

Application-Agnostic DPoP Framework for MOQT

[draft-nandakumar-moq-dpop-proof](#)

Authors: Suhas Nandakumar, Cullen Jennings

IETF 125

Why we need a new draft for DPoP proofs in MoQ?

- CAT tokens can be **bearer tokens or a sender-constrained** tokens (via DPoP).
- The problem with bearer token is they can be easily replayed.
- Sender-Constrained tokens cryptographically bound to the client's key pair, so only the holder of the private key can use it, not anyone who merely possesses the token.
- CAT-DPoP Tokens **are bound to HTTP request and methods.**
- Scope of the draft is to:
 - This draft defines **DPoP bindings to MOQ protocol** message for CAT tokens.
 - Fills in the gap and doesn't address new security concerns.

Attack vectors and motivation for DPoP in MOQ

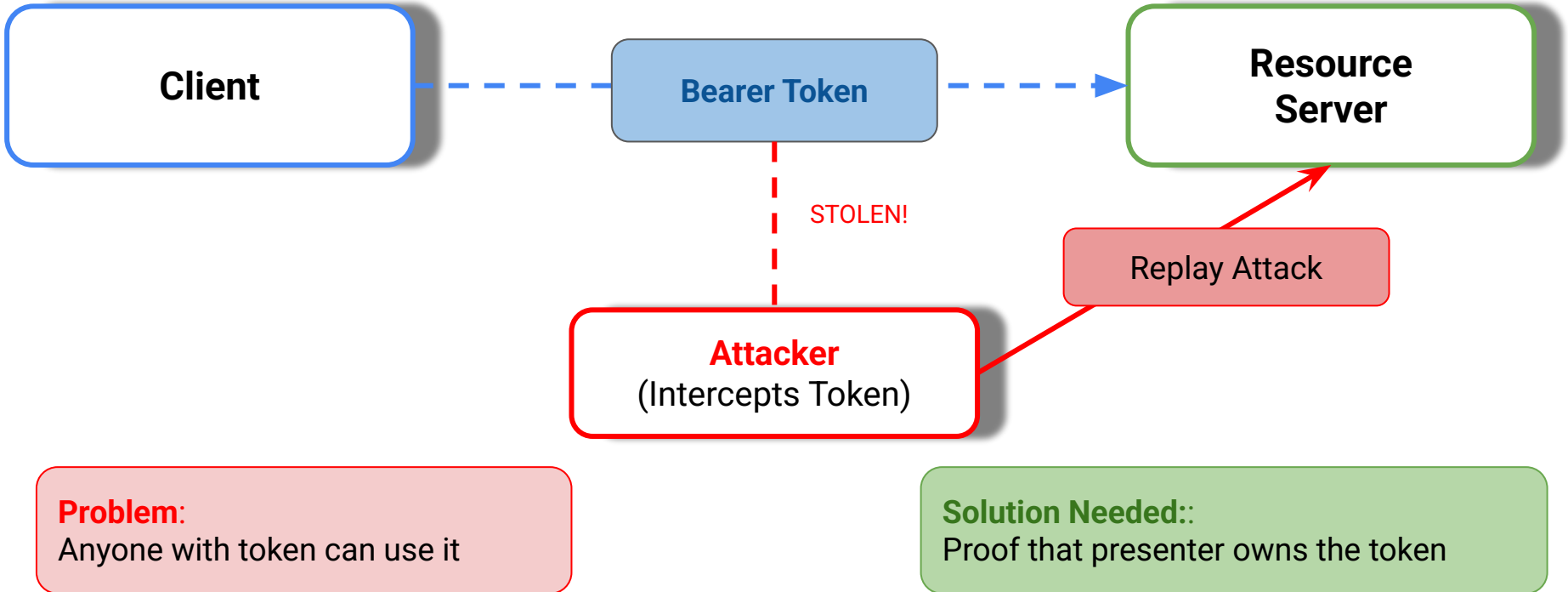
- **Compromised Relay Token Theft** A malicious or breached relay can extract bearer tokens from MOQT connections and distribute them to botnets, enabling millions of unauthorized subscribers to a stream.
- **Token Replay Attacks** Captured tokens can be replayed indefinitely until expiry; attacker doesn't need to prove possession of any key material
- **JavaScript Exfiltration** In browser-based MOQT clients, malicious scripts or XSS attacks can steal tokens from memory/storage and send them to external attackers
- **Token Sharing/Credential Stuffing** Users can trivially share bearer tokens across devices/accounts; no cryptographic binding to original client
- **Man-in-the-Middle Token Harvesting** Any intermediary in the path (CDN edge, proxy, compromised network) can collect tokens for later unauthorized use
- **Amplification via Token Distribution** Single stolen token → distributed to N attackers → N simultaneous unauthorized streams, multiplying resource abuse and content piracy

Why MOQ WG (Not OAUTH WG)

Background discussions with OAUTH Chairs and Security AD

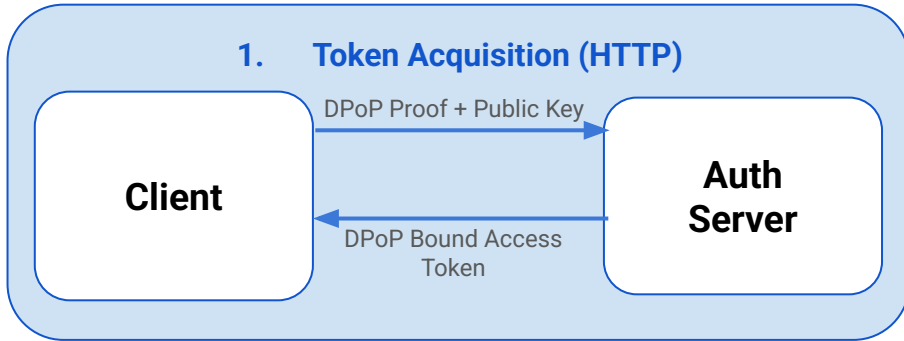
- **Long-term:** IETF may establish an OAuth Directorate - multiple WGs need OAuth expertise
- **Near-term:** Get knowledgeable OAuth expert to help with draft - quickest path forward and continue the work within MOQ WG
- Security AD (Deb Cooley), after talking to OAuth chairs suggested that we do this work in MOQ WG, since we are the users and probably fits within existing charter
 - We need to circle back with all the chairs and ADs for MOQ and OAuth before adopting.

The Problem - Token Replay Attacks

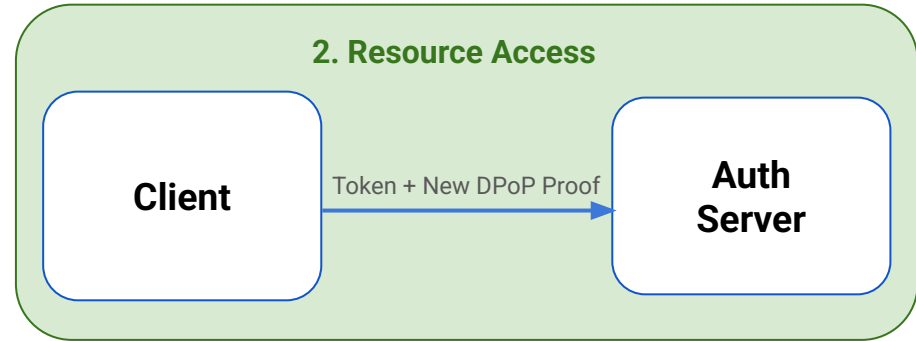


What is DPoP ? (RFC9449)

1. Token Acquisition (HTTP)



2. Resource Access



The Limitation: HTTP-SPECIFIC

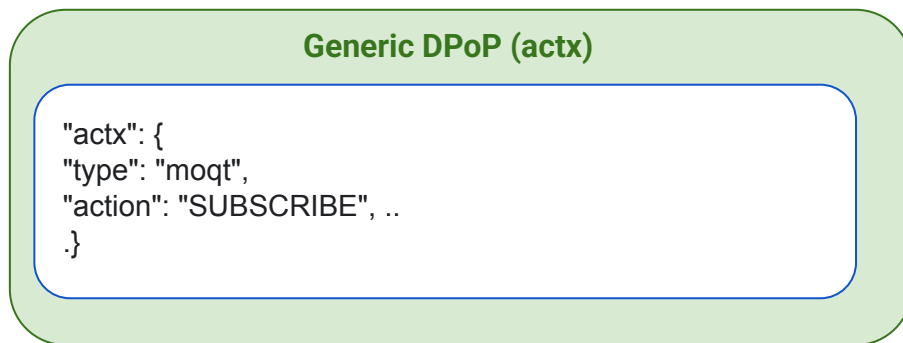
RFC 9449 binds proofs to:

htm	HTTP Method
htu	HTTP URI

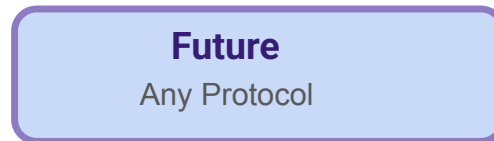
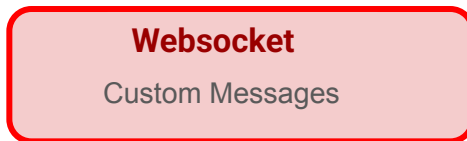
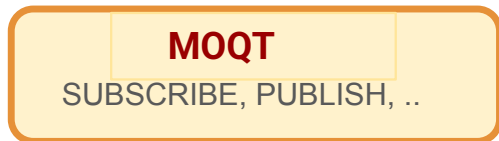
MOQT runs over QUIC,
not HTTP

Need application-agnostic
binding

Proposed Solution - Generic DPoP Framework



Extensible to Any Protocol



Token acquisition remains unchanged (RFC 9449 Sections 4-6, 8)

MOQT Context Type Definition

Field	Description	Required
type	"moqt" - Protocol identifier	Yes
action	MOQT operation (SUBSCRIBE, PUBLISH, etc.)	No
tns	Track Namespace (serialized per MOQT 1.5.1)	Yes
tn	Track Name	No
parameters	Additional MOQT-specific parameters	No

Validation

Server verifies action matches MOQT operation

Validates namespace/track permissions

Prevents replay across different operations

Dual Format Support - JWT and CWT

JWT Format

JSON-based,
human-readable

Works with existing
OAuth infrastructure

typ:"dpop-proof+jwt"

CWT Format

CBOR-based,
compact binary
encoding

Ideal for constrained
environments

typ:"dpop-proof+cwt"

CAT Interoperability

Integrates with
Common Access Token
Systems

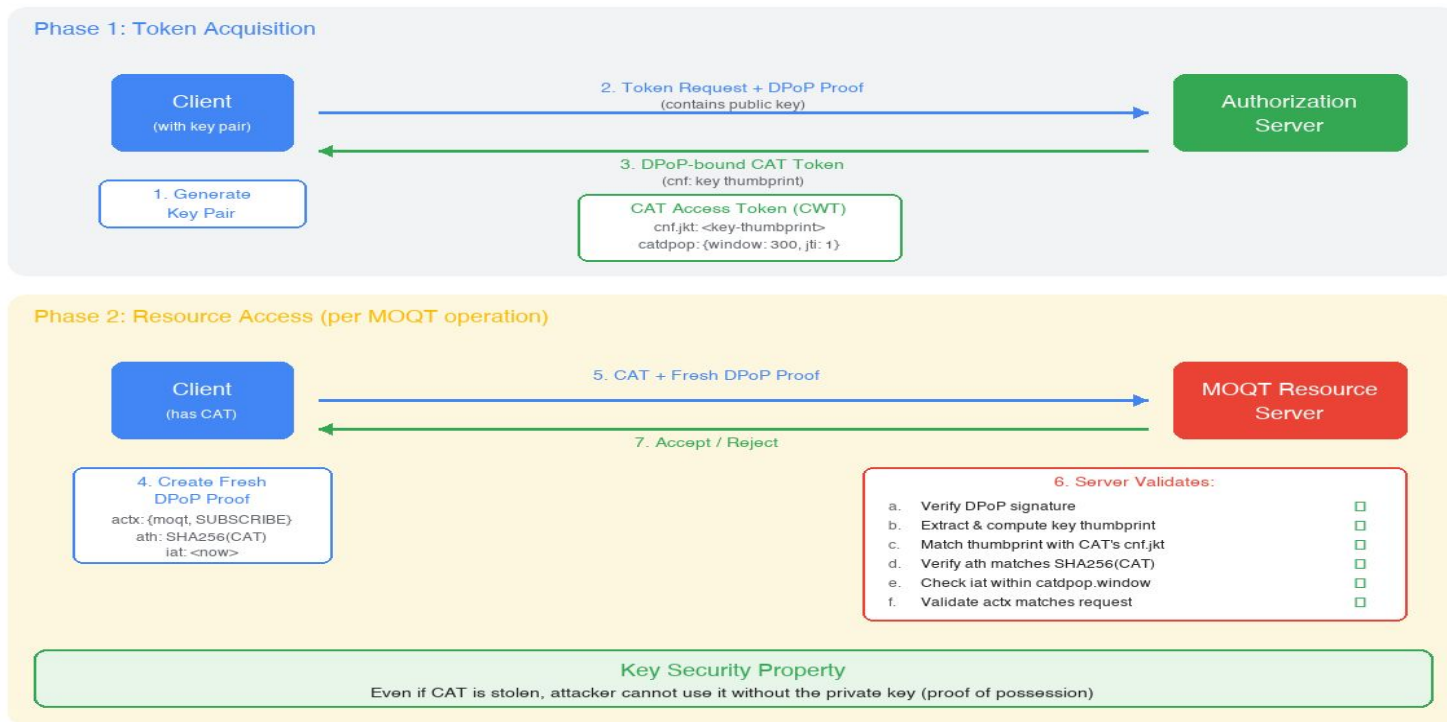
Uses by major
streaming platforms

Servers MAY support both formats simultaneously, enabling flexibility for
different deployment scenarios

DPoP Proof of Possession Flow

DPoP Proof of Possession Flow (MOQT + CAT)

Demonstrating Proof-of-Possession with Common Access Token



Security Properties

Inherited from RFC 9449

Sender Contains Tokens

Cryptographic Key Possession Proof

Nonce-based replay protection

Additional Protections

Cross Protocol Security

Operation Binding

Context Validation

Application Separation

Attacks Prevented

Token Theft

Cross Protocol Misuse

Operation Hijacking

Replay Attacks

Next Steps

- Looking for feedback from people that want to use sender-constrained tokens
- We have a working implementation for C4M with DPoP Support here: <https://github.com/Quicr/catapult/>
- Looking for interop & feedback
- Ongoing discussions with Chairs & AD to decide the right WG to continue the work