Handshake attacks: (with MitM position)

* We can manipulate unprotected parameters such as CID and stk token, and then to force the client to downgrade to a TLS/TCP connection (then maybe use TCP ACK flooding confusing the server to drop the connection.
* When a client sends ‘ClientHello’ to a server, we can send him a reject message (with spoofed source address token\*) when the client approach to the server with invalid source address token, the server replied with reject message that forces him to repeat the connection. **DoS from the client**

**\*-** an authenticated an encrypted byte string from server to client (client IP and server timestamp encrypted with AES-GCM).

* There is an attack which manipulates unprotected parameters (CID, sa token…) and causes the client and the server to possess different initial keys, DoS.
* State-overflow: QUIC keeps a state from each connected client therefore we can flood him with connection similarly to TCP syn flood. A defensive mechanism that could defend against this attack is to retry authentication if the IP address cannot be verified, but it is against QUIC’s design goals since it produces an additional overhead.

QUIC flooding:

We can flood the server with 0-RTT connections (apparently the best option for flooding) to increase its CPU usage significantly. This attack paralyzes the server (depends on the server implementation) for 20-30 seconds repeatedly. Servers can employ the RETRY feature of QUIC to prevent this attack.

Future and challenges section:

* None of the works so far has examined the QUIC endurance against IP spoofing attacks.

We can examine whether address validation protection is indeed effective and maybe

Using UDP amplification attack.

* We can check web cache poisoning attacks.