draft-urien-tls-keygen-00.txt

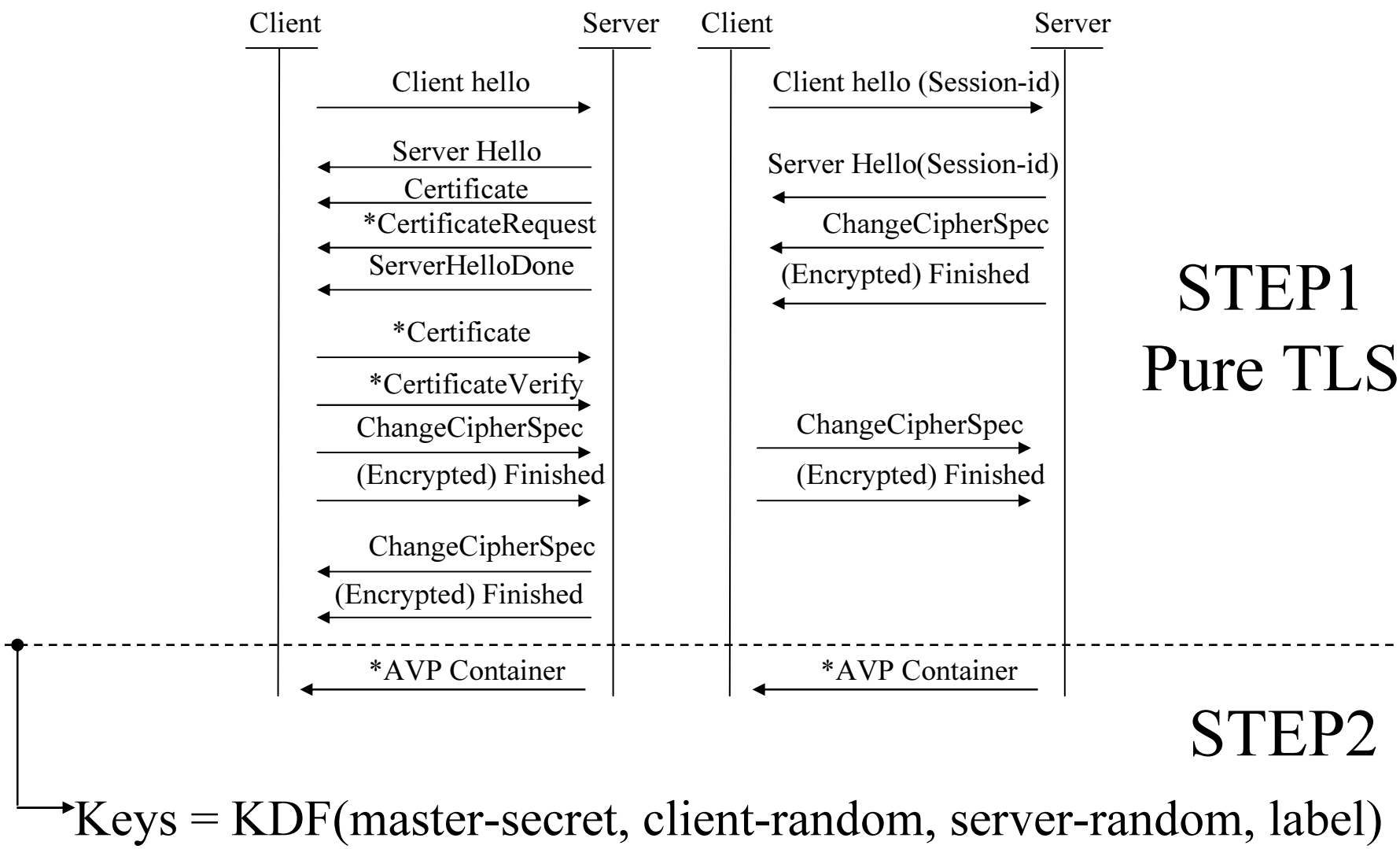
TLS Key Generation

Pascal.Urien@telecom-paristech.fr

Summary

- **The TLS protocol is widely deployed and used over the Internet.**
  - Binary encoding, wide adoption
- **There is an increasing need in the Internet to set up efficient key distribution infrastructures.**
- **This draft proposes a keying infrastructure based on the TLS protocol.**
- **Differences with the draft-ietf-tls-extractor**
  - Computed keys
    - In draft-ietf-tls-extractor-01.txt the TLS PRF function is used
    - In draft-urien-tls-keygen-00.txt a separate KDF function is used
  - Pushed keys
    - The point that is not addressed by draft-ietf-tls-extractor-01.txt, is the case for which keys are pushed by server and not computed by both parties (client and server)

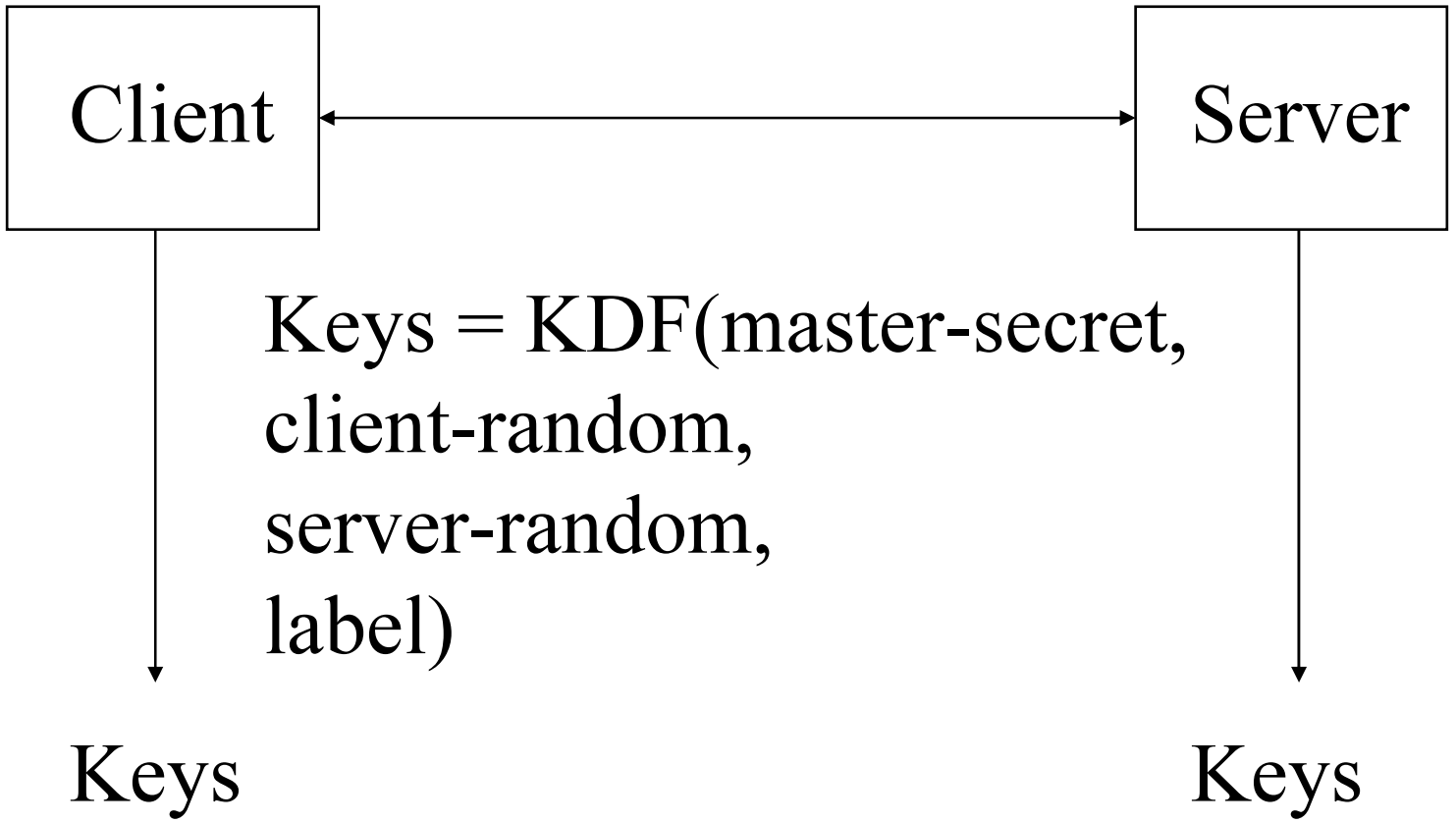
Full and abbreviated mode choreography



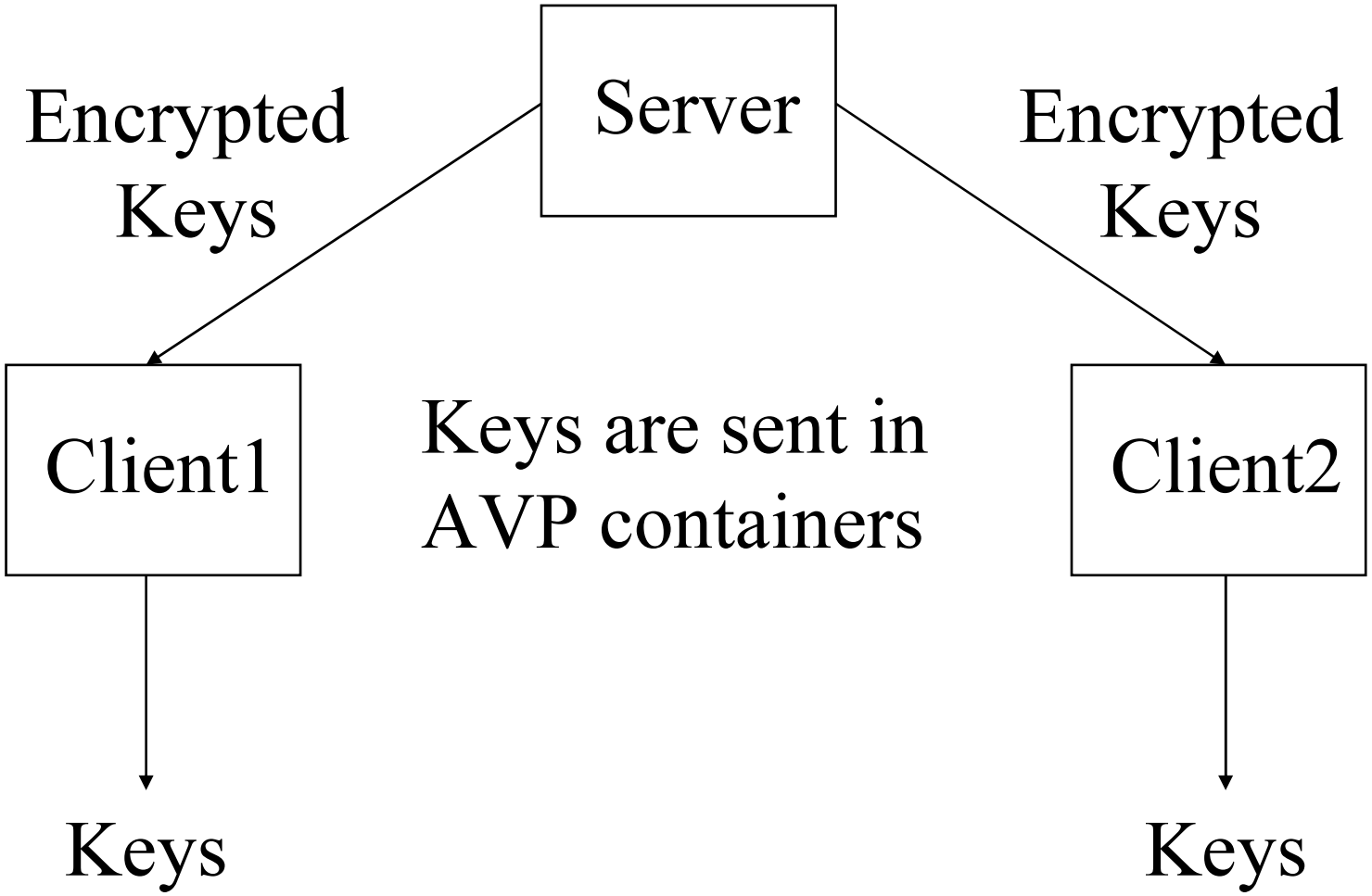
Full and abbreviated mode choreography

- **What is available at the end of STEP1.**
  - Client-random, Server-random, master-secret, negotiated cipher-suite
  - Implicit KDF function, with an implicit label
    - Other KDF functions MAY be negotiated via TLS-Extensions
  - Cryptographic keys (Kc and Ki), used by encryption algorithms (Kc) and MAC procedures (Ki), are derived according to TLS specifications, but use KDF in place of the TLS PRF function.
    - Keys = KDF(master-secret, client-random, server-random, label)
- **What MAY be done during STEP2.**
  - Keys MAY be sent encrypted in an AVP container
  - A new label (for the KDF function) MAY be sent in an AVP container.

Peer to Peer Mode

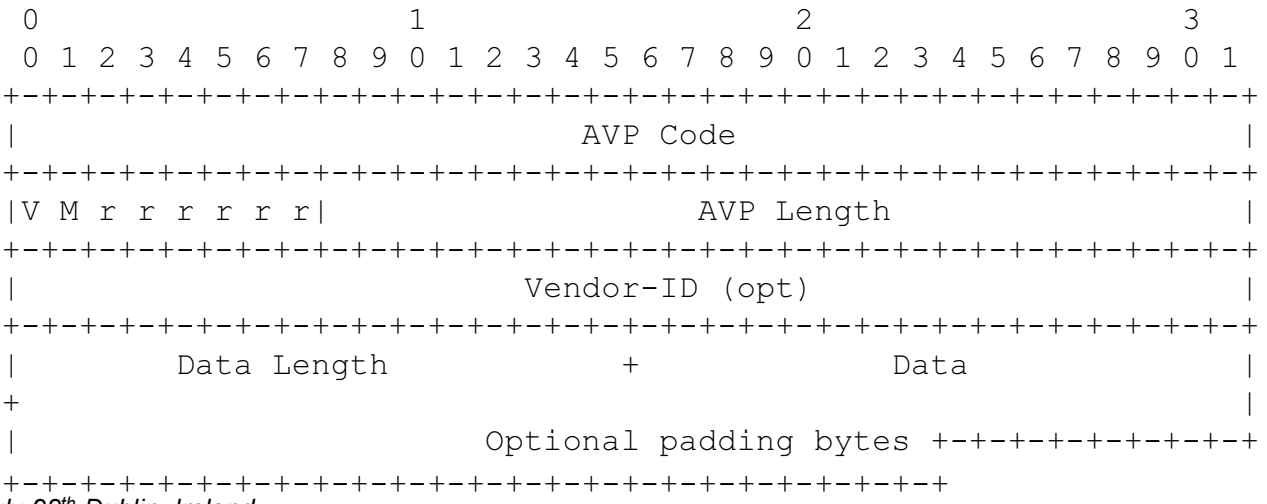


Distributed Mode



AVP Container Coding (imported from TTLS)

- The AVP Code is four octets
  - Combined with the Vendor-ID field if present, identifies the attribute (i.e. the container structure) uniquely.
- The 'V' (Vendor-Specific) bit indicates whether the Vendor-ID field is present.
- The 'M' (Mandatory) bit indicates whether support of the AVP is required.
- The 'r' bits are unused and set to 0 by the sender.
- The AVP Length field is three octets, and indicates the length of this AVP including the AVP Code, AVP Length, AVP Flags, Vendor-ID (if present) and Data.
- Data Length is two octets and indicates the size of data without padding bytes.



KDF Consideration

- Based on the work
  - Krawczyk, H, "On Extract-then-Expand Key Derivation Functions and an HMAC-based KDF", http://www.ee.technion.ac.il/~hugo/kdf/, March 2008
- Already use in IKEV2

For Transform Type 2 (Pseudo-random Function), defined Transform IDs are:

Name	Number	Defined In
PRF_HMAC_SHA1	2	RFC 2104
PRF_HMAC_TIGER	3	RFC 2104
PRF_AES128_XCBC	4	RFC 3664

Keying material is generated according to the selected prf

prf+ (K, S) = T1 | T2 | T3| ..., where

T1 = prf (K, S | 0x01), T2 = prf (K, T1 | S | 0x02), T3 = prf (K, T2 | S | 0x03)

SKEYSEED = prf (Ni | Nr, g^ir) , SK_d = **prf+** (SKEYSEED, Ni | Nr | SPIi | SPIr)

KEYMAT = **prf+** (SK_d, Ni | Nr) or KEYMAT = **prf+** (SK_d, g^ir | Ni | Nr)

Proposed default KDF

- **Notations.**
  - The first argument to a keyed function denotes the key, the value K is the key to PRF and x its input.
  - The symbol || denotes concatenation.
  - Given two numbers N and n the symbol N:n represents the value N written as n-bit integer.
  - L is the length in bits, of this output value delivers by the HMAC procedure (L=160 for SHA1)
- **Pseudo Random Key**
  - PRK = HMAC (client-random || server-random, master-secret)
- **Keying material, whose length in bits is D, required D/L operations.**
  - First is expressed as
    - K(1) = HMAC(PRK, 0:L || KeyLabel || 0:32),
  - Further operations (whose number is i) are computed according to
    - K(i+1) = HMAC(PRK, K(i) || KeyLabel || i:32)
    - where KeyLabel is an ASCII string set to "key expansion".

Questions ?