Stepping back a bit

The structure of TLS1.3 communications is becoming a bit clearer. First there are records. A record starts with a header

XX 03 03 L1 L2

where XX=16 for Handshake records, and XX=17 for Application records. The length of the subsequent record data is given by L1 and L2, a 16-bit number.

If XX=17 then the record contents are encrypted, using an AEAD mode of a block cipher (probably the AES). In this case the header is included unencrypted in the AEAD calculation. A final encrypted byte indicates the *actual* record type (as handshake data may be disguised as application data!), again 16 for handshake data, 17 for application data. Note that this final byte is encrypted. Cunning or what?

If the record contains application data, simply pass it up to the application. But if it contains handshake data we need to know what kind of handshake message it contains. This is encoded as

YY L1 L2 L3

Where YY indicates the handshake message type, and its length is the 24-bit number given by concatenating L1, L2 and L3

Notice that it would appear that the handshake data size at 24-bits could exceed the 16-bit length of a record. And indeed this is the case. The reason is – fragmentation. Records are kept small to improve communications latency, and so handshake data might be spread across multiple records. So

RH HH HD RH HD RH HD ..

where RH is a record header, HH is a handshake header, and HD is handshake data.

Tickets

In fact I never actually completed the TLS handshake, working from the example at https://tls13.ulfheim.net/. So I have now done so, down to picking up the two “tickets” that the server sends at the very end. Note that not all servers supply tickets – www.bbc.co.uk is a good example of one that does. By convention at most 2 tickets are supplied at the end of the full handshake. These contain information which allows a session resumption at much lower cost than going through the whole handshake again.

So next up is to attempt a session resumption using one of these tickets…

Small problem

When the whole TLS handshake is finished the application should take over, with all its communications encrypted. So I was thinking if as application data I send a HTTP GET instruction to the website, I should get a whole ton of encrypted HTML in response. But its not happening, the server simply times-out and drops the link. So what am I doing wrong (using www.bbc.co.uk for testing).

On https://tls13.ulfheim.net/ at the end he sends “ping” and gets a “pong” response. I am sending “GET …” to the website, but getting no response.

Maybe GET should be sent on a different port?? Or something??