A First Approach to Provide Anonymity in Attribute Certificates*

Vicente Benjumea, Javier Lopez, Jose A. Montenegro, and Jose M. Troya

Computer Science Department
University of Malaga, Spain
{benjumea,jlm,monte,troya}@lcc.uma.es

Abstract. This paper focus on two security services for internet applications: authorization and anonymity. Traditional authorization solutions are not very helpful for many of the Internet applications; however, attribute certificates proposed by ITU-T seems to be well suited and provide adequate solution. On the other hand, special attention is paid to the fact that many of the operations and transactions that are part of Internet applications can be easily recorded and collected. Consequently, anonymity has become a desirable feature to be added in many cases. In this work we propose a solution to enhance the X.509 attribute certificate in such a way that it becomes a conditionally anonymous attribute certificate. Moreover, we present a protocol to obtain such certificates in a way that respects users' anonymity by using a fair blind signature scheme. We also show how to use such certificates and describe a few cases where problems could arise, identifying some open problems.

Keywords: Authorization, PMI, anonymity, pseudonym, credential, X.509 attribute certificates

1 Introduction

Identity certificates (or *public-key certificates*) provide the best solution to integrate the authentication service into most of those applications that are developed for the Internet and make use of digital signatures. The use of a wide-range authentication service based on identity certificates is not practical unless it is complemented by an efficient and trustworthy mean to manage and distribute all certificates in the system. This is provided by a *Public-Key Infrastructure* (PKI).

However, new applications, particularly in the area of e-commerce, need an authorization service to describe what the user is granted to. In this case, privileges to perform tasks should be considered. Thus, for instance, when a company needs to establish distinctions among their employees regarding privileges over

* This work has been partially supported by the Spanish Ministry of Science and Technology under the Project TIC2002-04500-C02-02

resources, the authorization service becomes important. Different sets of privileges over resources (either hardware or software) will be assigned to different categories of employees. Also, in those distributed applications where company resources must be partially shared through the Internet with other associated companies, providers, or clients, the authorization service becomes an essential part.

Authorization is not a new problem, and different solutions have been used in the past. However, traditional solutions are not very helpful for many of the Internet applications. *Attribute Certificates*, proposed by the ITU-T (International Telecommunications Union) in the X.509 Recommendation [14], provide an appropriate solution. Additionally, the attribute certificates framework defined by ITU provides a foundation upon which a *Privilege Management Infrastructure* (PMI) can be built.

On the other hand, during last years users have paid special attention to the problem caused by the fact that many of the operations and transactions they carry out through the Internet can be easily recorded and collected. Thus, anonymity has become a desirable feature to be added in many cases.

Since early 80's many studies have been oriented towards the protection of users' privacy in electronic transactions [4–6, 18]. Those studies have originated with new cryptographic primitives and protocols that have been applied to several specific applications oriented to solve some specific problems such as electronic cash [8], electronic voting [1, 10, 12], and others, and some proposals with a multi-purpose point of view that cope with organizations and credentials [6, 7, 9, 15, 16]. However, such a technology have not been transferred to general applications in the real world. To the best of our knowledge, only one system have been designed and implemented with a practical point of view [2, 3]. However, even this system does not follow proposed standards such as X.509 attribute certificates.

It is our belief that one of the main steps to transfer such a technology to multi-purpose real world applications is the ability to apply them to open standard systems. Therefore, in this paper we show a first approach to provide anonymity in X.509 attribute certificates, transferring *fair blind signature* schemes to those standard certificates, and defining *Anonymous Attribute Certificates* in which the holder's identity can be conditionally traceable depending on certain conditions.

The structure of the paper is as follows. In section 2 we briefly argue the use of blind signatures as basic construction block for our solution. Section 3 describes the standard X.509 attribute certificates proposed by ITU-T, and how the framework that this type of attributes define is linked to PKIs. Section 4 describes, throughout three subsections the overview of the scheme, the adaptation of attribute certificates to support anonymity, and the protocol for a user to obtain an anonymous attribute certificate. Section 5 concludes the paper, presenting an interesting discussion about results and open issues.

2 Blind signatures as a basic construction block

It is widely known that *blind signature* protocols [5] provide a mean for a signer to sign a message sent by an entity. The signer is unable to know anything about the message, and can not link the signed message with its originator.

These schemes have been widely studied and applied to solve specific problems where anonymity is fundamental, such as electronic voting systems [1, 10, 12] and electronic cash [8]. However, these schemes present an open door for fraud, since perfect anonymity offers the best coverage for dishonest behaviour [19]. Therefore, these schemes must be used with the maximum of caution, by subjects under control, and where perfect anonymity is the only solution to the problem.

Other schemes have been developed to avoid that inconvenience. *Fair blind signature* protocols [18, 11] try to close the gap between anonymity and fairness. In these schemes, the anonymity can be broken and the signed message can be linked (only under certain conditions) with the person who requested such a blind signature. In these cases, a *Trusted Third Party* (TTP) is needed in order to run the protocol, and the collusion of the TTP with the signer and the signed message is a necessary condition.

3 X.509 Attribute Certificates

One of the main advantages of an attribute certificate is that it can be used for various purposes. It may contain group membership, role, clearance, or any other form of authorization. A very essential feature is that the attribute certificate provides the means to transport authorization information in distributed applications. This is especially relevant because through attribute certificates authorization information becomes "mobile", which is highly convenient for Internet applications.

The mobility feature of attributes have been used in applications since the publication of the 1997 ITU-T X.509 Recommendation [13]. However, it has been used in a very inefficient way. That recommendation introduced an ill-defined concept of attribute certificate. For this reason, most of actual applications do not use specific attribute certificates to carry authorization information. On the contrary, attributes of entities are carried inside identity certificates. The *subjectDirectoryAttributes* extension field is used for this purpose. This field conveys any desired directory attribute values for the subject of the certificate, and is defined as follows:

```
subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX   AttributesSyntax
    IDENTIFIED BY id-ce-subjectDirectoryAttributes }
AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute
```

This solution does not make entity attributes independent from identity, what can cause problems. Firstly, this is not convenient in the frequent situations where the authority issuing the identity certificate is not the authority for the assignment of privileges. Secondly, even in the situations where the authority is the same one, we must consider that life of identity certificates is relatively long when compared to the frequency of change of user privileges. Therefore, every time privileges change it is necessary to revoke the identity certificate, and it is already known that certificate revocation is a costly process.

Moreover, many applications deal with authorization issues like delegation (conveyance of privilege from one entity that holds a privilege to another entity) or substitution (one user is temporarily substituted by another user, and this one holds the privileges of the first one for a certain period of time). Identity certificates support neither delegation nor substitution.

The most recent ITU-T X.509 Recommendation of year 2000 provides an approach to these problems because it standardizes the concept of attribute certificate, and defines a framework that provides the basis upon which a PMI can be built. Precisely, the foundation of the PMI framework is the PKI framework defined by ITU. In fact, ITU attribute certificates seem to have been mainly proposed to be used in conjunction with identity certificates; that is, PKI and PMI infrastructures are linked by information contained in the identity and attribute certificates (figure 1).

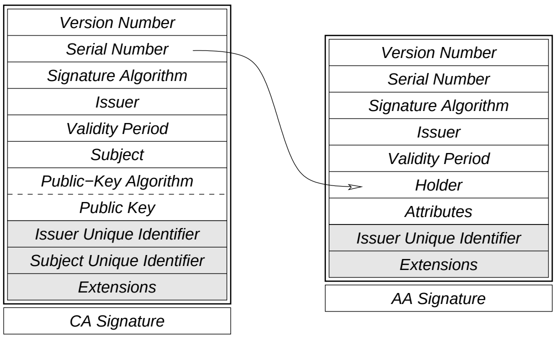

Fig. 1. Relation between identity and attribute certificates

Although linked, both infrastructures can be autonomous, and managed independently, what provides a real advantage. In the most recent recommendation, attribute certificates are conveniently described, including an extensibility mechanism and a set of specific extensions. A new type of authority for the assignment of privileges is also defined, the *Attribute Authority* (AA), while a special type of authority, the *Source of Authority* (SOA), is settled as the root of delegation chains. The recommendation defines a framework that provides a foundation upon which a PMI is built to contain a multiplicity of AAs and final users. Revocation procedures are also considered by defining the concept of *Attribute Certificate Revocation Lists*, which are handled in the same way as *Certificate Revocation Lists*, published by *Certification Authorities* (CAs) in the PKI case.

As shown in figure 1, the field *holder* in the attribute certificate contains the *serial number* of the identity certificate. As mentioned in [17], it is also possible to bind the attribute certificate to any object by using the hash value of that object. For instance, the hash value of the public key, or the hash value of the identity certificate itself, can be used. All possibilities for the binding can be concluded from the ASN.1 specification of the field *holder*, where other related data structures are also specified:

```
Holder  ::= SEQUENCE {
    baseCertificateID   [0] IssuerSerial     OPTIONAL,
    entityName          [1] GeneralNames     OPTIONAL,
    objectDigestInfo    [2] ObjectDigestInfo OPTIONAL
}

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0] INSTANCE OF OTHER-NAME,
    rfc822Name                  [1] IA5String,
    dNSName                     [2] IA5String,
    x400Address                 [3] ORAddress,
    directoryName               [4] Name,
    ediPartyName                [5] EDIPartyName,
    uniformResourceIdentifier   [6] IA5String,
    iPAddress                   [7] OCTET STRING,
    registeredID                [8] OBJECT IDENTIFIER
}

ObjectDigestInfo    ::= SEQUENCE {
    digestedObjectType  ENUMERATED {
        publicKey               (0),
        publicKeyCert           (1),
        otherObjectTypes        (2)
   },
    otherObjectTypeID       OBJECT IDENTIFIER  OPTIONAL,
    digestAlgorithm         AlgorithmIdentifier,
    objectDigest            BIT STRING
}
```

As we will see in next section, the content of this specification is essential for the scheme that we have developed.

4 Introducing anonymity into attribute certificates

4.1 Overview of the scheme

Our scheme coexists with standards PMI and PKI. While a PKI provides support for users' identities, the AA issues certificates about attributes that the users hold. Additionally, we suppose that some organizations provide services to users based on their respective attributes. We have introduced in the scheme a TTP which provides (in collusion with the AAs) the ability to disclose anonymous users' identities. Some of the AAs will have the special capacity to issue anonymous attribute certificates. Each of those AAs is in connection with several Attribute sub-Authorities, that will be in charge of verifying that a user fulfills the requirements needed to obtain an “anonymous” certificate containing a specific attribute (figure 2).

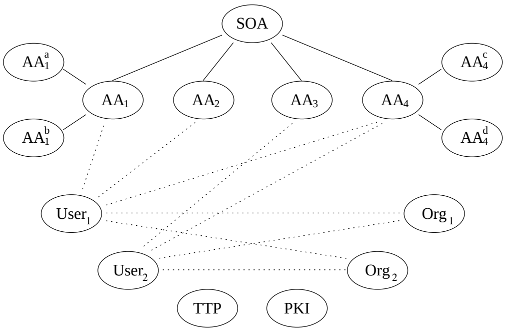

Fig. 2. System Overview

The role that the actors play in our solution can be roughly seen as follows. A user can anonymously acquire as many pseudonyms as he needs from the TTP, where the validity of the pseudonyms are limited in time. Every obtained pseudonym is composed by two related parts: one of them is public and the other one is private. In the following we will refer these parts as public pseudonym and private pseudonym respectively. The TTP keeps such a relationship until the end of the validity period. For each anonymous certificate that the user wants to get, he will collect all proofs needed to apply for a specific attribute (or set of attributes), and will send the proofs, together with his identity and his public pseudonym, to the Attribute sub-Authority in charge of verifying such proofs. If the set of proofs is complete, a special token related to the public pseudonym will be issued (by using a fair blind signature scheme), and a link stating the relationship between the user’s identity and his public pseudonym will be stored.

This special token will be modified (again, using a fair blind signature scheme) by the user in order to hide its relationship with the public pseudonym and will reflect, since that moment, the relationship with the private part. This token, now associated with the private pseudonym, will be used by the user to anonymously apply for an anonymous attribute certificate to the AA. Note that if the anonymous user holds that token, then he fulfills the requirements needed to get the certificate containing a (set of) specific attributes.

Once the AA checks that everything is correct, it issues the certificate of the attributes that corresponds with the Attribute sub-Authority that issued such a token. As stated, these certificates are issued anonymously and are related with the user’s private pseudonym. Therefore, nobody can link them with the real users’ identity unless the TTP and the Attribute sub-Authority collude and some conditions are met. By definition, it is supposed that the TTP will remain trusted and will not reveal the link between both parts of the pseudonym unless a condition expressed in the certificate is fulfilled and such a condition is signed by the user and the AA.

The user will make use of the attribute certificate in order to enforce his privileges. As it is anonymous, it is not linked to any PKI. However it contains a public key and the user who knows the corresponding private key is considered the one who owns such an attribute.

4.2 Adapting attribute certificates to support anonymity

In section 3 we have mentioned that the field *holder* of the attribute certificate can contain the digest of any object. Thus, we will define an object, called *Pseudonym Structure* (figure 3), to support the conditionally anonymity of the owner and will link such an object with the attribute certificate by using this field. The pseudonym structure fields are the following ones:

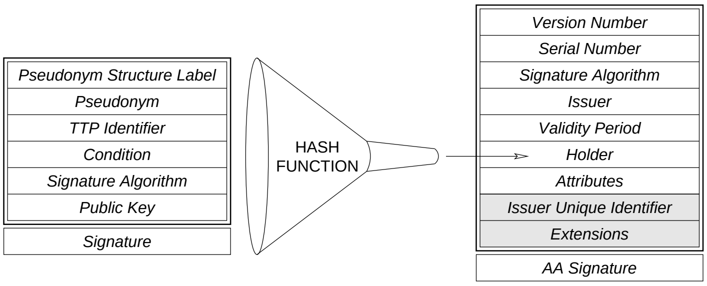

Fig. 3. Relation between pseudonym structure and attribute certificate

- *Pseudonym Structure Label*: A static field that allows us to interpret the object as a proper pseudonym structure.
- *Pseudonym*: The holder's private pseudonym, issued by the TTP specified in the next field.
- *TTP Identifier*: The issuer of the pseudonym, that keeps a record linking the private pseudonym with the public one.
- *Condition*: The condition under which, if fulfilled, both the TTP and the AA will collude and will reveal the user's identity.
- *Signature Algorithm*: Identifies the algorithm for signature and verification of documents using the public key stored in the next field.
- *Public Key*: The key used to authenticate the owner of the attribute certificate in such a way that the anonymous user who holds the corresponding private key will be the attribute owner. For a proper authentication procedure, the anonymous user should sign a challenge with that private key every time that authentication is needed.
- *Signature*: The anonymous user signs the pseudonym structure to prove that it is a valid structure and that he knows the corresponding private key. Moreover, the signature is the proof that the anonymous user accepts the condition stated above with respect to revealing his real identity.

We will define a *conditionally anonymous* X.509 attribute certificate as the attribute certificate itself together with the pseudonym structure, linked by mean

Nomenclature	Meaning
$A : act$	$A's$ action act
$A \rightarrow B : m$	m is sent from A to B
$m = (m_1, m_2)$	m is composed by m_1 and m_2
$c = E_z\left(m\right)$	m is encrypted with the symmetric key z
$m = D_z\left(c\right)$	c is decrypted with the symmetric key z
A_{publ}, A_{priv}	$A's$ asymmetric public and private keys
$c = E_A\left(m\right)$	m is encrypted with $A's$ asymmetric public key
$m = D_A\left(c\right)$	c is decrypted with $A's$ asymmetric private key
$h = H\left(m\right)$	$m's$ one way hash function
$s_m = \mathbb{S}_A\left(m\right)$	$m's$ message signature with $A's$ asymmetric private key $[\mathbb{S}_A\left(m\right) \Leftrightarrow E_{A_{priv}}\left(H\left(m\right)\right)]$
$m_s = S_A\left(m\right)$	Signed message composed by the message m and its signature with $A's$ asymmetric private key $[S_A\left(m\right) \Leftrightarrow \left(m, \mathbb{S}_A\left(m\right)\right)]$
$b = V_A^?\left(m_s\right)$	Verify the signed message m_s with $A's$ asymmetric public key $[V_A^?\left(m_s\right) \Leftrightarrow \Big(H\left(m'\right) \overset{?}{=} D_{A_{publ}}\left(\mathbb{S}_A\left(m\right)\right)\Big)] \;/\; [m_s \equiv \left(m', \mathbb{S}_A\left(m\right)\right)]$
$z = NSK()$	Create new symmetric key z
$A = NAK()$	Create new asymmetric key pair for A

Table 1. Cryptographic protocol nomenclature

of the *holder* field as stated before. The attribute certificate is signed by the attribute authority, what means that the AA agrees on the terms expressed in the linked pseudonym structure. Therefore, the user should know the authorization policy and the conditions under which an attribute certificate request is granted.

It is supposed that the TTP will not reveal pseudonym links unless the condition stated in the certificate is fulfilled. It is also supposed that the user will not transfer his anonymous attribute certificate by revealing the corresponding private key to any other user. This is probably the weakest requirement in our solution and it needs a further study, as discussed later.

4.3 Protocol to obtain and use an attribute certificate

In this subsection we will explain the protocol to obtain an attribute certificate, and how it can be used. This protocol uses as fundamental construction block the fair blind signature scheme presented in [18] under the name of *fair blind signatures with registration.* Most of the structure of Parts I and II of our protocol correspond with the aforementioned protocol, but the nomenclature has been adapted, and an abstraction of the protocol has been used to masquerade the underlying mathematics. Additionally, in Part II, some steps have been introduced to adapt it to our scheme.

The cryptographic nomenclature used in the protocol is shown in Table 1. Actors involved in the protocol can be seen as follows:

- Actors and terminology
  - U is the user. His certified public key, U_{publ}, is supported by an external PKI.
  - N is a user's anonymous asymmetric key with no PKI support.

- P is a user's pseudonym. It has a public part P_{publ}, which is associated to the user, and a private part, P_{priv}.
- The *Trusted Third Party* [TTP] provides pseudonyms to users and keeps a link between both parts (public and private) of the pseudonym.
- The *Attribute Authority* [AA] provides attribute certificates, and its certified public key AA_{publ} is supported by an external PKI.
- The *Attribute subAuthorities* [AA^i / $\forall\ i \in$ *Attributes*] verify that a user fulfills the requirements needed to apply for an attribute certificate on $ATTR^i$. Their certified public key ${AA^i}_{publ}$ are supported by an external PKI.
- $ATTR^i$ is the attribute for which the *Attribute subAuthority* [AA^i / $\forall\ i \in$ *Attributes*] checks for requirement fulfillment, and for which the *Attribute Authority* [AA] provides attribute certificates.
- $ATTR^i_U$ is the proof that the user U fulfills the requirements to apply for the attribute i.
- SP is a *Service Provider* that offers services to those users that have the attribute certificate $ATTR^i$.
- f_{publ} and f_{priv} are two flags that specify which part of the pseudonym is public and which one is private.
- *val_period* is the period in which the pseudonym remains valid.
- $\mathit{fblind}_X\ (m)$ represents that the message m is protected to be "fair blind" signed by X.
- $S_X^{P_{publ}}\ (\mathit{fblind}_X\ (m))$ is the fair blind signature of X over message m under the public pseudonym P_{publ}, as specified in [18].
- $S_X^{P_{priv}}\ (m)$ is the fair blind signature of X over message m under the private pseudonym P_{priv}, after transforming the public blind signature to the corresponding private clear form. It is composed by the message and the fair blind signature under P_{priv}.

The whole protocol is divided into the following parts:

Part I. Obtaining a pseudonym. This part corresponds with the registration phase in the *fair blind signatures with registration* protocol from [18]. It deals with the user's acquisition of a pseudonym. The user will request a pseudonym from the TTP that is able to produce valid pseudonyms. This TTP must be recognized by the entity that issues the attribute certificates. This TTP will create a new pseudonym, which consists of two parts, the public and the private parts, respectively. Both parts must be created in a related way that makes possible the fair blind signature. The TTP will store and keep such a linked pair, so that the relation could be disclosed if some conditions are met. Then, both parts will be signed (with a flag identifying its purpose and its validity period) and sent to the user who requested them. This part of the protocol is achieved in an anonymous way and the TTP does not know anything about the user who requests a pseudonym. This part will be run whenever a user needs a new pseudonym.

1. $U : z = NSK()$
2. $U \rightarrow TTP : E_{TTP}\ (z, \mathit{Pseudonym\_Request})$
3. $TTP : \mathit{New\_Pseudonym}\ (P_{publ}, P_{priv})$
4. $TTP : STORE\ (\mathit{val\_period}, P_{publ} \leftrightarrow P_{priv})$
5. $TTP \rightarrow U : E_z\ (S_{TTP}\ (f_{publ}, \mathit{val\_period}, P_{publ}), S_{TTP}\ (f_{priv}, \mathit{val\_period}, P_{priv}))$

Part II. Obtaining a fair blind signature. This part of the protocol corresponds with the phase of getting a signature in the *fair blind signatures with registration* protocol from [18]. In this phase, the user obtains a message signed by the Attribute subAuthority $[AA^i]$ in charge of verifying fulfillment of the requirements needed to get a certificate over the attribute i. The way in which the fair blind signature operates guarantees that the signer is unable to know what he is signing, and that the signature is done over a public pseudonym related with the user, but such a relationship will be removed by transforming the signature over the private pseudonym.

Therefore, in this phase, the goal of the user is to obtain a proof that reveals that its owner fulfills the requirements needed to get an attribute certificate on a specific attribute. However, nobody must be able to link such a proof with the user.

In our protocol, the proof that a user fulfills a set of requirements consists of a public key signed by the authority in charge of verifying such requirements. The owner of the signed public key remains anonymous; that is, nobody is able to establish a relationship with the user that created it. However, the signature has a link with a private pseudonym, but nobody knows who the owner is. At the moment of issuing the fair blind signature the authority operates over a public pseudonym that is able to relate with the user's identity.

Thus, in the second part of the protocol the user creates a new asymmetric key pair (this key pair will be associated with the attribute certificate). He prepares such a public key to be fair blind-signed by the authority in charge and sends it together with information about the TTP, his public pseudonym and the set of proofs that show that the user fulfills the needed requirements.

The authority checks that the pseudonym is valid and that the TTP is recognized, and then checks if the user fulfills the requirements needed in order to get a certificate containing the attribute i. These requirements depend on the entity's policy.

If the requirements are met, this information is stored for its later use and the public key will be fair blind signed over the public pseudonym. Once the user gets that signature, he transforms it into a clear signature of the public key over the private pseudonym.

1. $U : N = NAK()$
2. $U \rightarrow AA^i : S_U\left(TTP, S_{TTP}\left(f_{publ}, val\_period, P_{publ}\right), ATTR_U^i, fblind_{AA^i}\left(N_{publ}\right)\right)$
3. $AA^i :$ `IF` $\left(\neg V_{TTP}^{?}\left(S_{TTP}\left(f_{publ}, val\_period, P_{publ}\right)\right)\right.$
   $\left.\vee \neg fulfill\_req\left(U, TTP, P_{publ}, ATTR_U^i\right)\right)$ `THEN Abort`
4. $AA^i : STORE\left(U \leftrightarrow ATTR_U^i \leftrightarrow TTP \leftrightarrow S_{TTP}\left(f_{publ}, val\_period, P_{publ}\right)\right)$
5. $AA^i \rightarrow U : S_{AA^i}^{P_{publ}}\left(fblind_{AA^i}\left(N_{publ}\right)\right)$
6. $U : S_{AA^i}^{P_{priv}}\left(N_{publ}\right)$

Part III. Obtaining a conditionally traceable attribute certificate. In this part of the protocol, the user will use the anonymous proof obtained in the previous part in order to apply for a standard attribute certificate. Thus, the user creates a structure to hold the information about his pseudonym and signs it to state that such information is correct and that the owner (the one who knows the private key associated with the public key) agrees on the terms expressed in such a structure.

At that moment, the user sends the proof obtained in the previous part, that is, the fair blind signature of the public key linked with the private pseudonym, the proof that the private pseudonym is valid, and the structure previously created.

The AA will verify every signature and will check the terms expressed in such a structure, specially in the condition under which the user's real identity will be revealed. Therefore, provided that the terms are signed by the holder and by the authority, the TTP will reveal the link between the private pseudonym and the public one whenever the attribute certificate is presented to the TTP and *condition* is verified. Additionally, the AA will reveal the link between the public pseudonym and the user's identity.

When everything works correctly, the AA creates an attribute certificate for a validity period stating that the holder of the related structure possesses such a specified attribute, and sends it to the user. The holder of such a structure is the one who knows the private key associated with the public key in it.

1. $U : \mathit{Pseud\_Inf} = S_N\left(\mathit{Label}_{PI}, P_{priv}, \mathit{TTP}, \mathit{Cond}, \mathit{Sig\_Alg}, N_{publ}\right)$
2. $U \rightarrow AA : \Big(S_{TTP}\left(f_{priv}, \mathit{val\_period}, P_{priv}\right), S_{AA^i}^{P_{priv}}\left(N_{publ}\right), \mathit{Pseud\_Inf}\Big)$
3. $AA : \texttt{IF} \Big(\neg V_{TTP}^{?}\left(S_{TTP}\left(f_{priv}, \mathit{val\_period}, P_{priv}\right)\right) \vee \neg V_{AA^i}^{?}\Big(S_{AA^i}^{P_{priv}}\left(N_{publ}\right)\Big)$
   $\vee \neg V_N^{?}\left(\mathit{Pseud\_Inf}\right) \vee \left(\neg \mathit{Agree\_on}\left(\mathit{Cond}\right)\right)\Big)$ `THEN Abort`
4. $AA : \mathit{Attr\_Cert} = S_{AA}\big(\mathit{Vers}, \mathit{Serial}, \mathit{Sig\_Alg}, AA, \mathit{Val\_Period}, H\left(\mathit{Pseud\_Inf}\right), \mathit{ATTR}^i\big)$
5. $AA \rightarrow U : \mathit{Attr\_Cert}$

Part IV. Using a conditionally traceable attribute certificate. In this part we show how the attribute certificate obtained in the previous part can be used. A user will send his anonymous attribute certificate plus the pseudonym information associated to any service provider, *SP*. This will verify that such a message is correct and that the certified attribute is enough to access to the service, sending a request to the anonymous user for the signature of a challenge in order to prove ownership. If the challenge is correctly signed then the service is granted to the user.

1. $U \rightarrow SP : \left(\mathit{Attr\_Cert}, \mathit{Pseud\_Inf}\right)$
2. $SP : \texttt{IF} \Bigg(\neg V_{AA}^{?}\left(\mathit{Attr\_Cert}\right) \vee \Bigg(H\left(\mathit{Pseud\_Inf}\right) \overset{?}{\neq} \mathit{Holder\_Field}\left(\mathit{Attr\_Cert}\right)\Bigg)$
   $\vee \neg \mathit{fulfill\_req}\left(\mathit{Service}, \mathit{Attr\_Cert}, \mathit{Pseud\_Inf}\right)\Bigg)$ `THEN Abort`
3. $SP \rightarrow U : \mathit{challenge}$
4. $U \rightarrow SP : S_N\left(\mathit{challenge}\right)$
5. $SP : \texttt{IF} \left(\neg V_N^{?}\left(S_N\left(\mathit{challenge}\right)\right)\right)$ `THEN Abort`
6. $SP \rightarrow U : \mathit{Service\_granted}$

If the user misuses his privileges obtained through an anonymous attribute certificate, then the service provider will collect all the proofs of that misuse, and will send them to the AA and the TTP requesting the revocation of the attribute certificate and revealing the user's identity (for an eventual prosecution). In these cases, it could be interesting that the challenge includes a timestamp and the transaction identification besides the random bits, in order to prove misuses where time is important.

5 Discussion and future work

New applications, particularly in the area of e-commerce, need an authorization service to describe privileges to perform tasks. Traditional authorization solutions are not very helpful for many of the Internet applications; however, attribute certificates proposed by ITU-T are well suited to solve this problem. On the other hand, during last years, users have paid special attention to the problem caused by the fact that many of the operations and transactions they carry out through the Internet can be easily recorded and collected. Thus, anonymity has become a desirable feature to be added in many cases.

We have presented a first approach to extend X.509 attribute certificates with anonymity capabilities, as well as a protocol to obtain certificates preserving user's anonymity by using a fair blind signature scheme.

The approach could be improved and adapted depending on the different scenarios where to be applied. We explain now how several improvements can be added to our scheme in order to have a better behavior.

In some applications when a user applies for a certificate, the system should provide a receipt of such a request in order to guarantee that the system will process it appropriately. Whenever the system replies to that request, it should get a receipt in order to prove that its duty was achieved properly. In those systems a fair non-repudiation scheme [20] should be used.

Moreover, in order to improve the user's anonymity, an anonymous communication channel (such as a mixnet [4]) could be used in part I, III and IV of the protocol to masquerade the originator IP address. This scheme should be used in systems where user's anonymity is the most important requirement to the system and user's identity could be guessed using the IP address of the message originator.

In order to avoid the possibility that organizations create anonymous user profiles, a user can run the protocol several times to get the same attribute certificate under a different pseudonym. However, it would be interesting to get a pseudonym with one public part and many private ones, in such a way that it would be only necessary to re-run part III of the protocol in order to get the attribute certificate under different pseudonyms (all related to the same public part).

The solution that we propose in tgis work does not solve all problems that could arise in a multi-purpose anonymous attribute system. We believe that the main drawbacks in our actual solution are:

- The user's identity in part II of the protocols could be linked with the private pseudonym in part III of the protocols if, during the protocol run, such a user is the only one who has an unfinished open request and the AA colludes with the Attribute subAuthority. Thus, the interleaving of user's requests between part II and part III is very important in our protocol.

- Actual version of the protocol does not avoid that the anonymous user U_1 transfers the use of his anonymous attribute certificate to another anonymous user U_2 just by letting U_2 know the associate private key. U_1 and U_2 would share in this way the use and advantages of possessing that attribute, even if U_2 does not posses it. This is, of course, one of the most important areas where we will focus our further research.

References

1. J. Benaloh and D. Tuinstra. Receipt free secret-ballot elections. In *Proc. of 26th Symp. on Theory of Computing (STOC'94)*, pages 544–553, New York, 1994.
2. J. Camenisch and E. V. Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proc. of 9th ACM Conference on Computer and Communications Security (CCS)*, Washington D.C., Nov. 2002. ACM, Academic Press.
3. J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer-Verlag, 2001.
4. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, Feb. 1981.
5. D. Chaum. Blind signatures for untraceable payments. In D. Chaum, R. Rivest, and A. Sherman, editors, *Advances in Cryptology–Crypto'82*, pages 199–203, Santa Barbara, CA USA, Aug. 1983. Plenum Press.
6. D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, Oct. 1985.
7. D. Chaum and J. H. Evertse. A secure and privacy-protecting protocol for transmitting personal information between organizations. In A. M. Odlyzko, editor, *Advances in Cryptology - Crypto '86*, volume 263 of *Lecture Notes in Computer Science*, pages 118–170, Berlin, 1986. Springer-Verlag.
8. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash (extended abstract). In *Advances in Cryptology–Crypto'88*, pages 319–327, 1989.
9. L. Chen. Access with pseudonyms. In E. Dawson and J. Golic, editors, *Cryptography: Policy and Algorithms*, volume 1029 of *Lecture Notes in Computer Science*, pages 232–243. Springer-Verlag, 1995.
10. L. Cranor and R. Cytron. Sensus: A security-conscious electronic polling system for the internet. In *Proceedings of the Hawaii International Conference on System Sciences*, Wailea, Hawaii, 1997.
11. C.-I. Fan and C.-L. Lei. A user efficient fair blind signature scheme for untraceable electronic cash. *Information Science and Engineering*, 18(1):47–58, Jan. 2002.
12. A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In J. Seberry and Y. Zheng, editors, *ASIACRYPT'92*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer-Verlag, 1992.
13. ITU-T Recommendation X.509. Information technology - open systems interconnection - the directory: Authentication framework. June 1997.

14. ITU-T Recommendation X.509. Information technology - open systems interconnection - the directory: Public-key and attribute certificate frameworks. Mar. 2000.
15. A. Lysyanskaya. Pseudonym systems. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, June 1999.
16. A. Lysyanskaya. *Signature Schemes and Applications to Cryptographic Protocol Design*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Sept. 2002.
17. S. Farrel and R. Housley. An Internet attribute certificates profile for authorization. Request for Comments 3281. Nework Working Group. Internet Engineering Task Force. April 2002.
18. M. A. Stadler, J. M. Piveteau, and J. L. Camenisch. Fair blind signatures. In L. C. Guillou and J. J. Quisquater, editors, *Advances in Cryptology–EUROCRYPT'95*, volume 921 of *Lecture Notes in Computer Science*, pages 209–219. Springer-Verlag, 1995.
19. S. von Solms and D. Naccache. On blind signatures and perfect crimes. *Computers & Security*, 11:581–583, 1992.
20. J. Zhou. Achieving fair nonrepudiation in electronic transactions. *Journal of Organizational Computing and Electronic Commerce*, 11(4):253–267, 2001.