Preshared Keys – implementing and testing

A preshared key consists of a label – any string of any length – the key itself, maybe 16 random bytes for an AES-128 key, and an indication of which cipher suite to use it with. TLS1.3 has cleverly merged the use of pre-shared keys with the resumption mechanism, so that very little change is necessary to get it to work. Not really surprising as after all a resumption ticket is basically a form of pre-shared key. And indeed the same ticket structure can be re-purposed to hold a preshared key. But of course a pre-shared key has no age limit so certain fields can be ignored. A preshared key then presents as a ticket with an age of 0.

Testing it

The problem with these non-browser configurations is that they are difficult to test. The only option seems to be to use OpenSSL to mimic a server, and figure out which of its blizzard of optional flags to use. In this case its

```
openssl s_server -tls1_3 -cipher PSK-AES128-GCM-SHA256 -
psk_identity 42 -psk 0102030405060708090a0b0c0d0e0f10 -nocert -
accept 4433 -www
```

Observe that there is no certificate involvement.

Some issues

Some interesting issues arise – for example which group to use for the key exchange? As you will recall in a full handshake the client in its clientHello sends a public key share from one of the standard groups. If the server accepts it the key exchange completes after the serverHello. But if the server does not support the proposed group an HelloRetryRequest is sent by the server, and the client tries another group.

Now once that is sorted out the client should know which group the server likes, so it won’t make the same mistake again. So on a resumption there should never be a need for an HelloRetryRequest. And if using a pre-shared key, surely if they can agree a key, they can also agree a group. So I would maintain that an HelloRetryRequest should never be required on a resumption, or if using a pre-shared key.

A similar issue arises when deciding what cipher suite to use on resumption. Initially, according to the standard, the same cipher suite should be used as for the initial full handshake, but in theory the server might now change its mind and request a switch to a different cipher suite. This would appear to require the client to maintain more than one transcript hash until the server makes its mind up as to which cipher suite and hash function to use.

These problems can be resolved by the client, on resumption, only offering one choice of key exchange group and one choice for cipher suite, these being those agreed previously. Since the server happily negotiated these on the first full handshake connection, there is no reason to assume that they won’t be acceptable on resumption. And in the worst case we can always allow resumption to fail, and fall back on a full handshake.

This has the added advantage of making the clientHello on resumption even smaller.

In short the server capabilities and preferences are discovered after a full handshake, or by pre-arrangement along with a pre-shared key. Any subsequent connection after that can assume that the servers capabilities and preferences have not changed.