



Aalto University

Network Security: TLS 1.3 0-RTT handshake

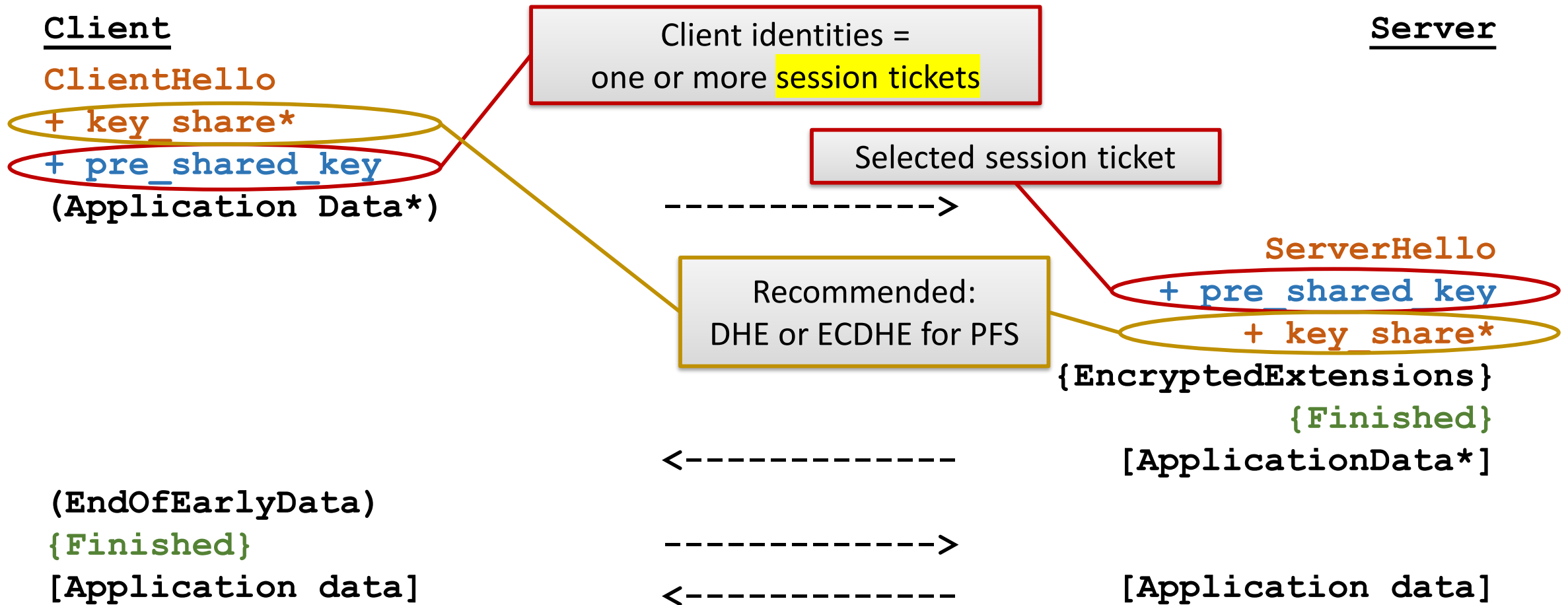
Tuomas Aura, Aalto University

CS-E4300 Network security

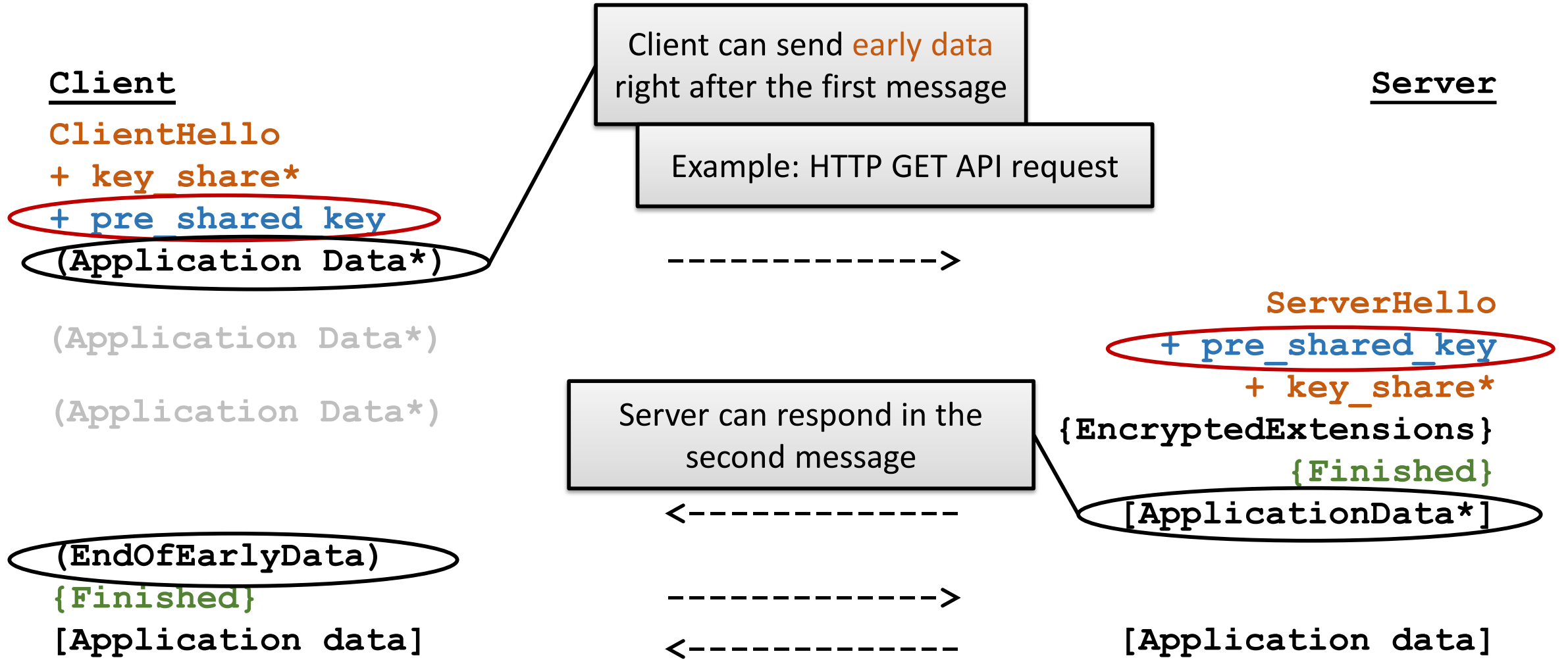
Outline

- Recall PSK handshake and session resumption
- 0-RTT handshake

TLS 1.3 session resumption



0-RTT handshake



Key derivation

Inputs to key derivation:

1. PSK (external PSK or resumption PSK)
 2. DHE/ECDHE secret
 3. Transcript of handshake messages, up to the point where the key is derived
- } one or both, as available

Keys:

- **client_early_traffic_secret** → used to derive AEAD keys for early data in 0-RTT (...)
- **client/server_handshake_traffic_secret** → used to derive AEAD keys for handshake messages {...} and Finished HMAC keys
- **client/server_application_traffic_secret_N** → used to derive AEAD encryption keys for post-handshake application data and messages [...]
- **resumption_master_secret** and **ticket_nonce** → derive resumption PSK
- **exporter_master_secret** → used to create keys for the application layer

0-RTT handshake

- With session resumption or PSK, client can send application data (early data) right after ClientHello
 - Lower latency for web browsing and APIs. However, TCP handshake in the underlying transport layer still takes one RTT
- **Serious security limitations:**
 - Early data is vulnerable to replay attacks (no fresh server nonce yet)
 - No PFS for the early data
- Ok for **idempotent requests** (mainly HTTP GET) that do not require long-term secrecy
- **Application must decide** whether to enable 0-RTT
 - TLS layer cannot decide when the lower security is ok