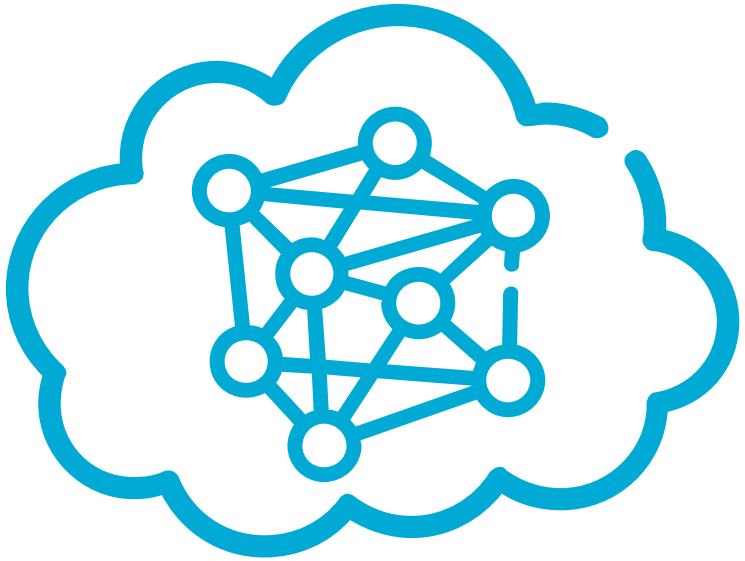
TLS SESSION KEY INTERFACE

DRAFT-CAIRNS-TLS-SESSION-KEY-INTERFACE-00

KELSEY CAIRNS, WASHINGTON STATE UNIVERSITY
JOHN MATTSSON, ROBERT SKOG, ERICSSON

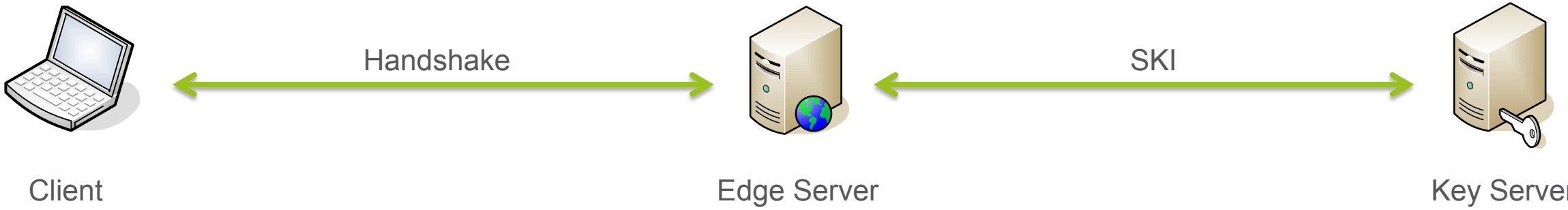
BACKGROUND

- Heartbleed has illustrated the security problems with storing private keys in the memory of the TLS server.
- HSMs provide good security, but are inflexible and difficult to deploy when TLS servers run virtualized in the cloud.
  - Especially if the application server that uses TLS moves between different data centers.
  - HSMs do not protect against misuse if an adversary gains possession of the HSM itself.
- Network-attached HSMs is a solution but they use proprietary network protocols tied to the specific HSM vendor.
  - There are several other proprietary session key interfaces deployed but no standardized solution.
- A standardized TLS Session Key Interface is needed.

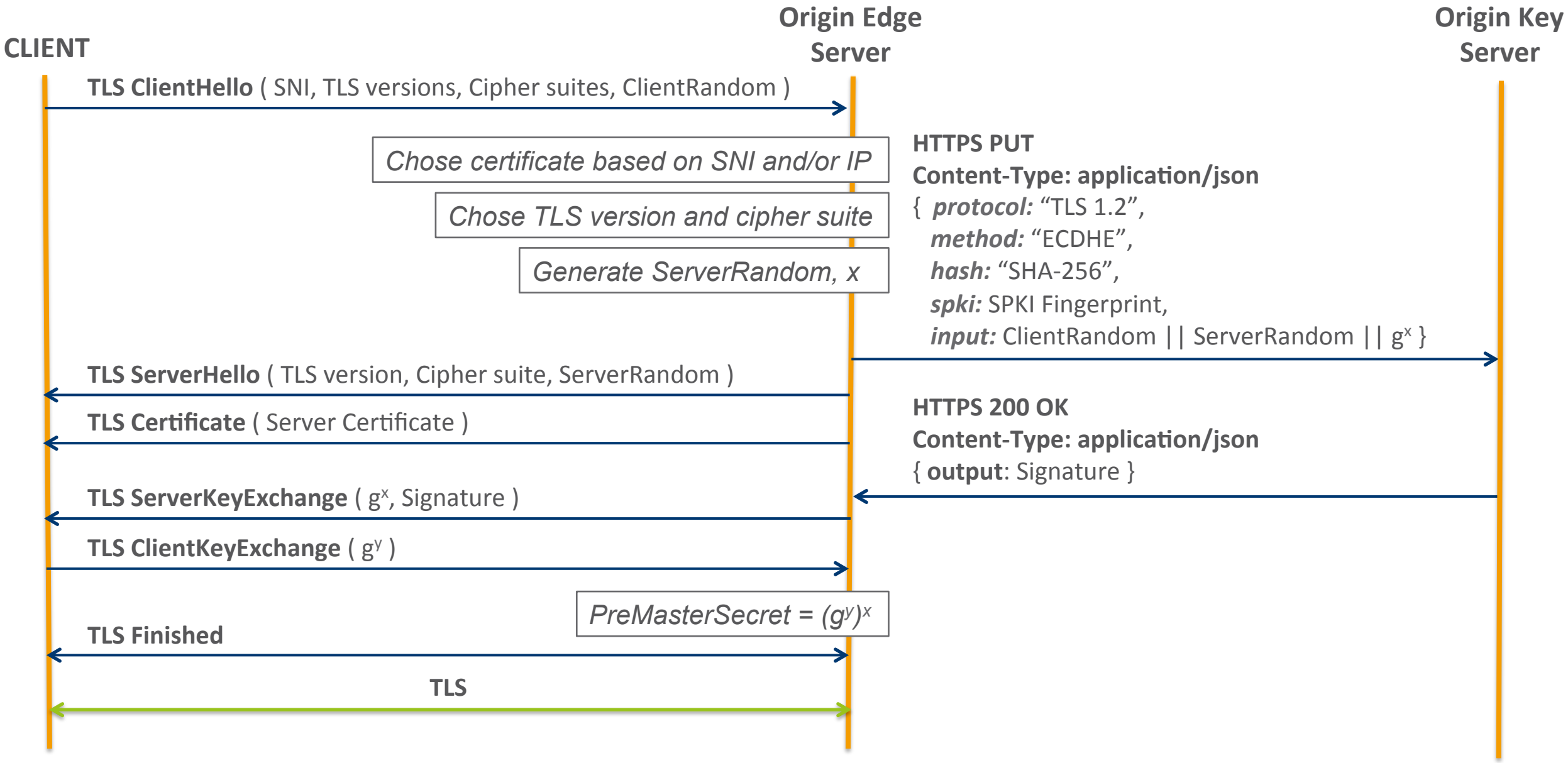


TLS SESSION KEY INTERFACE (SKI)

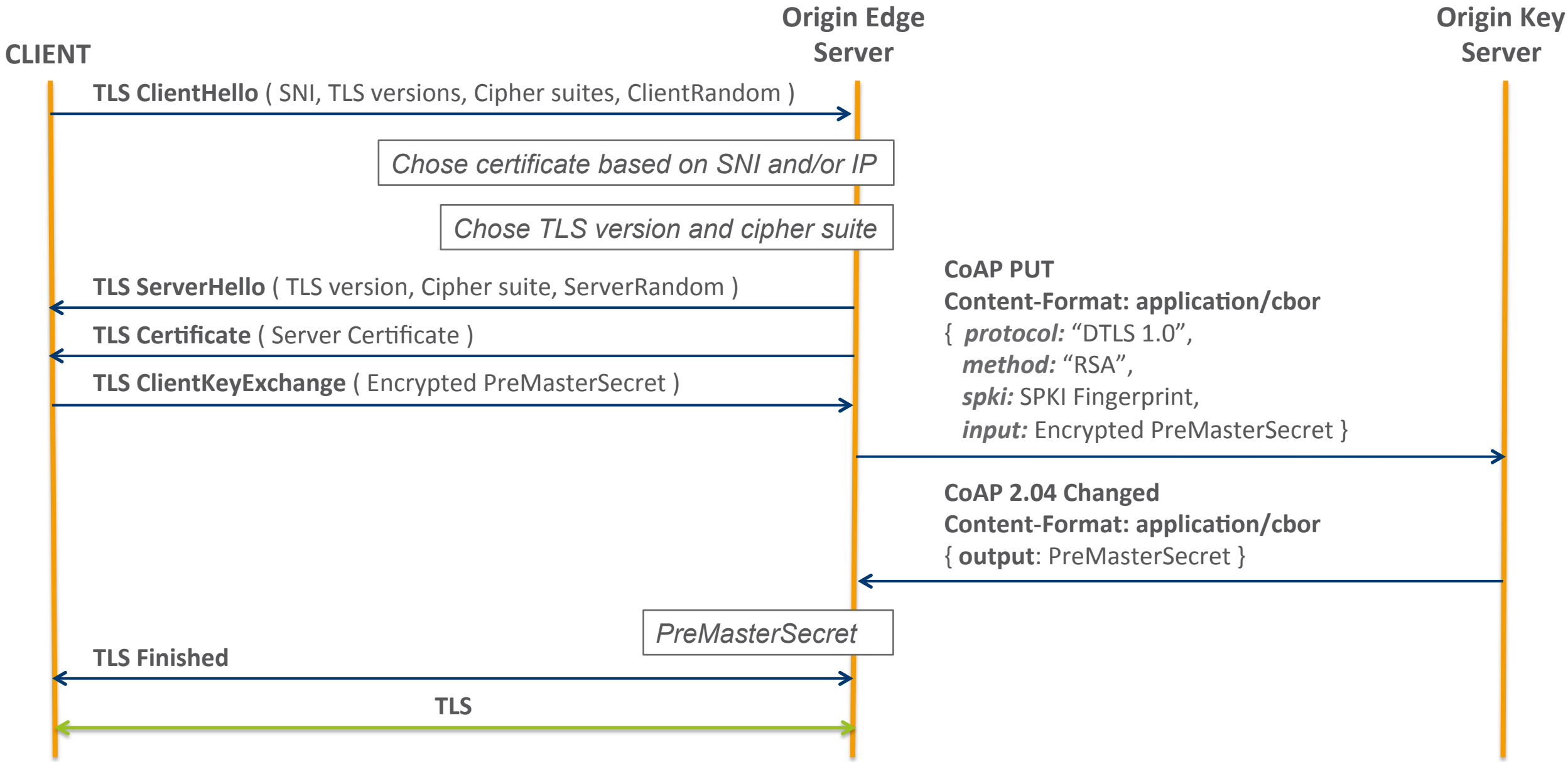
- With TLS SKI the TLS server is split into two distinct entities called Edge Server and Key Server that communicate over an encrypted and mutually authenticated channel.
  - The public certificates (**not private keys**) are pre-provisioned in the Edge Server. The private keys are stored in a highly trusted key server, separated from client facing servers.
  - This increases the security by shrinking the attack surface and reducing damage if the Edge Server is compromised.
  - It also enables secure handling of TLS connections in the cloud, and the Edge Server can be placed close to the clients, reducing latency.
- The interface uses modern web technologies like JSON, CBOR, HTTP, CoAP, TLS, and REST.
- SKI supports the most commonly used key exchange methods ECDHE_ECDSA, ECDHE_RSA, and RSA, together with X.509 or raw public key authentication.



ECDHE KEY EXCHANGE



RSA KEY EXCHANGE



SESSION KEY DERIVATION

- With PreMasterSecret, ClientRandom, and ServerRandom, the Origin Edge Server can derive all the different TLS session keys.