A VOMS Attribute Certificate Profile for Authorization

Vincenzo Ciaschini

December 20, 2005

Contents

1 Introduction

X.509 Attribute Certificates (ACs) [1] are used to bind a set of attributes, like group membership, role, security clearance, etc... with an AC holder. Their well-defined, standardized format and easy extensibility make them a premium way to distribute those informations in large system, and in particular in environments where authentication is done via X.509 Certificates [2]. This is the reason why ACs are the format chosen by the VOMS server [5] to encode authorization data.

However, the reference documantation about ACs leaves a huge amount of freedom regarding exactly how ACs should be encoded. The scope of this paper is to document the particular vernacular of ACs used by VOMS, and how the data they contain is supposed to be encoded. This format is in any case fully compatible with what described in [1], and should any incompatibility be found between what is described here and what is described in [1], the latter is the authoritative source.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [3].

2 FQAN

The FQAN (short form for Fully Qualified Attribute Name) is what VOMS ACs use in place of the Group/Role attributes. It is better described in [4], although a brief summary will be given in the following paragraphs. It has been developed because of two perceived problems with the standard-defined[1] Group and Role attributes:

1. The Group and Role attributes are completely independent of each other; in particular, Roles are meant to be global, associated directly to the AC holder, regarless of group membership. On the other hand, besides this behaviour VOMS also allows groups and roles to be bound together, using one as a qualifier of the other. While it is indeed possible to encode groups and roles inside the standard attributes in a format that could represent this information, there is no way to have the same format also be readable by other AC users without risking misunderstandings.

2. Also, practical use of group/role attributes in defining ACLs has showed that having them separate is inconvenient, and it is much simpler to have them all expressed together.

For these reasons, a new format has been devised, as documented in [4]. However, here follows a copy of the relevant informations.

Group membership, Role holding and Capabilities may be expressed in a format that bounds them together in the following way:

<group name>/Role=[<role name>][/Capability=<capability name>]

where the elements between [] are optional.

This format specifies that the AC holder is a member of group <group name>, and in this group he holds the role <role name> while having the capability <capability name>.

<group name>, <role name> and <capability name> are described by the following grammar:

```
group name       ::= entity
                  |  groupname ‘‘/’’ entity
role name        ::= entity
capability name ::= entity

entity           ::= [a-zA-Z0-9 _]*
```

It can be noted that while role and capability names have a flat structure, group name can be expressed as a series of identifiers separated by the “/” character. This happens because groups are a structured entities, where a group can have subgroups, that can have subgroups, ad libitum. They are represented in the same format as Unix path names, where the first directory name corresponds to the VO name, the second one to a group, the third one to a subgroup of the preceding group, etc...

3 VOMS Attribute Certificate Profile

This is the general format of an AC as defined by [1]. Customizations used by VOMS will be discussed in individual subsections. Everything not specifically mentioned here is intended to be in accordance with [1].

```
AttributeCertificate ::= SEQUENCE {
  acinfo              AttributeCertificateInfo,
  signatureAlgorithm  AlgorithmIdentifier,
  signatureValue      BIT STRING
}

AttributeCertificateInfo ::= SEQUENCE {
  verson                  AttCertVersion,
  holder                  Holder,
  issuer                  AttCertIssuer,
  signature               AlgorithmIdentifier,
  serialNumber            CertificateSerialNumber,
  attrCertValidityPeriod AttCertValidityPeriod,
  attributes              SEQUENCE OF Attribute,
  issuerUniqueID          UniqueIdentifier OPTIONAL,
  extensions              Extensions OPTIONAL
}
```

```
AttCertVersion ::= INTEGER { v2(1) }

Holder ::= SEQUENCE {
  baseCertificateID         [0] IssuerSerial OPTIONAL,
}

AttCertIssuer ::= CHOICE {
  v2Form   [0] V2Form
}

V2Form ::= SEQUENCE {
  issuerName           GeneralNames  OPTIONAL,
  baseCertificateID     [0] IssuerSerial  OPTIONAL,
  objectDigestInfo      [1] ObjectDigestInfo  OPTIONAL
}

IssuerSerial  ::=  SEQUENCE {
  issuer         GeneralNames,
  serial         CertificateSerialNumber,
  issuerUID      UniqueIdentifier OPTIONAL
}

AttCertValidityPeriod  ::= SEQUENCE {
  notBeforeTime  GeneralizedTime,
  notAfterTime   GeneralizedTime
}

Attribute ::= SEQUENCE {
  type      AttributeType,
  values    SET OF AttributeValue
  -- at least one value is required
}

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType
```

3.1 Holder

The holder of a VOMS AC MUST always be an X.509 PKC. As a consequence of this, in VOMS ACs the only admissible choice for the field is the baseCertificateID, hence the absence in the above decription, of the other two choices from this SEQUENCE. The issuerUID field in this case MUST be present if and only if it is also present in the holder's PKC, and in this case they MUST have the same value. Note that [2] says that conforming implementations of PKCs SHOULD NOT use this field, but that implementations SHOULD be capable to handle it.

3.2 AttCertIssuer

The AttCertIssuer field MUST always be encoded using the V2Form data format.

3.3 V2Form

Conforming ACs MUST NOT use either the baseCertificateID or the objectDigestInfo fields. They MUST use the issuerName field, which MUST contain one and only one distinguished name belonging to the certificate that the AC issuer will use to sign the AC. This in particular means that this subject MUST NOT be empty.

4 Attributes

The attributes field contains informations about the AC holder. At least one attribute MUST always be present.

Attributes types use the format defined in [1], repeated here for convenience:

```
IetfAttrSyntax ::= SEQUENCE {
  policyAuthority [0] GeneralNames    OPTIONAL,
  values          SEQUENCE OF CHOICE {
    octets    OCTET STRING,
    oid       OBJECT IDENTIFIER,
    string    UTF8String
  }
}
```

The attributes Group and Role, defined in [1] are not used by VOMS AC, and SHOULD NOT be present in conforming ACs. Instead, it defines a new attribute, FQAN, which holds informations about both, and in fact also binds them together.

```
name         : voms-attribute
OID          : { voms 4 }
syntax       : IetfAttrSyntax
values       : Multiple allowed
```

where “voms” is the OID 1.3.6.1.5.3004.100.100 and has been registered for VOMS.

The policyAuthority field of the IetfAttrSyntax MUST contain an encoding of both the VO to which the AC issuer belongs and the server which generated this particular attribute, in the following format:

<vo name>://<fqhn>:<port>

all of this component should be omitted, and the IA5STRING choice of the GeneralName type should be used.

On the same way, the octets choice of the values field shoud be used to encode the FQANs.

4.1 Extensions

In the current version, only a specific subset of the extensions specified in [1] is used and they are decribed here, along with any specifics points that were originally only loosely defined. A VOMS-compliant AC is allowed to use extensions other than those indicated here, on the condition that they should not be critical.

4.1.1 AC Target

This extension MAY be present. If it is present, then then targetName option MUST be used, with the FQDNs of the hosts which the AC is targeted to. Compliant implementation MUST honor this extension. Also, they MUST be capable of understnading at least the targetName option.

4.1.2 No Revocation Available

This extension MUST be used in the current version of VOMS ACs.

4.2 Attributes

While in principle any attribute may be used here, this section will specify what attributes are included in the current version of ACs and which are expected to be recognized by conforming implementations.

4.2.1 Fully Qualified Attribute Name (FQAN)

This attribute is used to express user membership in groups and ownership of roles in an integrated way that makes easier to express relations between the two elements. It is fully documented in [FQAN], and MUST be included in any and all VOMS ACs.

4.2.2 Group and Role

This two attributes are not used in current version, but they MAY be present. However, in this case their content should be consistent with the content of the FQAN attribute. The suggested way to ensure this is the following:

1. Role and Group have the same number of elements as FQAN.

2. If the n-th element of FQAN denote membership in group G and ownership of role R, then those are the values of the n-th Group and the n-th Role.

> If no role R is specified in an element of the FQAN attribute, then the corresponding element in the Role attribute is the empty string.

Conforming implementations MAY recognize this two attributes, but if they do they SHOULD the verify correspondence between their values and the content of the FQAN attributes. Should there be a miscrepancy, the normative data should be that included in the FQAN element. It is up to the implementation whether to consider a discrepancy enough cause for an error or to settle for a warning.

5 Attribute Certificate Validation

All mechanisms described by [1] are kept as they are with only the following change:

> It is not required at any time during signature verification that:
>
> - The AC issuer certificate has the signing bit set, or that
> - any proxy certificate or user certificate as the signing bit set.
>
> It is although preferred for AC issuer certificate that the signing bit is set.

References

[1] S. Farrell, R. Housley, RFC 3281: An Internet Attribute Certificate Profile for Authorization.

[2] R. Housley, W. Polk, W. Ford, D. Solo, RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.

[3] S. Bradner, RFC 2119: Key words for use in RFCs to Indicate Requirement Levels.

[4] V. Ciaschini, A. Frohner, Voms Credential Format, http://edg-wp2.web.cern.ch/edg-wp2/security/voms/edg-voms-credential.pdf

[5] R. Alfieri, R. Cecchini, V. Ciaschini, L. Dell'Agnello, A. Frohner, A. Gianoli, L. Karoly, F. Spataro, An Authorization System for Virtual Organizations, Forthcoming in Proceedings of the 1st European Across Grids Conference.