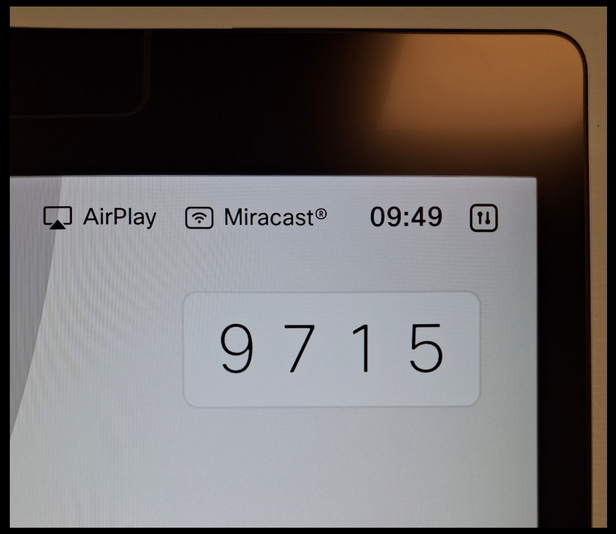
draft-bmw-tls-pake13-01

TLS 1.3 PAKE authentication extension

Laura Bauman, Apple
David Benjamin, Google
Samir Menon, Apple
Chris Wood, Apple

Goals and Motivation

- When peers share only a low entropy secret there is no mechanism to directly establish a TLS 1.3 connection
- Some motivating use cases where this would be valuable include:
  - Connections authenticated with a user-entered secret (PIN / passcode)
  - Embedded/IoT device setup and pairing
- More bootstrapping than long-term authentication
  - NOT aiming for general web login use case
  - Could be used for a one time connection
  - Or as a connection over which long term credentials are exchanged



Goals and Motivation Cont’d

- TLS 1.3 provides a mechanism for authentication with pre-shared keys (PSKs), but the keys must be high entropy
- There is no direct path to get from a low entropy secret to a TLS 1.3 connection
- Different use cases must design mechanisms to move from a low entropy secret to a PSK or Raw Public Key(s) or use a less secure protocol than TLS 1.3.
  - e.g.TLS 1.2 Secure Remote Password Protocol (SRP) [RFC5054]

Proposal

- TLS extension *pake* that can carry data necessary to execute a PAKE within the handshake.
- PAKE registration phase is not covered by the draft
- The choice of PAKE and any required parameters will be explicitly specified using IANA assigned values.
- Each PAKE is used as a black box
- As a first case, we've defined a concrete protocol for executing the SPAKE2+ PAKE protocol [RFC9383].

0. (OOB or application specific registration phase)

Client

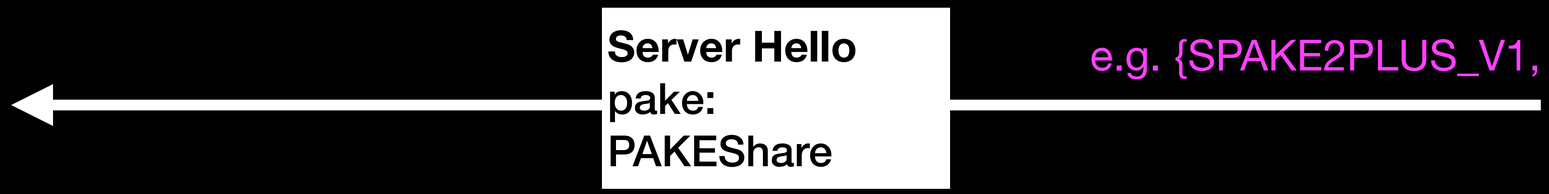
Server

1. Client sends an array of PAKEShares (PAKEScheme identifier + first message) it is willing to use

e.g. [{SPAKE2PLUS_V1, SPAKE2+ client msg}]

2. Server selects PAKEScheme from those offered and sends back next message

e.g. {SPAKE2PLUS_V1, SPAKE2+ server msg}



3. PAKE shared secret replaces* (EC)DHE input to key schedule

e.g. SPAKE2+ derived shared secret

PAKE Requirements

- A compatible PAKE MUST provide forward secrecy*
- And conform to the message flow of the extension**
- Several current PAKE protocols satisfy these requirements including:
  - CPace [CPACE]
  - SPAKE2+ [RFC9383] (specified in the draft already)
  - OPAQUE [OPAQUE]

* Open issue is to combine with regular TLS key exchange. Could relax this.

**A PQ PAKE standard does not yet exist and the specified message flow may need to be modified to allow for an additional round trip

Open Issues

- Certificates (Issue #25)
- Combine with TLS key exchange (Issue #28)
- Generic or PAKE-specific
- Formal analysis/Input from FATT

Status + Next Steps

- Two implementations completed with support for SPAKE2+
  - Including an open source implementation in boringssl
- Does the WG view this as a worthwhile problem?
- Is our approach plausible and issues tractable?