CAT-4-MOQT

Authentication and Access Control for MOQT

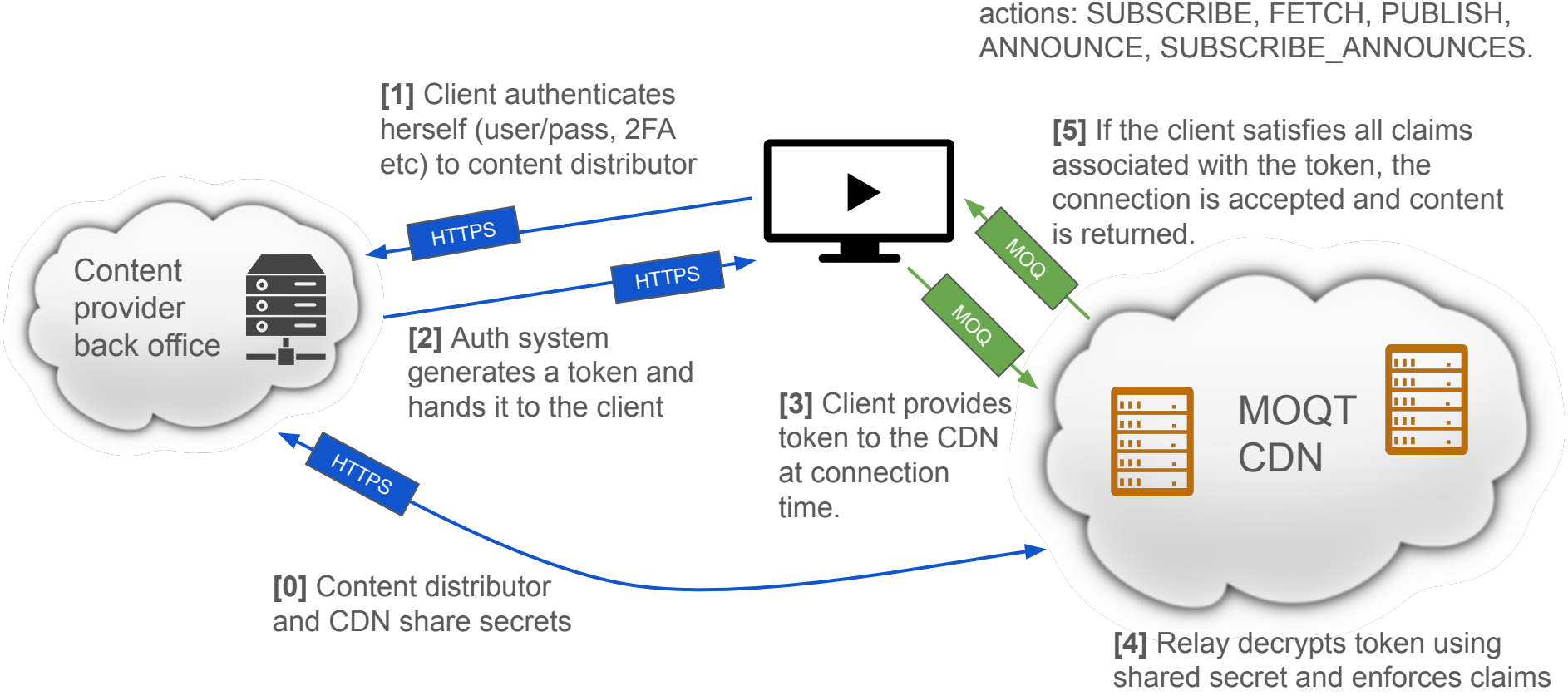
Will Law, Akamai
with input from Gwendal, Cullen and Suhas

IETF #122, Bangkok, March 2025

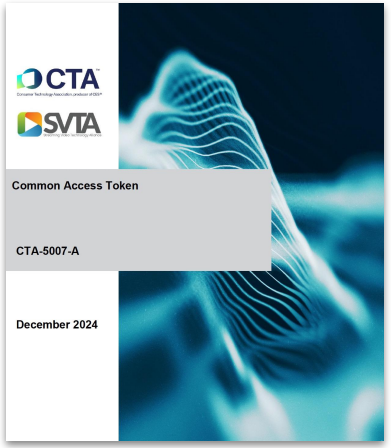
Auth Proposal at Denver interim, February, 2025

- Identified token auth as a practical mechanism to protect and control access & actions on MOQT networks
- Proposed the use an existing standard: Common Access Token CTA-5007-A
  - Compact binary
  - Contains many existing claims useful to MOQT
  - Being used at scale for HTTP media protection
- Two actions requested by chairs following this meeting
  - Generate a draft to show how it might work
    - https://github.com/wilaw/CAT-4-MOQT/
  - Reach out to CTA WAVE ato ask about the feasibility of extending CAT to support a new MOQ claim
    - Had a meeting with CTA WG on March 10. The WG indicated that the core CAT spec could be extended by an external RFC and that no work was necessary by CTA.
    - They advised they will shortly be updating the spec to CTA 5007-B

Token workflow



CTA-5007-A Common Access Token



- Published Dec 2024
- Built on top of
  - IETF RFC 8392, CBOR Web Token (CWT)
  - IETF RFC 9053, CBOR Object Signing and Encryption (COSE):
- Open standard
- Uses CBOR for binary compactness
- Being used to harmonize access control across HTTP media by major CDNs and distributors
- Provides a wide set of claims: issuer, audience, token ID, network IP, network protocol, URL (scheme, host, port, path, query, parent,filename,stem,extension)
- Supports token renewal
- Recipients MUST support the HMAC 256/256, PS256 and ES256 algorithms. They SHOULD accept a CWT Signed or MACed with any algorithm they support.

We have two layers at which we need to apply access control

The **initial CONNECTION (before MOQT is even active)**

WebTransport	https://example.com/myapp
QUIC	moqt://74.6.231.20

Time
IP CIDR
AS
Geo
URL (host, port. path) etc

The **ACTIONS** we can take once connected

SUBSCRIBE
FETCH
PUBLISH
ANNOUNCE
SUBSCRIBE_ANNOUNCES

Initial Connection

WebTransport	https://example.com/myapp
QUIC	moqt://74.6.231.20

Covered well by the existing claims within CAT:

- Expiration time, notBefore
- IP: address, CIDR, range, method, ALPN
- AS number
- Geo location
- URL (scheme, host, port, path, query, parent,filename,stem,extension)
- Renewable
- Issuer
- Audience
- Token ID
- Replay
- Probability of Rejection
- Version
- TLS public key
- Issued At

How to authorize MOQT actions with the same token?

All the MOQT authorizable actions (SUBSCRIBE, FETCH, PUBLISH, ANNOUNCE, SUBSCRIBE_ANNOUCE) can be constrained by a **NAMESPACE|NAME prefix match.**

The rules are:

1. All actions are disallowed by default. Only allowed actions must be explicitly enabled.
2. For each ACTION, we specify a PERMISSION
   a. 0 - Allowed for all Namespaces and Names
   b. 1 - Allowed with an exact match
   c. 2 - Allowed with a prefix match
3. The client is authorized to perform the action if the NAMESPACE|NAME requested in the action matches the PERMISSION

How to authorize MOQT actions with the same token?

Example: Allow with an exact match "example.com/bob".

```
Permits

* example.com/bob

Prohibits

* example.com

* example.com/bob/123

* example.com/alice

* example.com/bob/logs

* alternate/example.com/bob

* 12345
```

Example: Allow with a prefix match of "example.com/bob".

```
Permits

* example.com/bob

* example.com/bob/123

* example.com/bob/logs

Prohibits

* example.com

* example.com/alice

* alternate/example.com/bob

* 12345
```

Reminder: MOQT namespace/name is a binary tuple. Showing the text representation here for clarity.

Creating a new CWT Claim type “moqt” to carry this data.

The "moqt" claim is defined by the following CDDL:

```
$$Claims-Set-Claims //= (moqt-label =>
moqt-value)

moqt-label = XXX // Need to register with IANA

moqt-value = [ + moqt-object ]
```

The moqt token needs to encode multiple instances of 5 actions, currently

0 - ANNOUNCE

1 - SUBSCRIBE_ANNOUNCES

2 - SUBSCRIBE

3 - PUBLISH

4 - FETCH

For each action, we need to communicate the permission

0 - Allowed for all Namespaces and Names

1 - Allowed with an exact match

2 - Allowed with a prefix match

For permissions options 1 & 2, we also need to specify the prefix as a byte array

Prefix - byte array

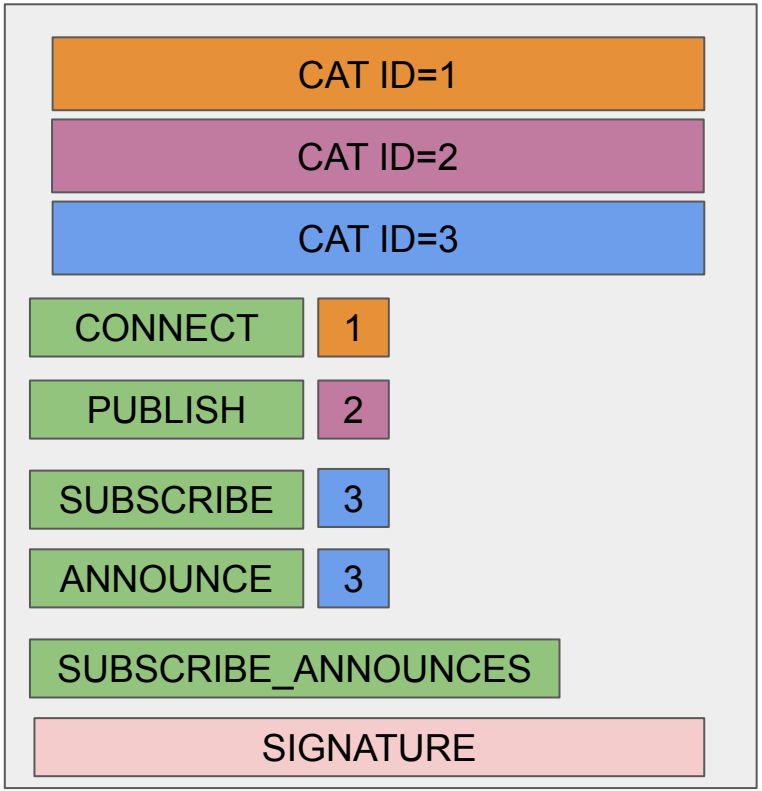
Composite claims

How do we bind ACTIONs to differentiated base CAT claims?

For example , you can connect and SUBSCRIBE from anywhere in Europe after 4pm but you can only PUBLISH after 6pm in Germany?

1. One solution is to give the client different CAT tokens for each ACTION, which the client then gives to the relay as it invokes each action.
2. A second solution would be for CAT to adopt Chris Lemmons proposed composite "and", "or", "nor", "crit", and "env" claims in this draft.
    a. Problem: A CWT token does not allow a claim to be repeated. We can't specify two expiry claims within the same token and have them apply to different actions.
3. Or we say that base CAT claims apply equally to all ACTIONS declared in a MOQ claim (which means we don't support the use-case above)
4. Or create a wrapper which combines several CAT tokens together.

A wrapper of multiple CAT tokens for use with MOQT




```
{
        CAT count (i)      // number of tokens declared
        [CAT ID (i)        // ID for this token
        CAT length (i)     // token length
        CAT value (..)]    // token value
        Actions count (i)  // number of actions declared
        Action type (i)    // 0 = connect, 1=subscribe etc
        CAT ID to apply (i)  // which declared token to apply
        Signature type (i)   // which type of signature, if any
        Signature length (i)  // length of signature
        Signature value (..)  // signature value
}
```

Issue #4: Token Revocation

Tokens can include an ID, which allows them to be revoked. This is used to stop a stream which is being leaked to pirates.

With CAT, we have a single ID. We can use this to completely revoke all the ACTIONS enabled by that token and terminate the connection.

Question: is it an auth use-case to be able to revoke a subset of ACTIONS declared in a token? For example, allow the client to remain connected and SUBSCRIBING but stop their ability to PUBLISH?

Issue #1: Token Renewal

CAT defines 4 methods by which a token can be renewed by the CDN, all biased for HTTP delivery:

- Automatic renewal - conveyed in the same manner it was originally received
- Cookie renewal - new token sent in a cookie
- Header renewal - new token sent in a response header
- Redirect renewal - redirecting the client to a new location which includes a new token in the PATH.

Question: how can we accomplish this in MOQT?

1. New control message - carries the Auth ID of the token it is renewing, along with the new token.
2. Out-of-band - client is given information about the expiry date of its token, or else it's simply given new token at the appropriate time.
3. Piggyback a new CAT token in the Extension Header of an Object. If the Subscriber has an ongoing streaming (FETCH or SUBSCRIBE) then it receives the token with the Object.
4. In the case of a client who has an ongoing PUBLISH, the Relay can piggyback the renewed CAT token in the Parameter in a SUBSCRIBE_UPDATE.
5. Some other vector?
6. Do we even need to renew tokens? The initial token gives a persistent secure connection which is not present in HAS workflows.