Pub-Sub Profile for Authentication and Authorization for Constrained Environments (ACE)

draft-ietf-ace-pubsub-profile-08

Francesca Palombini
Cigdem Sengul
Marco Tiloca

IETF 118
November  2023
The document defines a way to authorize pub-sub clients and protect group communication using the ACE framework

- Focuses on securing group communications for CoAP pub-sub clients (may support MQTT)
  - Relies on transport or application layer security profiles of ACE (e.g. DTLS [RFC9202] or OSCORE [RFC9203]) to achieve communication security, server authentication and proof-of-possession for Access Tokens
- Message exchanges/message formats and processing follow the specifications for provisioning and renewing keying material in key-groupcomm
  - Current description satisfies most key-groupcomm requirements (30) / and options (15)
Version -07 to -08

• Revised the scope format.
• Revised Join Request-Response exchange.
  • The 'cnonce' parameter must be present in the Join Request.
  • The 'kid' of the group key is used as Group Identifier.
  • Relaxed inclusion of the 'peer_roles' parameter.
• More detailed description of the encryption and signing operations.
  • Defined construction of the AEAD nonce.
• Clarifications and editorial improvements.
Authorisation Flow

Workflow mostly complete

Further discussion needed: How to support topic resource discovery besides .well-known/core
Authorisation Request & Scope

- The Client sends **two Authorisation Requests** to the AS for **two audiences**: the Broker and the KDC.

- Audience (required) and Scope (optional) and MUST be of data model AIF-PUBSUB-GROUPCOMM
  - The object identifier ("Toid") is a CBOR text string, specifying the topic name for the scope entry.
  - The permission set ("Tperm") is a CBOR unsigned integer with value, signaling the Client role, operations the Client can execute on Topic Data in which type of group.

```plaintext
AIF-PUBSUB-GROUPCOMM = AIF-Generic<pubsub-topic, pubsub-perm>
pubsub-topic = tstr ; pub/sub topic name
              ; (the associated security group)
pubsub-perm = uint .bits pubsub-perm-details

pubsub-perm-details = &(
    Admin: 0,
    AppGroup: 1
    Publish: 2,
    Read: 3,
    Delete: 4
)

scope_entry = [pubsub-topic, pubsub-perm]
```

*Figure 5: Pub/sub scope using the AIF format*
Next steps

• Define the replay-checks at the subscribers: based on the same approach used for OSCORE and using a Replay Window

• REQ10: Register a Resource Type for the root url-path, which is used to discover the correct url to access at the KDC: the Resource Type (rt=) Link Target Attribute value "core.ps.gm" is registered in Section Section 8.3.
  • ToDo: This possibly will not stay as the final method for KDC discovery

• REQ20: If used, specify the format and content of 'group_policies' and its entries. Specify the policies default values:

• Rekeying:
  • OPT12: Optionally, specify for the KDC to perform group rekeying (together or instead of renewing individual keying material) when receiving a Key Renewal Request: ToDo.
  • OPT14: Optionally, specify additional information to include in rekeying messages for the "Point-to-Point" group rekeying scheme: ToDo.