External PSK Importers

draft-wood-tls-external-psk-importer-00

Christopher A. Wood (cawood@apple.com)

TLS
IETF 103, November 2018, Bangkok

Motivating Text

TLS 1.3 takes a conservative approach to PSKs by binding them to a specific KDF. By contrast, TLS 1.2 allows PSKs to be used with any hash function and the TLS 1.2 PRF. **Thus, any PSK which is used with both TLS 1.2 and TLS 1.3 must be used with only one hash in TLS 1.3, which is less than optimal if users want to provision a single PSK**. The constructions in TLS 1.2 and TLS 1.3 are different, although they are both based on HMAC. While there is no known way in which the same PSK might produce related output in both versions, only limited analysis has been done. **Implementations can ensure safety from cross-protocol related output by not reusing PSKs between TLS 1.3 and TLS 1.2.**

Hash Reuse

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

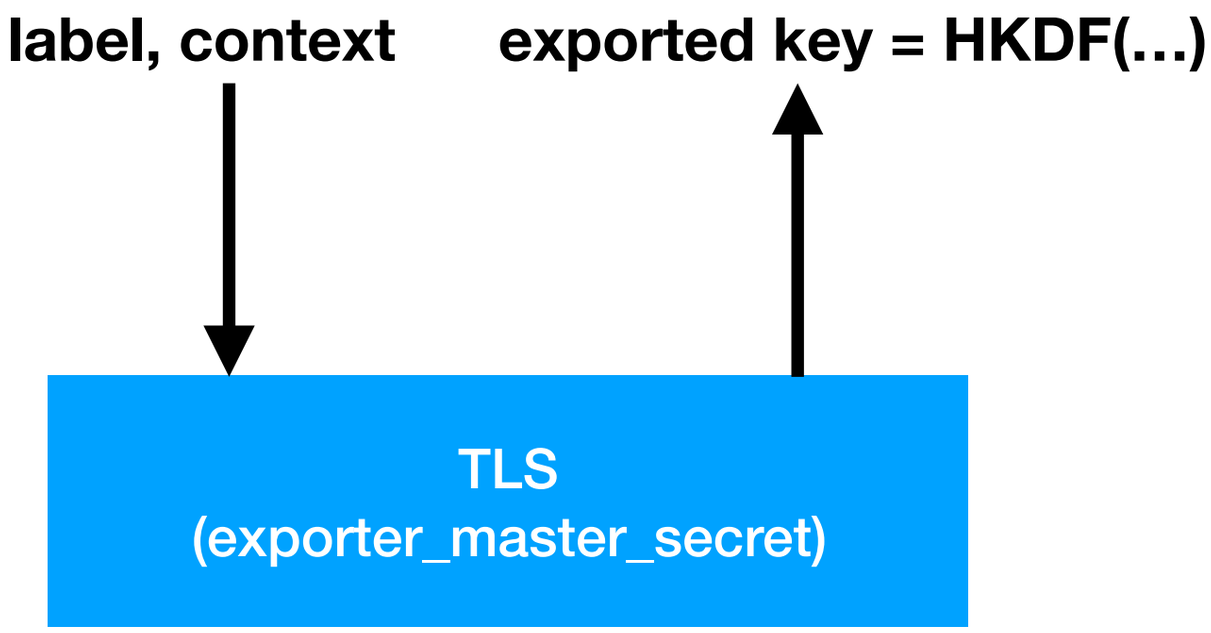
TLS 1.2 has no such restriction

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

Exporters

Purpose: diversify the TLS exporter secret using context information



Importers

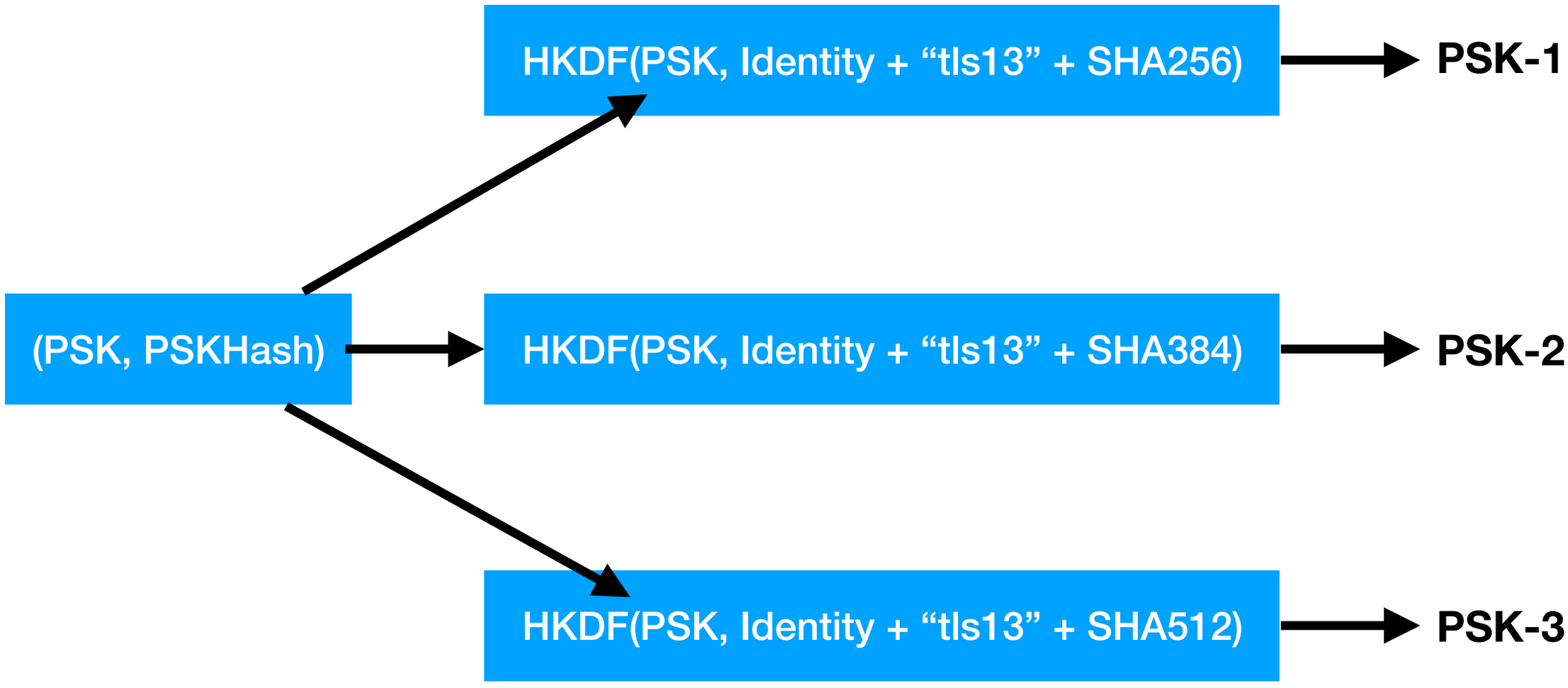
Purpose: “import” existing PSKs into TLS 1.3 for use, diversifying them by hash function and an optional label meant to change across protocols (“quic”, “tls13”, …)

PSK identities are joined with these diversifiers

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

Importers split the PSK “base key” into multiple keys used for PSK binders

Importer Derivation



***HKDF hash function PSKHash, SHA256 if unspecified**

Constraints and Issues

External PSKs may only be imported for use with early data if the relevant connection (ALPN) and transport (QUIC parameters) values are known in advance

Labels potentially leak sensitive information (known issue for PSK identities)

No specific details for TLS 1.2. Should that change?

Universal PSKs

Abstract the TLS 1.3 requirement that each PSK be used with at one hash function [1]

Derive a binder key from the universal PSK and identity

Upon PSK and ciphersuite negotiation, derive a hash-specific PSK to use in the key schedule

[1] https://datatracker.ietf.org/doc/draft-davidben-tls-universal-psk/

Comparison

PSK importers (unnecessarily?) inflate the size of CHs based on the number of supported PSKs

Universal PSKs modify the existing TLS 1.3 key schedule (in a seemingly safe way)

PSKs and Privacy

PSKs must have meaningful and deterministic labels for clients and servers to associate with secret keys

Tickets (and labels) allow servers to track clients across resumption (or connection) attempts

- Proposition #1: do not resume if there's no early data to write
- Proposition #2: use semi-static DH secret to anonymously encrypt early data [1]
- Proposition #3: use VOPRFs [2] to *derive* resumption PSKs

Is this something the WG should focus on?

[1] https://datatracker.ietf.org/doc/draft-rescorla-tls-semistatic-dh/
[2] https://datatracker.ietf.org/doc/draft-sullivan-cfrg-voprf/

Questions?