted by means of web services implemented in Java and accessible through an Apache-Tomcat 6.0.26 server [36].

To enable issuance of delegation tokens and signatures, and to verify signatures and include extensions, the functionalities provided by Java JDK through APIs and toolkits had to be extended. Specifically, Java's cryptography support was enhanced with a cryptography API provided by Bouncycastle [37]. SAML assertions and their integration in proxy certificates which are the basis of the delegation tokens were handled with a part of the Globus Toolkit ® [38] combined with the GridShib plugin [39].

5.1 Applicability in a real environment

To demonstrate its applicability, we shall now present the particular features of a model for possible provision of services by the Spanish Public Administration. Specifically, our example consists of a hypothetical Spanish citizen who is obliged to undertake administrative processes with the State Tax Revenue Agency telematically and decides to entrust the task to an agent, and hence must delegate part of his identity. The agent, in the role of delegatee, will undertake the steps that the citizen, in the role of delegator, has entrusted through use of the identity attributes necessary to access the service.

As the communication model would be similar to that of the generic model, the most important task in specifying the model for a real environment is to clearly define which real entities correspond to which entities in the model. In our scenario:

- *Delegator*: A citizen who seeks to carry out bureaucratic procedures with the state tax revenue agency but who decides to delegate.
- *Delegatee*: The agent to whom the citizen delegates part of his identity.
- *Identity provider*: The entity responsible for managing citizens' attributes information, authenticating citizens by means of their digital identity and for generating signed attribute statements through SAML 2.0 assertions. In Spain, this task is performed by a platform supported by the national police force that validates digital identities in the electronic national identity card. This platform does not yet have the capacity to generate SAML assertions with identity attributes, but one assumes that it would acquire the capability for such operations if a model such as the one discussed herein is adopted.
- *Service provider*: In our scenario, this would be the State Tax Revenue Agency (Agencia Estatal de Administración Tributaria).
- *Delegation Token Revocation Authority*: Given that this entity is one of the contributions made by this paper, no entity yet performs the tasks proposed for this authority. As we have noted, it is expected to be integrated in the national PKI as a trusted third party (TTP).
- *PKI*: In Spain, the PKI is closely linked to citizens' digital identities, as these entities are based on X.509 certificates. Nationally, there are two authorities—the Ministry of Public Administrations and the Royal Spanish Mint. For the sake of simplicity, we will use the Ministry of Public Administrations in this example.

Figure 17 depicts this scenario and the interactions between them, the details of which are as follows:

1. The citizen (delegator) is authenticated by means of his electronic ID card and uses his X.509 certificate with the authentication platform of the national police force. Further, this process also requests from the platform delivery of an attribute statement using an SAML v2 assertion to include all the user attributes required by the State Tax Revenue Agency to allow access to the service to be delegated.
2. Once the national police's force authentication platform verifies who the citizen claims to be through the X.509 certificate, it returns a signed SAML attribute statement with the attributes requested. The citizen verifies the

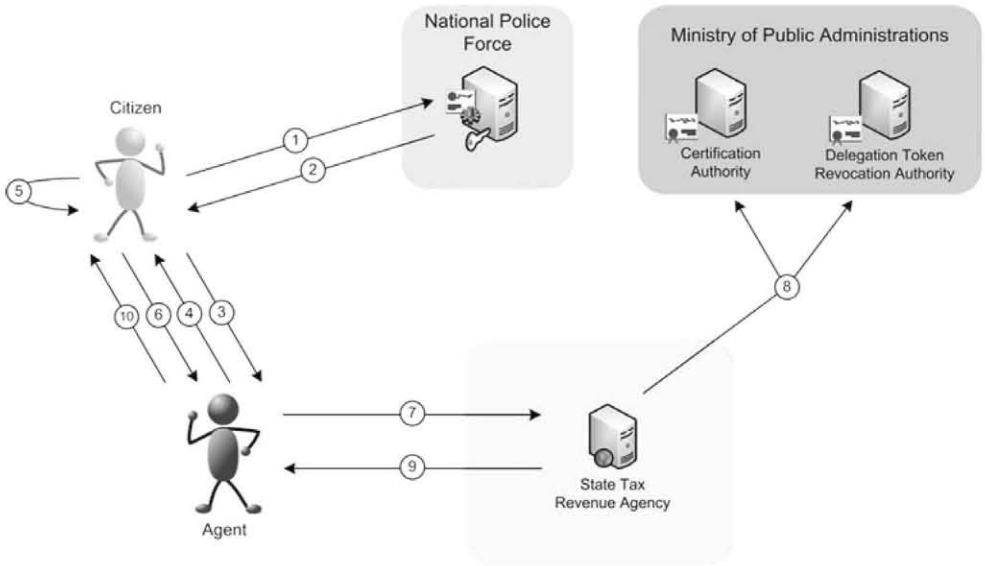

Fig. 17 Real scenario of application of identity delegation solution and interactions between entities

integrity of the reply, the validity of the signature and checks the accuracy of the identity attributes asserted in the SAML v2 attribute statement delivered by the authentication platform.

3. The citizen contacts a trusted agent and requests that he initiate the identity delegation process in order to access to a service provided by the state tax revenue agency.
4. In order to obtain a delegation token that will authorize him to act as a delegatee before the State Tax Revenue Agency, the agent issues an asymmetrical key pair—a public key and a private key—and sends the citizen the public key through a secure channel like SSL.[4]
5. The citizen forms a delegation token with the public key received from the agent, the attribute statement signed by the authentication platform of the national police force and the indication in the IRI of the service offered by the State Tax Revenue Agency needed to perform the steps commissioned to the delegatee. This involves generation of a proxy certificate for the public key received and the addition of extensions and the IRIs of the services, yielding the delegation token as explained in 4.3.3 and 4.5.
6. When the delegation token is issued, the citizen sends it to the agent through a secure channel.
7. The agent now has a delegation token that allows him to request provision of the service from the State Tax Revenue Agency on behalf of the citizen.
8. The State Tax Revenue Agency will validate the delegation token and verify that the delegation path is correct and that the token has not been revoked. It will also check the name of the citizen on whose behalf the agent is seeking to accede and whether the token has been generated for that delegatee (it can check this because only the delegatee has the private key paired with the public key included in the delegation token).
9. Therefore, on the basis of the SAML assertion with the attribute statement and the IRI of the service specified in the delegation token, the State Tax Revenue Agency will decide if it can provide the service requested by the agent acting in the name of the citizen. Assuming that everything is in order, the State Tax Revenue Agency will deliver to the agent the results of the provision of the service requested.
10. Once the interaction with the State Tax Revenue Agency has concluded, the agent will provide the citizen with the results of the service requested.

It is clear that this delegation solution can be perfectly integrated in a real scenario. If we use Spain as our scenario, the modifications to be made can be summarized as follows:

- Add a trusted third party, the proxy certificate revocation authority, to the present PKI infrastructure.
- Centralize user attributes necessary for accessing services in a single entity that acts as the Identity Provider, and enable it to support the issuance of SAML attribute statements. As the present electronic ID platform acts as the Identity Provider in Spain, it would be sufficient to enact these changes in said platform.
- Modify Service Providers with a view to basing access control on attributes and accepting delegation tokens. As most Service Providers in the present Public Adminis-

[4] This secure channel can be implemented with smart cards and by establishing SSL communication between the delegator and the delegatee with use of their respective certificates.

tration base user authentication and service provision on X.509 certificates, modifying their behavior to enable them to understand delegation tokens should be a simple task. Ultimately, it is sufficient to know how to process a definite set of X.509 v3 certificate extensions.
- Provide users with the tools they need to use their electronic ID cards to issue delegation tokens and complete the related steps (revocation, etc.).

6 Conclusions and future research

This paper provides a solution to a problem that is both ongoing and subject to public demand: delegation of identity in accessing telematic services generally and, in particular, those provided by Public Administrations. The solution herein lends a telematic form to processes of representation traditionally found in interactions between citizens and entities or Public Administrations and enshrined in the law of many member states of the European Union. It also facilitates pan-European implementation and expansion of the potential offered by Public Administrations to undertake telematic procedures.

This solution is notably simple to implement and can easily be integrated in present-day identity management systems in most member states of the European Union; hence, in principle, it can be quickly adopted and implemented in present scenarios of provision of telematic services by Public Administrations.

Thus, the solution is remarkably robust, as well as flexible and easy to use from the point of view of both users and Public Administrations; the legitimacy of the origin and the integrity of the delegation token can be checked at any time. It enables mechanisms for canceling the validity of a delegation token that has been delivered. For such a cancelation to take effect, mechanisms exist to query the status of delegation tokens. In the solution proposed, flexibility and dynamism have been enhanced. For instance, users need not engage in any processes of pre-registration or notification of administrative authorities when delegating. In like manner, delegation is effected immediately, with no complex steps or participation from a large number of entities, unlike the few scenarios in which delegation for telematic processes is now permitted.

One significant and novel aspect of this solution lies in its provision of mechanisms for achieving a high degree of granularity. Users can establish specific restrictions on the processes delegated and Service Providers can control and limit the use of delegation by a representative.

The solution opens up new possibilities and fields of study that should be considered for future research. This paper proposes the existence in each application environment of an entity called the Delegation Token Revocation Authority that would be responsible for maintaining the revocation status of a delegation token. Further research should determine whether more than one such entity should exist in a scenario and, if so, how their hierarchical organization might optimize query and revocation processes, providing answers to questions such as: Which entity should notified of a revocation? or, Which of them should reply to a query concerning a specific delegation token?

Moreover, the solution to identity delegation is circumscribed to a specific, controlled environment, namely the Public Administration, where all digital identities are issued by the same provider and in the same format. In the future, research should address possible application of this solution in more heterogeneous environments like citizens' interactions with private entities like banks or in environments dedicated to specific groups, like those related to eHealth scenarios.

Another issue not addressed in this solution is anonymity in the delegation process. This solution enables dynamic delegation in which the Service Provider knows the identity of both the delegator and the delegatee. Future research would also envisage the possibility of a form of identity delegation that is similar yet which supports anonymity, that is: a dynamic identity delegation process in which the delegator, the delegatee or both are anonymous to the Service Provider, while ensuring the guarantees required from a security perspective. This potential would open new doors to the use of delegation in fields where anonymity is crucial, such as processes of citizen participation or electronic voting. As some countries today allow for delegated voting, the possibility of integrating the identity delegation solution proposed herein in a telematic voting system in a way that maintains all the security assurances and restrictions inherent to such systems would pose a major challenge.

Acknowledgments This paper is part of the work being conducted by the authors in the project ADMISSION (TSI2006-4864), Telematic platform for e-Government based on a choreography of services, supported by the Ministry of Education and Science of Spain through the National Plan for R+D+I.

References

1. Decision 2004/387/EC of the European Parliament and of the Council of 21 April 2004 on the interoperable delivery of pan-European eGovernment services to public administrations, businesses and citizens (IDABC). Off. J. Eur. Union Luxembourg, **L 181**, 25–35 (18 May 2004)
2. Communication from the Commission to the Council, the European Parliament, the Ecomnomic and Social Commitee and The Comittee of Regions—eEurope 2005: An information society for all [COM(2002) 263 final]. Brussels (28 May 2002)
3. Communities, Commission of the European. Communication from the Commission to the Council, the European Parliament, the European Economic and Social Committee and the Committee of the

Regions. i2010 eGovernment Action Plan: Accelerating eGovernment in Europe for the Benefit of All. (25 Apr 2006). http://ec.europa.eu/idabc/servlets/Doc?id=25286
4. General, European Commission Information Society and Media Directorate. Signpost towards eGovernment 2010. (Nov 2005)
5. AAVV: A Roadmap for a pan-European eIDM Framework by 2010. Version 1.0. http://ec.europa.eu/information_society/activities/egovernment/docs/pdf/eidm_roadmap_paper.pdf. Accessed Aug 2010
6. Graux, H., Dumortier, J.: Report on the State of Pan-European eID Initiatives. Technical Department, ENISA (2009)
7. Team, The Modinis IDM Study: Modinis Study on Identity Management in eGovernment: Common Terminological Framework for Interoperable Electronic Identity Management. Version 2.01. (23 Nov de 2005). http://www.cosic.esat.kuleuven.be/modinis-idm/twiki/bin/view.cgi/Main/GlossaryDoc. Accessed Aug 2010
8. Brmme, A., Busch, C., Hhnlein, D.: Cross-Context Delegation through Identity Federation. In: Peeters, R., et al. (eds.) Proceedings of the Special Interest Group on Biometrics and Electronic Signatures. Vols. Lecture Notes in Informatics (LNI) P-137. Bonner Kllen Verlag, pp. 79–92 (2008)
9. Alrodhan, W., Mitchell, C. J.: A delegation Framework for Liberty. In: Proceedings of the 3rd Conference on Advances in Computer Security and Forensics (ACSF'08). Liverpool John Moores University, Liverpool, pp. 67–73 (2008)
10. Sandhu, R., et al.: Role-based access control models. IEEE Comput. **29**, 38–47 (2006)
11. Zhang, L., Ahn, G., Chu, B.: A rule-based framework for role-based delegation. In: Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies. pp. 153–162 (2001)
12. Welch, V., et al.: X.509 Proxy Certificates for Dynamic Delegation. 3rd Annual PKI R&D Workshop (2004)
13. Gomi, H., et al.: A delegation framework for federated identity management. In: Proceedings of the 2005 Workshop on Digital identity management. Fairfax, VA, USA (2005)
14. Farrell, S., Housley, R.: An Internet Attribute Certificate Profile for Authorization. IETF RFC 3281 (2002)
15. Liberty Alliance Project. http://www.projectliberty.org. Accessed Aug 2010
16. Komura, T., et al.: Proposal of delegation using electronic certificates on single sign-on system with SAML-protocol. SAINT'09. Ninth Annual International Symposium on Applications and the Internet (2009)
17. Tuecke, S., et al.: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. IETF RFC 3820 (2004)
18. Ragouzis, N., et al.: Security Assertion Markup Language (SAML) V2.0 Technical Overview. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.pdf. v16. (2008), Accessed June 2010
19. Housley, R., et al.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF RFC 3280 (2002)
20. Cooper, D., et al.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF RFC 5280 (2008)
21. Myers, M., et al.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol—OCSP. IETF RFC 2560 (1999)
22. OASIS: SAML. Online Community for the Security Assertion Markup Language (SAML) OASIS Standard. http://saml.xml.org/saml-specifications. Accessed June 2010
23. Maler, E., Mishra, P., Philpott, R.: Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1. http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf. Accessed June 2010
24. OASIS: OASIS Security Services Technical Committee. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security. Accessed June 2010
25. OASIS: Organization for the Advancement of Structured Information Standards. http://www.oasis-open.org/home/index.php. Accessed June 2010
26. GridShib: http://gridshib.globus.org/. Accessed June 2010
27. VVAA: An X.509 Binding for SAML. http://spaces.internet2.edu/display/GS/X509BindingSAML
28. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifier (URI): Generic Syntax. IETF RFC 3986 (2005)
29. Web Services Addressing 1.0—Core: W3C Recommendation. http://www.w3.org/TR/2006/REC-ws-addr-core-20060509. Accessed May 2010
30. Duerst, M., Suignard, M.: Internationalized Resource Identifiers (IRIs). IETF RFC 3987 (2005)
31. Federal Information Processing Standards Publication (FIPS PUB) 180-2. Secure Hash Standard (2002)
32. Myers, M., et al.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol—OCSP. IETF RFC 2560 (1999)
33. Oracle Corporation: Java. http://java.sun.com/. Accessed June 2010
34. Java SE Downloads: http://java.sun.com/javase/downloads/widget/jdk6.jsp. Accessed June 2010
35. Oracle Corporation: Netbeans IDE. http://netbeans.org/. Accessed June 2010
36. The Apache Software Foundation: Apache Tomcat. http://tomcat.apache.org/. Accessed June 2010
37. The Legion of the Bouncycastle. Bouncy Castle Crypto APIs for Java. http://www.bouncycastle.org/java.html. Accessed June 2010
38. University of Chicago. Globus Toolkit. http://www.globus.org/toolkit/. Accessed June 2010
39. Board of Trustees of the University of Illinois. GridShib. http://gridshib.globus.org/. Accessed June 2010