draft-ietf-emu-bootstrapped-tls-02 (aka TLS-POK)

Dan Harkins & Owen Friel

EMU WG

IETF 116, Yokohama, Japan

What?

- How to solve the on-boarding *Catch-22* for wired devices w/o a rich UI
- Reuse Wi-Fi alliance Easy Connect / Device Provisioning Profile (DPP) EC bootstrap key format and all supported bootstrapping methods for wired on-boarding
- Provides mutual authentication between client and server in the resurrecting duckling security model
  - Client proves knowledge of public bootstrapping key as well possession of its private analog
  - Server proves knowledge of public bootstrapping key– demonstrates ownership, authorization to provision *thing*
- Use existing RFCs:
  - RFC 5869 HKDF to derive PSK identity from bootstrap key
  - RFC 8773 Cert Based Auth with External PSK
  - RFC 7250 TLS with raw public key using bootstrapping key
  - RFC 9258 to import derived external PSK
  - RFC 7170 (incl. –bis) to do certificate enrollment
- No new TLS extensions or changes required!

TLS authentication w/DPP bootstrapping keys

out-of-band bootstrapping method

RFC 5869 → RFC 9258

Client

ClientHello

RFC 8773 → **+ cert_with_extern_psk**

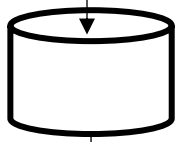
RFC 7250 → **+ client_cert_type=RawPublicKey**

+ key_share

+ pre_shared_key

Server

RFC 5869 → RFC 9258



Server looks up RFC 9258 ImportedIdentity to find bootstrap key

epsk

--------->

ServerHello

+ cert_with_extern_psk

+ client_cert_type=RawPublicKey

+ key_share

+ pre_shared_key

{EncryptedExtensions}

{CertificateRequest}

{Certificate}

{CertificateVerify}

{Finished}

<---------

{Certificate}

{CertificateVerify}

{Finished}

--------->

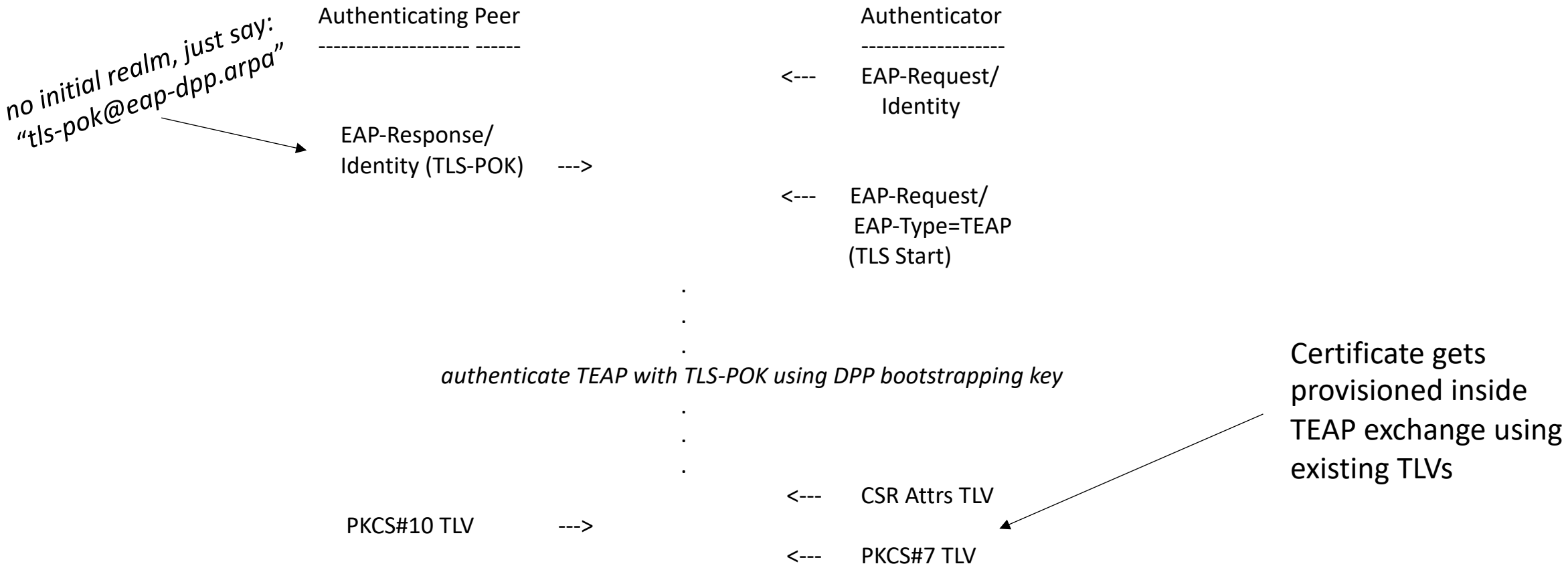
[Application Data] <-------> [Application Data]

Legend:

present for dpp

existing exchange

TEAP w/DPP bootstrapping keys



Supplicant's subsequent connection uses provisioned certificate

Activity Since IETF 115

- New version -02
  - Cleanup stale reference and some nits
  - Tighten up language on TLS handshake
- Question on list concerning RFC 9258 interface
  - Why the additional key derivation since RFC 9258 will hash espk (to ipskx)?
  - Knowledge of bskey authorizes server to provision the client (duckling model*)
  - The ImportedIdentity is passed in the ClientHello and cannot contain bskey
    - Similar to DPP chirp– H(“chirp” | bskey)
  - bskey cannot be <u>both</u> EPSK and identity
  - Will leave the epskid and remove the extraneous epsk derivation: epsk = bskey

epsk = HKDF-Expand(HKDF-Extract(<>, bskey), "tls13-imported-bsk", L)
epskid = HKDF-Expand(HKDF-Extract(<>, bskey), "tls13-bspsk-identity", L)

where:
- epsk is the EPSK Base Key
- epskid is the EPSK External Identity
- <> is a NULL salt
- bskey is the DER-encoded ASN.1 subjectPublicKeyInfo representation of the BSK public key
- L is the length of the digest of the underlying hash algorithm

The [RFC9258] ImportedIdentity structure is defined as:

```
struct {
   opaque external_identity<1...2^16-1>;
   opaque context<0..2^16-1>;
   uint16 target_protocol;
   uint16 target_kdf;
} ImportedIdentity;
```

and is created using the following values:

external_identity = epskid
context = "tls13-bsk”
target_protocol = TLS1.3(0x0304)
target_kdf = HKDF_SHA256(0x0001)

The EPSK and ImportedIdentity are used in the TLS handshake as specified in [RFC9258].

* Stajano, F. and R. Anderson, "The Ressurecting Duckling: Security Issues for Ad-Hoc Wireless Networks", 1999

Where we are and where to?

- Stable document
  - No substantive protocol changes since adoption
  - Reviewed by TLS WG
- Need to fix some references
  - Eric Rescorla didn’t co-author the Resurrecting Duckling paper
  - There’s a new version of DPP out from WFA
- Address Hannes’s comment, removing extraneous derivation but leaving critical one
- Ready for WGLC?

ありがとう