Long time no blog

Well we were waiting for standardised post quantum KEM and signature schemes to arrive. Well now they have – MLKEM (based on Kyber) and MLDSA (based on Dilithium). Experimental support has now been added to TiigerTLS, and also a hybrid mode which uses a combination of a PQ method with an established elliptic curve method.

Some other developments. A new crypto library has been developed which supports elliptic curve cryptgraphy more efficiently in rust and C – see https://github.com/mcarrickscott/TLSECC . This has now been integrated into an optional C++ SAL, and adopted by default for the rust SALs.

A weird bug was detected and fixed. Resumption tickets were not being accepted by BoringSSL, due to the resumption client hello not including an extension to indicate supported signature schemes. It turns out that if a resumption fails for whatever reason the server should still be able to fallback to a full handshake in response to the client hello – and to do this signature support is required. Obviously if the resumption succeeds, this would not be needed. But BoringSSL flags an error anyway.

In fact TiigerTLS does not attempt this fallback, but rather terminates the protocol attempt and starts afresh with a full handshakem with a new client hello. This is not disallowed by the standard.

Now we regard this as a feature rather than a flaw. Valid resumption tickets will work the vast majority of the time, so such events will be rare. And our approach allows a resumption handshake to be completely free of any PKI baggage. So we only need to support full and resumption handshakes, and not a third option where what starts out as a resumption handshake and then changes into a full handshake.

Maybe some day PKI will disappear and all handshakes will be resumptions!

Another change is that the STEK, the key that the server uses to encrypt its tickets, is now randomised every time that the server starts. This is a lot more secure than having a fixed built-in key.

A smaller client example program has been provided which provides a minimal implementation. This illustrates how with just a few function calls it is possible to securely connect to a server, and to exchange encrypted data with it.

The arduino code and README has been updated to support the latest Raspberry Pi Pico boards.