

# draft-friel-tls-eap-dpp

Dan Harkins & Owen Friel

EMU WG, IETF 113

# Background

- Wi-Fi alliance Device Provisioning Protocol (DPP) solves the on-boarding Catch-22– you need a credential to get on the network but need to be on the network to get a credential
  - Uses a raw “bootstrapping” public key, obtained in variety of methods, to authenticate supplicant
  - True zero touch provisioning is possible– plug it in, turn it on, walk away
  - DPP is able to provision all possible network credentials on a supplicant– PSKs, passwords, certificates
- DPP is for Wi-Fi but also supports communication over TCP/IP
  - But such “wired DPP” assumes connectivity we don’t have yet when we do EAP
- We want to use DPP bootstrapping with EAP for non-Wi-Fi connections
  - Use RFC 8773 “external PSK” derived from bootstrapping key
    - PSK derived from bootstrapping key is injected into key schedule
    - Client and server prove knowledge of PSK (and therefore bootstrapping key)
  - Use RFC 7250 TLS with raw public key using bootstrapping key
    - Client signs with bootstrapping key, proves possession of private key to server
  - Use draft-group-tls-extensible-psks
    - Client signals the derived PSK identity and type in extended\_psk extension
  - No TLS changes/extensions required over and above defining new BSK type for draft-group-tls-extensible-psks

# TLS authentication w/DPP bootstrapping keys

bskeypsk = HKDF-Expand(HKDF-Extract(<>, bskey),  
 "tls13-extended-psk-bskey", L)  
 identity = HKDF-Expand(HKDF-Extract(<>, bskey),  
 "tls13-psk-identity-bskey", L)

bskeypsk = HKDF-Expand(HKDF-Extract(<>, bskey),  
 "tls13-extended-psk-bskey", L)  
 identity = HKDF-Expand(HKDF-Extract(<>, bskey),  
 "tls13-psk-identity-bskey", L)

Client

Server

-----  
 ClientHello

+ **extended\_psk=bskey\_id**  
 + cert\_with\_extern\_psk  
 + client\_cert\_type=RawPublicKey  
 + key\_share

----->

ServerHello

+ **extended\_psk=bskey\_id**  
 + cert\_with\_extern\_psk  
 + client\_cert\_type=RawPublicKey  
 + key\_share  
 {EncryptedExtensions}  
 {CertificateRequest}  
 {Certificate}  
 {CertificateVerify}  
 {Finished}

<-----

{Certificate}

{CertificateVerify}

{Finished}

[Application Data]

----->

<----->

[Application Data]

Legend:

**new stuff**

**present for dpp**

existing exchange

# TEAP w/DPP bootstrapping keys

no initial realm,  
just say "tls-pok"

Authenticating Peer  
-----

Authenticator  
-----

<--- EAP-Request/  
Identity

→ EAP-Response/  
Identity (TLS-POK) --->

<--- EAP-Request/  
EAP-Type=TEAP  
(TLS Start)

.  
.  
.  
*authenticate TEAP with TLS-DPP using bootstrapping key*  
.  
.  
.

PKCS#10 TLV --->

<--- CSR Attrs TLV

<--- PKCS#7 TLV

SupPLICANT's subsequent connection uses provisioned certificate

# Where we are and where to?

- Specification:  
draft-friel-tls-eap-dpp-04
- Running code:  
<https://github.com/upros/mint>
- Rough consensus:  
adoption as a work item?