Early Data

The client sends a simple indication in their initial **`clientHello`** that they intend to send some early data in the first flight of client to server communication. The key to be used is derived from a pre-shared key, so this will only work after session resumption, or if a pre-shared key has somehow been established out-of-band.

The server responds after its **`serverHello`** by immediately sending “encrypted extensions”. These are encrypted with the full handshake keys (which incorporate the Diffie-Hellman key exchange secret). These encrypted extensions are commonly empty, but now they should now contain an indication of whether or not the early data was accepted.

If the early data were accepted (and for various reasons the server might have rejected it), then the client must respond by sending an end-of-early-data signal to the server, which tells the server that the client has switched over to using the (safer) full handshake keys for future communication.

So that’s it. Quite simple really.

Not so fast

Session resumption happens when the client presents a “ticket” to the server. This ticket will normally be gifted to the client after a previous successful full TLS handshake. The ticket has a lifetime roughly in the range from 5 minutes to 5 days.

However whereas a valid ticket can always be used to resume a session, that does NOT mean that it permits early data as part of that resumption. A ticket can be used for early data if it includes an extension indicating the maximum size of that early data.

So a ticket may simply specify a maximum size of zero, in which case early data is not possible. And it turns out that the tickets offered by many websites use this mechanism to disallow early data. Of the 10 sites I use for testing (including facebook and google), only one obscure site issues tickets which support early data on resumption!

So just because you have a ticket does not mean you can do early data. Some tickets are more useful than others. And even if you get a more permissive ticket which allows early data, the size of that data may be very constrained.

Conclusion

It appears that early data is not used much, despite its big fanfare as a major feature of TLS1.3, certainly not in the context of Web servers. In the end I could only find one site I could do a fully successful test against – swifttls.org

Again a big problem is firing stuff at a server and trying to get some indication back as to why its not working! A nightmare for debugging. So I contacted the owner of swifttls.org who agreed with me that the RFC is quite ambiguous as to what constitutes a proper implementation. He modified his server side code to respond to our early data.

Next up I need to tidy my code.