OCF 1.0 Security CR - CR36 (1385/1303)

Legal Disclaimer

5.1.1 ACL Architecture (Informative)

As mentioned, an OIC Client device requests access to resources from an OIC Server. The OIC Server examines the OIC client's access rights to its resources based on either OIC client's identity (if SBAC) or role (RBAC). Access requests may be authorized based on group or device credentials. The ACL architecture illustrates four client devices seeking access to server resources. A server evaluates each request using local ACL policies and access manager services.

Each ACE contains the permission set that will be applied for a given resource requestor. Permissions consist of a combination of Create, Read, Update, Delete and Notify (CRUDN) actions. Requestors authenticate as a device and optionally operating with one or more roles. OIC devices may acquire elevated access permissions when asserting a role. For example, an ADMINISTRATOR role might expose additional resources and interfaces not normally accessible.

5.3.3 Credential provisioning

Several types of credential may be configured in a /oic/sec/cred resource. Currently, they include at least the following credential types; pairwise symmetric keys, group symmetric keys, certificates, asymmetric keys and signed asymmetric keys. Keys may be provisioned by a credential management service (e.g. "oic.sec.cms") or dynamically using a Diffie-Hellman key agreement protocol or through other means.

The following describe an example on how a device can update a PSK for a secure connection. A device may discover the need to update credentials, e.g. because a secure connection attempt fails. The device will then need to request credential update from a credential management service. The device may enter credential-provisioning mode (e.g. /oic/sec/pstat.Cm=16) and may configure operational mode (e.g. /oic/sec/pstat.Om="1") to request an update to its credential resource. The CMS responds with a new pairwise pre-shared key (PSK).

5.3.4 Role assignment and provisioning

The OIC servers, receiving requests for resources they host, need to examine the role asserted by the entity requesting the resource (OIC client) and compare that role with the constraints described in their ACLs. Thus, an OIC client device may need to be provisioned with one or more role credentials.

Each OIC device holds the role information as a property within the credential resource. Thus, it is possible that an OIC client, seeking a role provisioning, enters a mode where both its credentials and ACLs can be provisioned (if they are provisioned by the same sever!). The provisioning mode/status is typically indicated by the content of /oic/sec/pstat.

Once provisioned, the OIC client can assert the role it is using as described in Section 10.3.1, if it has a certificate role credential.

Alternatively, if the server has been provisioned with role information for a client, or the client has previously asserted roles to the server, the client can assert a specific role with the CoAP payload:

e.g., GET a/light?role=admin

The client has no way to know in advance what roles are provisioned on the server, and must attempt an action and observe the server's response. If the response is permission denied, the client learns that either the server is not provisioned with the role, or the ACLs are misconfigured. If no specific

role is specified in the CoAP payload, all provisioned roles are used in ACL enforcement. When a server has multiple roles provisioned for a client, access to a resource is granted if it would be granted under any of the roles.

5.5 Credential Overview

OIC Devices may use credentials to prove the identity and role(s) of the parties in bidirectional communication. Credentials can be symmetric or asymmetric. Each device stores secret and public (if applicable) parts of its own credentials, as well as credentials for other devices that have been provided by the On-boarding Tool or a Credential Management Service. These credentials are then used in the establishment of secure communication sessions (e.g., using DTLS) to validate the identities of the participating parties. Role credentials are used once an authenticated session is established, to assert one or more roles for a device.

9.4.2 Certificate Format

9.4.2.1.1 Supported Certificate Extensions

[Editorial note: This subsection is defined in another CR]

- Subject Alternative Name

  If the EKU extension is present, and has the value XXXXXX, indicating that this is a role certificate, the Subject Alternative Name (subjectAltName) extension shall be present and interpreted as described below. When no EKU is present, or has another value, the subjectAltName extension SHOULD be absent. The subjectAltName extension is used to encode one or more Role ID values in role certificates, binding the roles to the subject public key. The subjectAltName extension is defined in RFC 5280 (Section 4.2.1.6):

```
id-ce-subjectAltName OBJECT IDENTIFIER ::=  { id-ce 17 }

SubjectAltName ::= GeneralNames

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

GeneralName ::= CHOICE {
     otherName                       [0]     OtherName,
     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 }

EDIPartyName ::= SEQUENCE {
```

```
nameAssigner            [0]     DirectoryString OPTIONAL,
partyName               [1]     DirectoryString }
```

The sequence of `GeneralNames` will be a `SEQUENCE` of `EDIPartyName` objects. The `nameAssigner` field may be used to specify the authority that defined the semantics of the role. If nameAssigner is unspecified, the certificate issuer has defined the role. The role ID is encoded in the `partyName` field. Other `GeneralName` types in the SEQUENCE may be present, but shall not be interpreted as roles. Therefore, if the certificate issuer includes non-role names in the `subjectAltName` extension, the extension should not be marked critical.

Note that the role, and authority need to be encoded as ASN.1 PrintableString type, the restricted character set [0-9a-z-A-z '()+,-./:=?].

9.4.5 Certificate Provisioning

The credential management service (e.g. a hub or a smart phone) issues certificates and private keys for new devices. The credential management service shall have its own certificate and private key pair. The certificate is either self-signed if it acts as Root CA or signed by the upper CA in its trust hierarchy if it acts as Sub CA. In either case, the certificate shall have the format described in Section 8.4.2.

The CA in the credential management service shall generate a device's certificate signed by this CA's private key, a paired private key, and then the credential management service transfers them to the device including its CA certificate chain. Optionally, the CMS may also transfer one or more role certificates, which shall have the format described in Section 9.4.2. The subjectPublicKey of each role certificate shall match the subjectPublicKey in the device certificate.

The sequence flow of a certificate transfer for a Client-directed model is described in Figure 17.

1. The credential management service retrieves information of the device that request a certificate.

2. The credential management service shall transfer the issued certificate, CA chain and private key to the designated device. The credential type (oic.sec.cred) used to transfer certificates in Figure 17 is also used to transfer role certificates, by including multiple credentials in the POST from CMS to Device.

z

Figure 17 – Client-directed Certificate Transfer

10 Device Authentication

When accessing a restricted resource on an OIC Server, the Server shall authenticate the OIC Client requesting the access. OIC Clients shall authenticate OIC Servers while requesting access. OIC

Clients may also assert one or more roles that the server can use in access control decisions. Roles may be asserted when the device authentication is done with certificates.

10.3 Device Authentication with Certificates

When using certificates to authenticate, the client and server shall each include their certificate chain, as stored in the appropriate credential, as part of the selected authentication cipher suite. Each device shall validate the certificate chain presented by the peer device. Each certificate signature shall be verified until a public key or its hash is found within the /oic/sec/cred resource. Credential resources found in /oic/sec/cred are used to terminate certificate path validation.

Note: Certificate revocation mechanisms are currently out of scope of this version of the specification.

10.3.1 Role Assertion with Certificates

This section describes role assertion by a client to a server using a certificate role credential. If a server does not support the certificate credential type, clients should not attempt to assert roles with certificates.

Following authentication with a certificate, a device may assert one or more roles. The role credentials must be certificate credentials and shall include a certificate chain. The server shall validate each certificate chain as specified in Section 10.3. Additionally, the public key in the end-entity certificate used for device authentication must be identical to the public key in all role (end-entity) certificates. Also, the subject distinguished name in the end-entity authentication and role certificates must match. The roles asserted are encoded in the subjectAltName extension in the certificate. Note that the subjectAltName field can have multiple values, allowing a single certificate to encode multiple roles that apply to the client. The server must also check that the EKU extension of the role certificate(s) contains the value XXX (see Section YYY) indicating the certificate may be used to assert roles. Figure 1XXX describes how a client device asserts roles to a server.

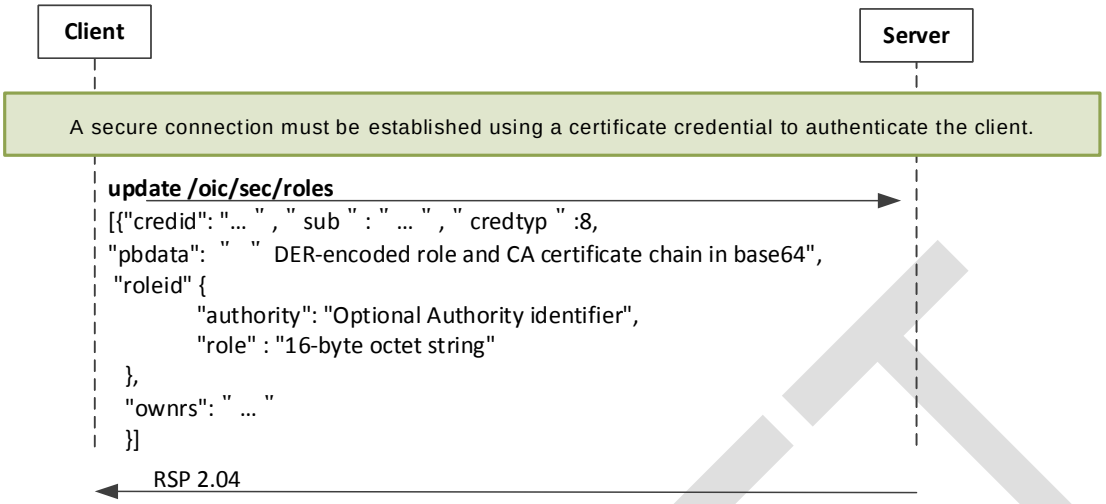

Figure 1XXX Asserting a role with a certificate role credential.

Figure 1XXX Notes

1. The response shall contain “204 No Content” to indicate success or 4xx to indicate an error. If the server does not support certificate credentials, it should return “501 Not Implemented”

2. ~~The client must post to /oic/sec/cred/<publicKey>/roles where <publicKey> is the public key from the client’s role certificate. The server must validate the client is asserting roles for the correct public key. The publicKey is encoded as the hex hash of the DER encoding of the public key, using SHA-256.~~

3. Roles asserted by the client may be kept for a duration chosen by the server. The duration shall not exceed the validity period of the role certificate. When fresh CRL information is obtained, the certificates in /oic/sec/cred/*/roles should be checked, and the role removed if the certificate is revoked or expired.

   Servers should choose a nonzero duration to avoid the cost of frequent re-assertion of a role by a client. It is recommended that servers use the validity period of the certificate as a duration, effectively allowing the CMS to decide the duration.

4. The format of the data sent in the create call shall be a list of credentials (oic.sec.cred, see Table 13). They shall have credtype 8 (indicating certificates) and PrivateData field shall not be present. For fields that are duplicated in the oic.sec.cred object and the certificate, the value in the certificate shall be used for validation. For example, if the Period field is set in the credential, the server must treat the validity period in the certificate as authoritative. Similar for the roleid data (authority, role).

5. Certificates shall be encoded as in Figure 17 (DER-encoded certificate chain in base64)

6. Clients may GET the /oic/sec/roles resource to determine the roles that have been previously asserted. An array of credential objects must be returned, or “204 No Content” to indicate that no previously asserted roles are currently valid.

13.2 Credential Resource(/oic/sec/cred)

Property Title	Property Name	Value Type	Value Rule	Unit	Access Mode	Mandatory	Description
Authority	authority	String	-	-	R	No	A name for the authority that defined the role. If not present, the credential issuer defined the role. If present, must be expressible as an ASN.1 PrintableString.
Role	role	String	-	-	R	Yes	An identifier for the role. Must be expressible as an ASN.1 PrintableString.

Figure 2 -- Definition of the oic.sec.cred.roletype Property.

13.5.2 ACL Resource

The /oic/sec/acl resource contains access control list entries governing access to OIC Server hosted resources.

Local ACL resources supply policy to a resource access enforcement point within an OIC stack instance. The OIC framework gates OIC client access to OIC server resources. It evaluates the subject's request using policy in the ACL.

Resources named in the ACL policy should be fully qualified or partially qualified. Fully qualified resource references should include the device identifier of a remote device hosting the resources. Partially qualified references imply the local resource server is hosting the resource. If a fully qualified resource reference is given, the intermediary enforcing access shall have a secure channel to the resource server and the resource server shall verify the intermediary is authorized to act on its behalf as a resource access enforcement point.

Resource servers should include references to device and ACL resources where access enforcement is to be applied. However, access enforcement logic shall not depend on these references for access control processing as access to server resources will have already been granted.

Local ACL resources identify a Resource Owner service that is authorized to instantiate and modify this resource. This prevents non-terminating dependency on some other ACL resource. Nevertheless, it should be desirable to grant access rights to ACL resources using an ACL resource.

An ACE is called currently valid if the validity period of the ACE includes the time of the request. Note that the validity period in the ACE may be a recurring time period (e.g., daily from 1:00-2:00). Matching the resource(s) specified in a request to the resource property of the ACE is defined in Section 12.2. For example, one way they can match is if the Resource URI in the request exactly matches one of the resources in the ACE.

A request will match an ACE if any of the following are true:

1. The deviceuuid associated with the secure session matches the "subjectuuid" of the ACE; AND the resource of the request matches one of the "resources" of the ACE; AND the ACE is currently valid.
2. The ACE "subjectuuid" contains the wildcard "*" character; AND the resource of the request matches one of the "resources" of the ACE; AND the ACE is currently valid.
3. When client authentication uses a certificate credential
    a. One of the roleid values associated with the public key used in the secure session (from /oic/sec/roles), matches the "roleid" of the ACE; AND the resource of the request matches one of the "resources" of the ACE; AND the ACE is currently valid.
    b. The CoAP payload of the request specifies a role, which is associated with the client public key (in /oic/sec/roles), and the specified role matches the "roleid" of the ACE; AND the resource of the request matches one of the "resources" of the ACE; AND the ACE is currently valid.
4. When authentication uses a symmetric key credential
    a. One of the roleid values associated with the symmetric key credential, matches the "roleid" of the ACE; AND the resource of the request matches one of the "resources" of the ACE; AND the ACE is currently valid.
    b. The CoAP payload of the request specifies a role, which is associated with the symmetric key credential, and the specified role matches the "roleid" of the ACE; AND the resource of the request matches one of the "resources" of the ACE; AND the ACE is currently valid.

A request is granted if ANY of the 'matching' ACEs contains the permission to allow the request. Otherwise, the request is denied.

Note that there is no way for an ACE to explicitly deny permission to a resource. Therefore, if one device with a given role should have slightly different permissions than another device with the same role, they must be provisioned with different roles

13.X Roles resource (/oic/sec/roles)

The roles resource maintains roles that have been asserted with role certificates, as described in Section 10.3.1. Asserted roles have an associated public key, i.e., the public key in the role certificate. Clients may only access roles associated with their public key.

In more detail, the Retrieve, Update and Delete operations on the Roles resource should behave as follows. Unlisted operations are implementation specific and not reliable. Note that this description is editorial, and the RAML provides the normative and formal behaviour description.

1. Retrieve shall return all previously asserted roles associated with the client's public key. Note that the public key is always available to the server as part of the secure channel information. Retrieve with query parameters is not supported.

2. Update includes the "roles" array property and distinct roles in this array are added to the resource. This is also scoped to the client's public key. Two roles are distinct if either of the "role" or "authority" properties differ.

3. Delete shall remove the entire "roles" array for the client's public key.

Fixed URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/oic/sec/roles	Roles	urn:oic.r.role	baseline	Resource containing roles that have previously been asserted to this server	Security

Table XX – Definition of the oic.r.roles Resource

Property Title	Property Name	Value Type	Value Rule	Unit	Access Mode	Mandatory	Description
Roles	roles	oic.sec.role	array	-	RW	Yes	List of roles previously asserted to this server

Table XX – Properties of the oic.r.roles Resource