ALLSEEN ALLIANCE

Certificate layout proposal

Policy Certificate

The policy certificate does not need to be transferred from the holder to a peer.

- Revocation is not required
- ValidityFrom and ValidatyTo is not needed, unless we want to install time based policy (could be an advanced feature)
- Delegate is not required
- No strict need for X509 layout, but reusing X509 makes sense
- Fields:
  - DN: CN=application GUID
  - validityFrom, valididityTo: map to validity sequence of X509
  - Issuer in x509 signature
  - Subject: part of SubjectPublicKeyInfo of X509
  - ECC curve, digest algorithm and DSA algorithm is part of X509
  - Basic constraints CA = false; (delegate)
- Not defined: version/type and digest (of authorization data)

Meeting comments: policy doesn't need to be in x509, but we will reuse the format. (type & digest see slide 7)

Membership certificate

- Provides proof of membership of a guild, transferred between peers
  - Revocation is required

–Fields:

- DN: CN=application GUID, OU= guild ID
- validityFrom, valididityTo: map to validity sequence of X509
- Issuer in x509 signature
- Subject: part of SubjectPublicKeyInfo of X509
- ECC curve, digest algorithm and DSA algorithm is part of X509
- Basic constraints CA = false or true; Pathlen = 0 or 1 (delegate)

– Not defined: version/type and digest (of authorization data)

Meeting notes: no comments, membership will be used as presented. (type & digest see slide 7)

User equivalence certificate

- Allows a single user to act(this includes generate certificates) on behalf of another user
  - Revocation is required

–Fields:

- DN: CN=application GUID
- validityFrom, valididityTo: map to validity sequence of X509
- Issuer in x509 signature
- Subject: part of SubjectPublicKeyInfo of X509
- ECC curve, digest algorithm and DSA algorithm is part of X509
- Basic constraints CA = true, path length = 1
  - User equivalence rights can NOT be transferred, but CA is true so it can generate membership certificates

– Not defined: version/type

Meeting notes: No comments, will implement as presented. (type see slide 7)

Identity certificate

- Links an identity to an application. Multiple apps can share the same identity

–Revocation is required

–Fields:

  - DN: CN=application GUID
  - validityFrom, valididityTo: map to validity sequence of X509
  - Issuer in x509 signature
  - Subject: part of SubjectPublicKeyInfo of X509
  - ECC curve, digest algorithm and DSA algorithm is part of X509
  - Basic constraints CA = false (delegate)
  - SubjectAltName: alias of the identity
    - –Applications share the same identity if the alt name and issuer are the same
    - –This alias could be GUID in order to decouple data from certificate

– Not defined: version/type and digest (of authorization data)

Meeting notes: will be used as presented (type & digest see slide 7)

– The alias is GUID. Identity certs with same GUID (from same issuer) represent the same owner. The authorization data contains the meta data describing the identity.

Guild equivalence certificate

- Allows to map membership certificates from other security managers to own.
  - This is part of the policy of an application
  - Propose not to use a separate guild equivalence certificate, but to express the guild equivalence rules as part of the policy authorization data.

Meeting notes: accepted as presented

Undefined fields

- Where to put the undefined fields?
  - Type/version
    - In the current HLD type and version are combined
      - Keep them together or split? Answer: split and only add the type
      - Implemented via a custom extension? Yes, and OID should be requested → http://pen.iana.org/pen/PenApplication.page
      - Custom certificate usage? No
  - Delegate flag
    - The delegate flag can be implemented by the basic constraint
      - It requires a single exception for user equivalence. The user equivalence rights cannot be delegated, but the certificate needs to create certificates
    - Use a custom extension to allow delegate user equivalence if needed?
    - Meeting conclusion:
      - Reuse the basic constraint section. No need for a custom extension
      - Should we use extended usage properties as defined by X509? This could limit the number of cases where certificates can be used. This does mean that we need to check those and dissallow certificates if they don't have rights

Authorization data signature

- Where to put the authorization data?
  - Linked with certificate (hash or data is part of the signed X509 structure)
    - Acceptable if this is stable data. We only want to revoke certificates when it is really needed.
      - Ok for manifest enforcing
      - Not ok for delegation (not only the certificate with delegation rights needs to be revoked, but all children of it as well)
  - Decoupled from membership certificate
    - Needs to be signed; is for some key; needs to be revoked
      - Is same as membership certificate + authorization data
      - In what format do we do this? X509 or something else?
        - » X509 Reuse of what we have; Ok for delegation; overkill for manifest enforcement (the membership certificate won't have added value.
        - » Something else: extra security related code → more chance for security issues
- Meeting notes:
  - Do not store the complete data in the certificate; only the hash
  - Decoupling would be beneficial for the delegation scenario, but the cost is to heavy (see next slide with details)
  - Adding permissions doesn't require to revoke the old certificate (reduce the number of certificate being revoked)
  - Removing permissions requires revocation (which will revoke all child certificates as well

Hybrid scenario

- For manifest enforcement

– hash in the certificate

- Delegation
  - Use 2 certificates: one for signing membership certificates and one with permissions (like manifest)
- Conclusion:
  - Good:
    - Clear split of where certificates are used for
    - Only one certificate needs to be revoked
  - Less good:
    - The delegated device requires 3 certificates: his membership, the signing certificate and the membership of the signer
      - Expensive when multiple guilds are involved
    - 2 CA involved → we need to send to OCSP requests
    - Complexity of the solution
    - Requires distribution of permission certificate to all peers with delegated membership certificate
- Meeting notes: The extra overhead is to costly compared to the benefits

Alternatives

- Can we express delegation rights in the policy?
  - Yes, this is the Guild equivalence feature
    - We only need to send 1 membership certificate, the certificate will be trusted based on policy info
  - But:
    - If I delegated rights to A and B, both A and B can talk to me, but A can't talk to B and vise versa. This kills the use case of Dad managing the home. In this case mom and son can to talk to each other, unless they befriend each other and manage their own interaction
- Trade-off: technical simpler solution vs feature?
  - The technical complexity is high, impact on the wire, …
  - The feature is a valuable
- Meeting notes: We need the delegation feature

Thank You

Follow Us On 

For more information on AllSeen Alliance, visit us at: allseenalliance.org & allseenalliance.org/news/blogs