TLS PAKE

Sometimes you only have a (low-entropy) password

With TLS <1.3, could use SRP ciphersuites [RFC5054]
... but SRP suites don't translate directly 1.3
... and there's been some more work on PAKEs since SRP
... for example, draft-irtf-cfrg-spake2, draft-krawczyk-cfrg-opaque

Proposal: Add an extension to enable TLS 1.3 to enable use of PAKEs for key exchange and mutual authentication

https://datatracker.ietf.org/doc/draft-barnes-tls-pake/
**TLS 1.3 + SPAKE2PAKE**

Key Exchange

Key Confirmation

---

Key ^ ClientHello
Exch + pake*
| + key_share*
| + signature_algorithms*
| + psk_key_exchange_modes*
v + pre_shared_key* -------->

ServerHello ^ Key
Exch + pake*
+ key_share*
+ pre_shared_key*
v {EncryptedExtensions} ^ Server
{CertificateRequest*} v Params
{Certificate*} ^
{CertificateVerify*} | Auth

{Finished} <-> [Application data*]

^ {Certificate*}
Auth | {CertificateVerify*}
\ {Finished} -------->
Since last time...

Interest in other PAKEs: Dragonfly, OPAQUE, ...

Changed to be a general framework for any PAKE with the right shape

ClientHello/ServerHello extension + Key schedule integration

Per-PAKE definition of messages carried in CH/SH

Open questions:

Identity protection?

Need to negotiate PAKEs?