What about Ed25519and Ed448?

Good questions. It took years for elliptic curve cryptography to reach a state of maturity. Many of the initial standardised curves were over time discovered to have weaknesses and/or trust issues. By the time the community had arrived at a consensus on the ideal properties of a secure elliptic curve (see https://safecurves.cr.yp.to/), post-quantum cryptography had arrived on the scene, suggesting that elliptic curve cryptography may not survive for long.

Almost as an afterthought Ed25519 and Ed448 based implementations of the EdDSA signature algorithm were added to the TLS1.3 specification. They should over time have replaced the older standards, but in fact they never achieved dominance. Why?

Well the tidal wave, that is a quantum computer, did appear at one time to be imminent. That threat has for the moment receded. But there are a couple of other contributing factors. One is the fact that the Edwards curves on which Ed25519 and Ed448 are based have a small co-factor (of 4 or 8) in the number of points on the curve. Now this would seem to represent just a small niggling detail, but in fact it created an endless amount lot of confusion. This issue doesn’t arise for the older standard curves, which have also recently benefited from the discovery of reasonably efficient exception free formulae (https://eprint.iacr.org/2015/1060). These old and less-than-perfect curves also benefit from an often overlooked attribute – longevity. They have been around for a long time, and, despite their shortcomings, have basically never been broken. So if it ain’t broke don’t fix it.

But that may not be the end of the story. Ed25519 in particular has achieved a lot of penetration in protocol settings other than TLS (https://ianix.com/pub/ed25519-deployment.html), in part due to the lovely simple-to-use and ultra-secure implementation in the well known libsodium library. The stumbling block for TLS is the absence of support for these curves in PKI certificates. That may come in time – PKI must anyhow re-invent itself in the context of post-quantum crypto. And if a quantum computer takes another generation to appear then these curves may yet find themselves a dominant place in a future TLS.

Supported

Anyway support is now included as standard for the Ed25519 and Ed448 curves in TiigerTLS, using by default by the implementation in the miracl core library. Ed25519 support can be tested against the site ed25519-test.germancoding.com which has an experimental Ed25519 based certificate (certificate checks might need to be temporarily turned off), or against openssl.

To enable fuller local testing a new CRYPTO_SETTING has been introduced (for C++ see tls1.3.h, for Rust see config.rs) which uses a special openssl generated EdDSA-based certificate chain for the Rust server. The root and intermediate certificates use Ed448, while the leaf (server) certificate uses Ed25519. The root cert has been place in our C++ and Rust client’s certificate store.