External PSK Importers

draft-wood-tls-external-psk-importer-01

David Benjamin (davidben@google.com)
Christopher A. Wood (cawood@apple.com)

TLS
IETF 104, March 2019, Prague

Hash Reuse

PSKs may only be used with one hash function in TLS 1.3

TLS 1.2 has no such restriction

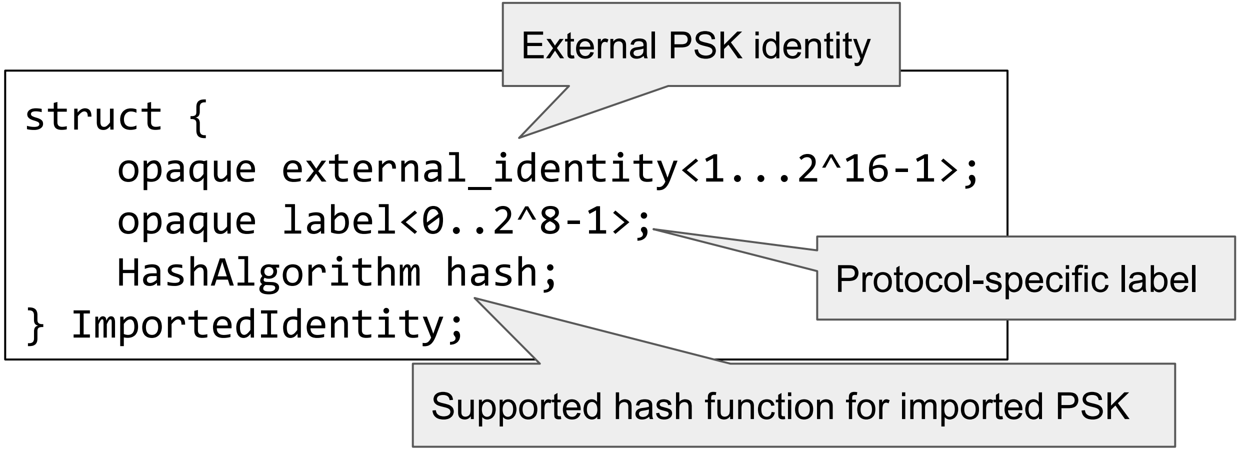
Problem: PSKs may be used in two different contexts with the same hash function

Goal: Allow safe use of existing PSKs and provisioning technologies with TLS 1.3

Key Importers

Diversify an existing “universal” PSK based on supported ciphersuites and protocol versions

Imported keys use an identity based on the external PSK identity



```
struct {
    opaque external_identity<1...2^16-1>;
    opaque label<0..2^8-1>;
    HashAlgorithm hash;
} ImportedIdentity;
```

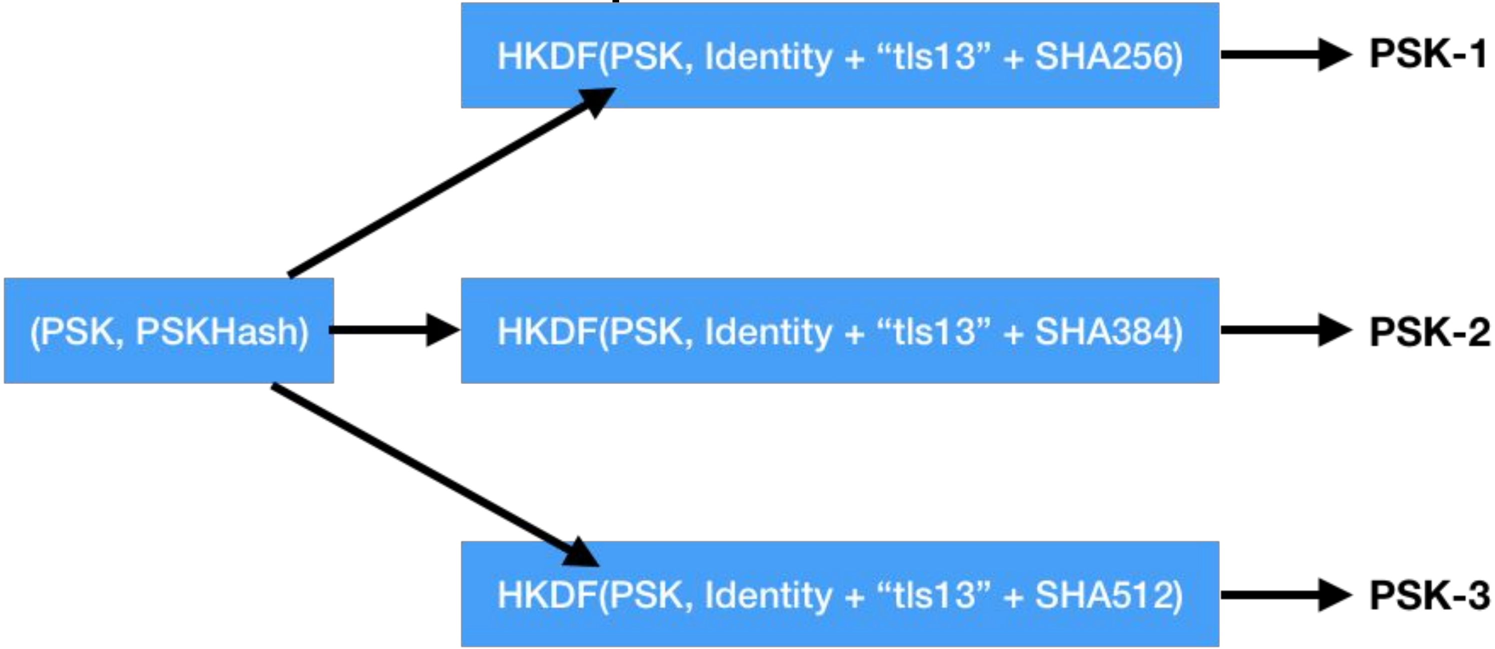
External PSK identity

Protocol-specific label

Supported hash function for imported PSK

Open issue: should `hash` be part of label so that future versions of TLS which do not use HashAlgorithm can still support this?

Diversification Example



(PSK, PSKHash)

HKDF(PSK, Identity + “tls13” + SHA256) → PSK-1

HKDF(PSK, Identity + “tls13” + SHA384) → PSK-2

HKDF(PSK, Identity + “tls13” + SHA512) → PSK-3

*HKDF hash function PSKHash,
SHA256 if unspecified

Assessment

Pros

- Makes no change to the key schedule
- Adds few (|label| + |hash|) bytes per PSK
- Supports TLS 1.3 and 1.2, and works with QUIC

Cons

- PSK count increases multiplicatively with each new version and hash function

Draft Changes

- Clarify requirements for single-hash-function PSKs
- Clarify that duplicate PSKs should be imported for each ciphersuite they support
    - TLS_AES_128_GCM_**SHA256** and TLS_CHACHA20_POLY1305_**SHA256** require one PSK
    - TLS_AES_128_GCM_**SHA256** and TLS_AES_256_GCM_**SHA384** require two PSKs
- Clarify TLS 1.2 support with an explicit label
- Add privacy considerations and a sketch design from draft-ietf-dnssd-privacy

Next Steps

Should the WG adopt this document?