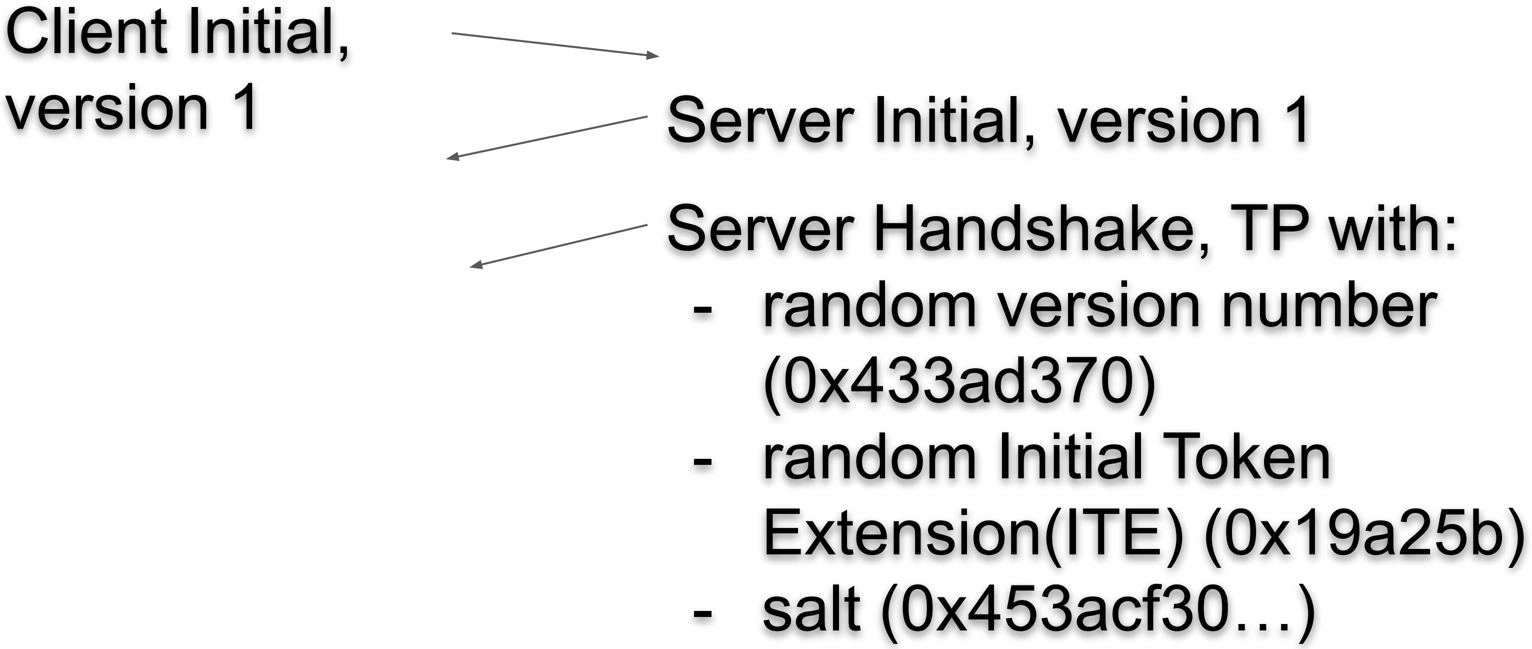
QUIC Version Aliasing

https://datatracker.ietf.org/doc/draft-duke-quic-version-aliasing/

First Connection



Client Initial, version 1

Server Initial, version 1

Server Handshake, TP with:

- random version number (0x433ad370)
- random Initial Token Extension(ITE) (0x19a25b)
- salt (0x453acf30…)

The salt is a secure hash f(version, ITE)

Next Connection

Client Initial,
version
0x433ad370
token {N} +
0x19a25b

Server computes salt from version, ITE

- Connection continues with aliased version number
- Server SHOULD issue TP with new values

Claimed Properties

- From second connection, Initial packet payloads are **entirely private** and **immune from ossification**
- Minimal TLS ossification vectors over QUIC
- Greases the version field
- Initial Injection attacks are over (maybe VN and Retry might work)
- Server has no per-client state
- More space-efficient than ECHO, covers the whole Initial packet, both authenticated and private in both directions
- (For the moment) does nothing for the first connection
- Dependency on quic-version-negotiation
- Browsers & economically important websites need to deploy it to prevent firewalls from killing it

Potential Improvements: First Connection

- Could use asymmetric keys + DNS, like ECHO, to avoid bootstrapping problem
- more code and more computation.
- first-connection CHLO private, not authenticated; SHLO authenticated, not private
- draft-kazuho-quic-authenticated-handshake-01 also did this to authenticate (but not encrypt) the Initial

Feedback wanted:

https://github.com/martinduke/quic-version-aliasing

Any browsers and “economically important websites” interested?