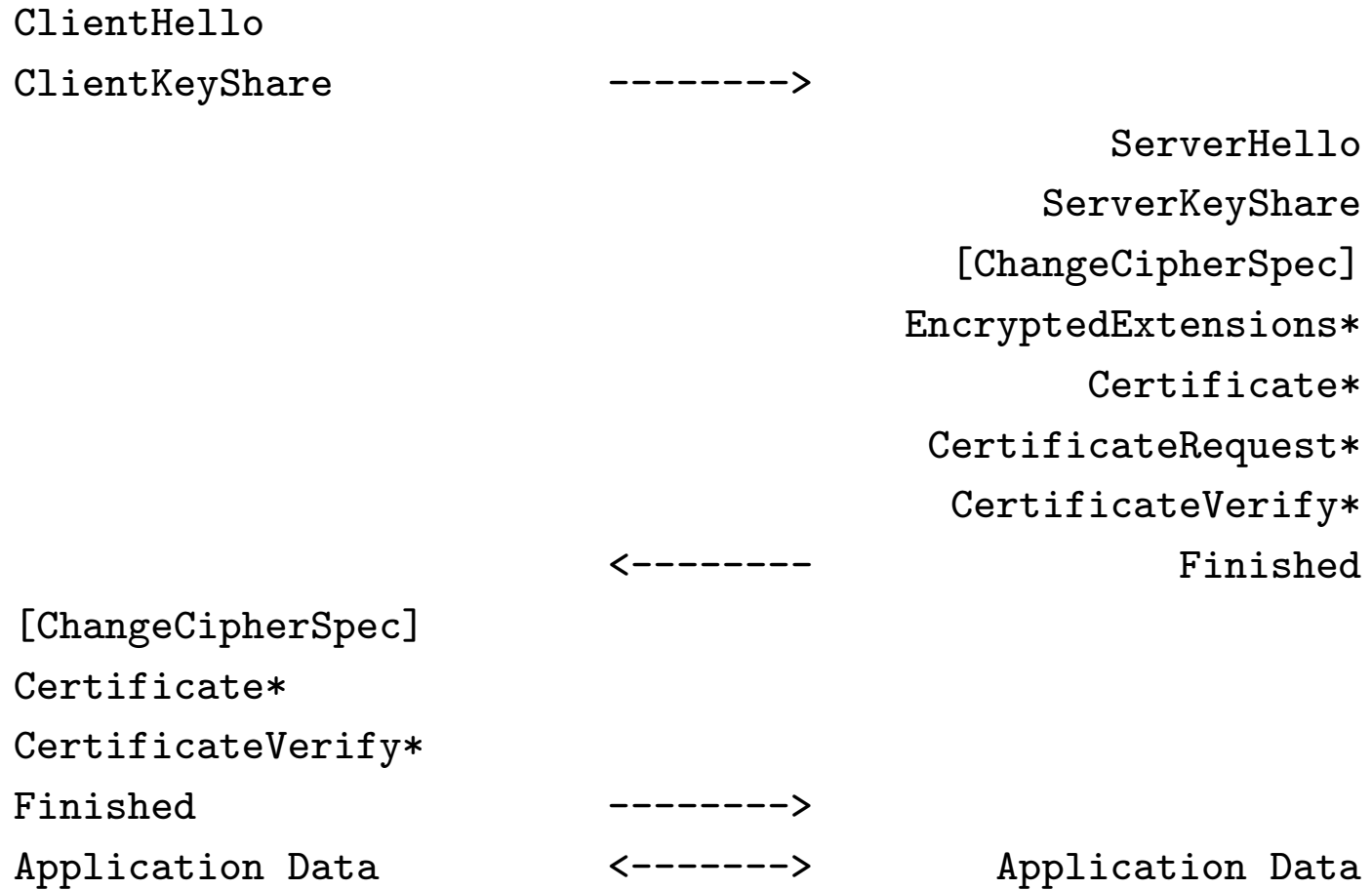
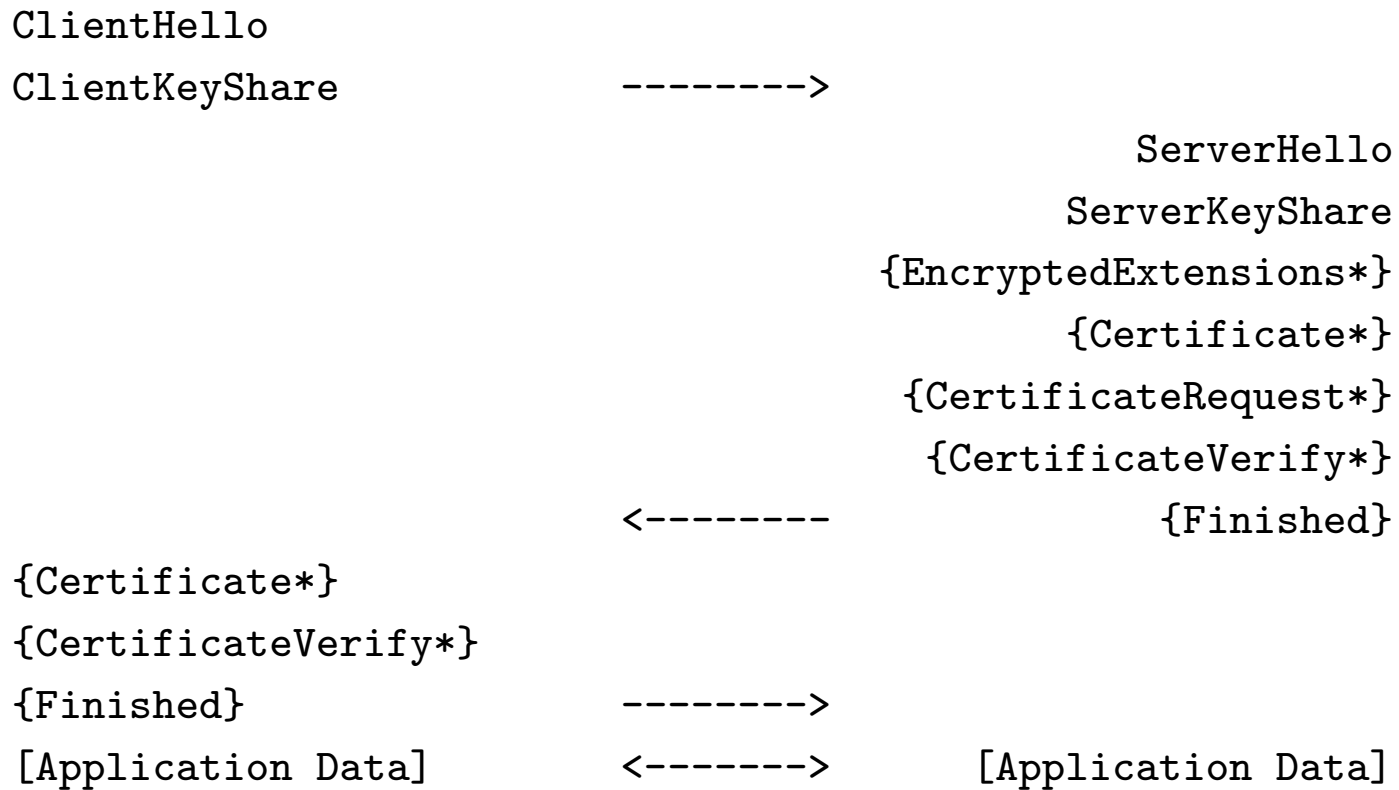


Existing Draft



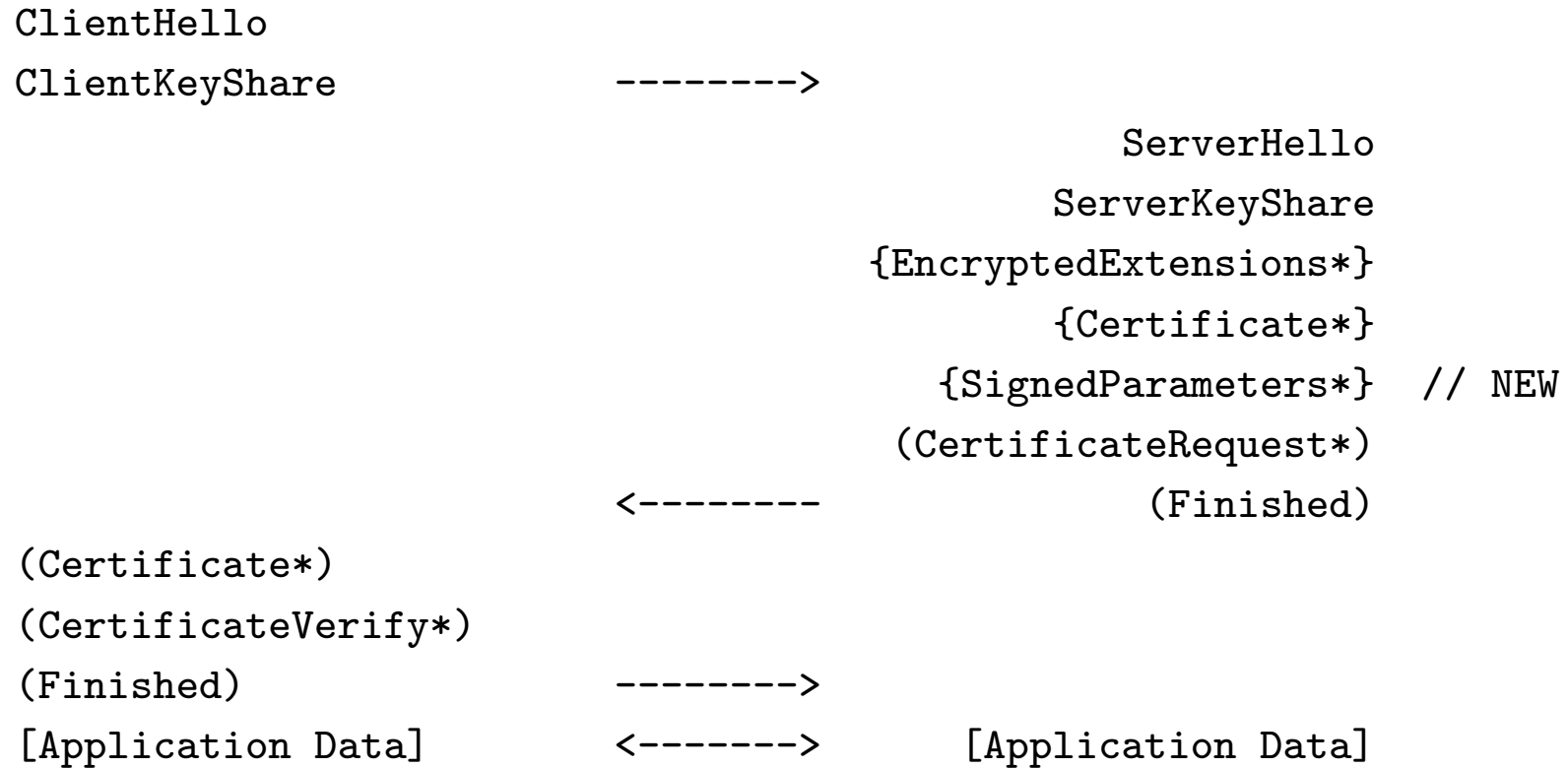
Session Hash Draft



{ } Indicates messages protected using keys derived from the handshake master secret.

[ ] Indicates messages protected using keys derived from the master secret.

Hugo's Proposal (slightly adapted, server-side only)



{ } encrypted under  $g^{xy}$

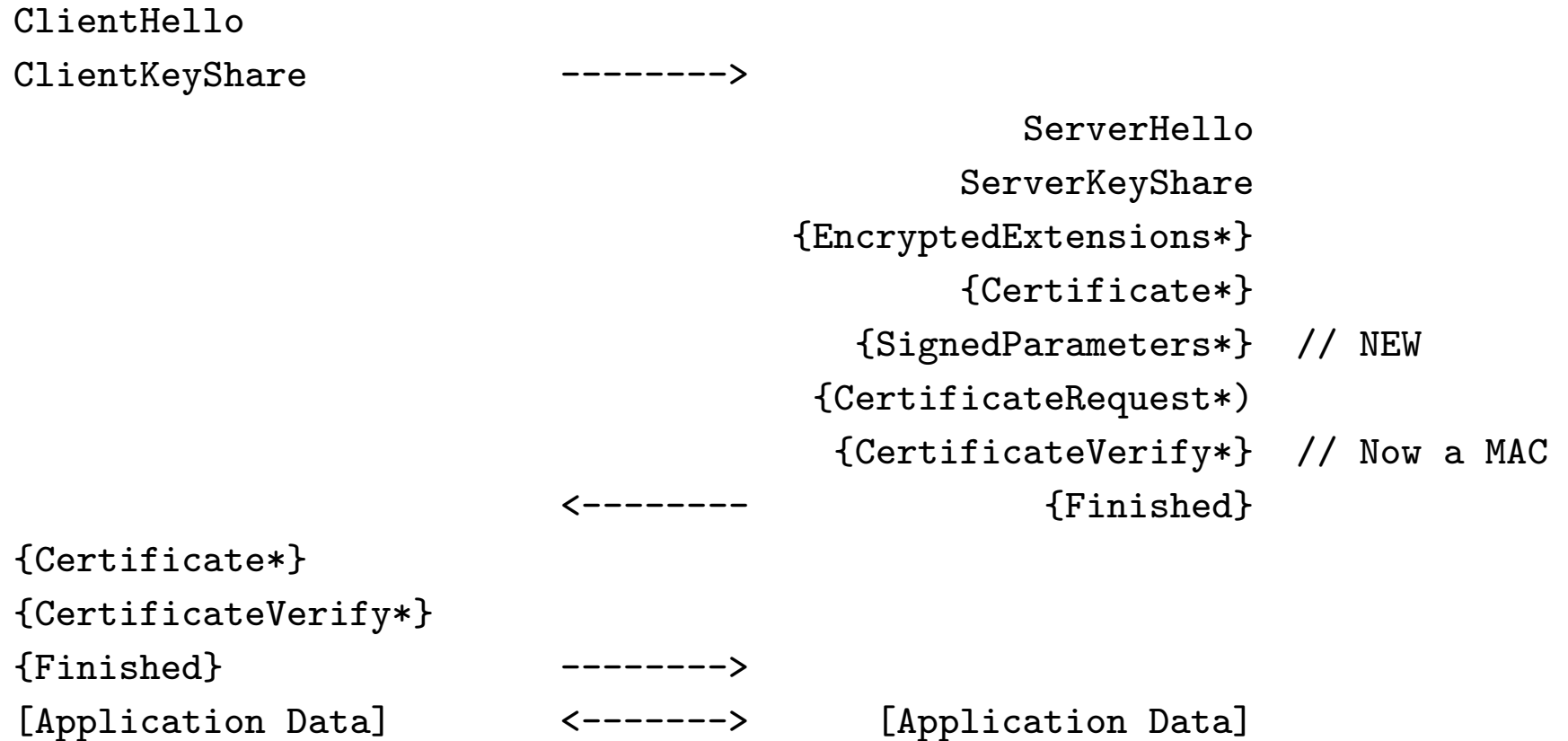
() encrypted under  $g^{xy}$ ,  $g^{xs}$  (handshake key)

[] encrypted under  $g^{xy}$ ,  $g^{xs}$  (application key)

```
struct {
    opaque identifier[16];
    uint64 not_before;
    uint64 not_after;
    NamedGroup group;
    opaque key_exchange;
} UnsignedParameters;
```

```
struct {
    UnsignedParameters parameters;
    digitally-signed struct {
        opaque zeros[64];
        UnsignedParameters parameters;
    };
} SignedParameters;
```

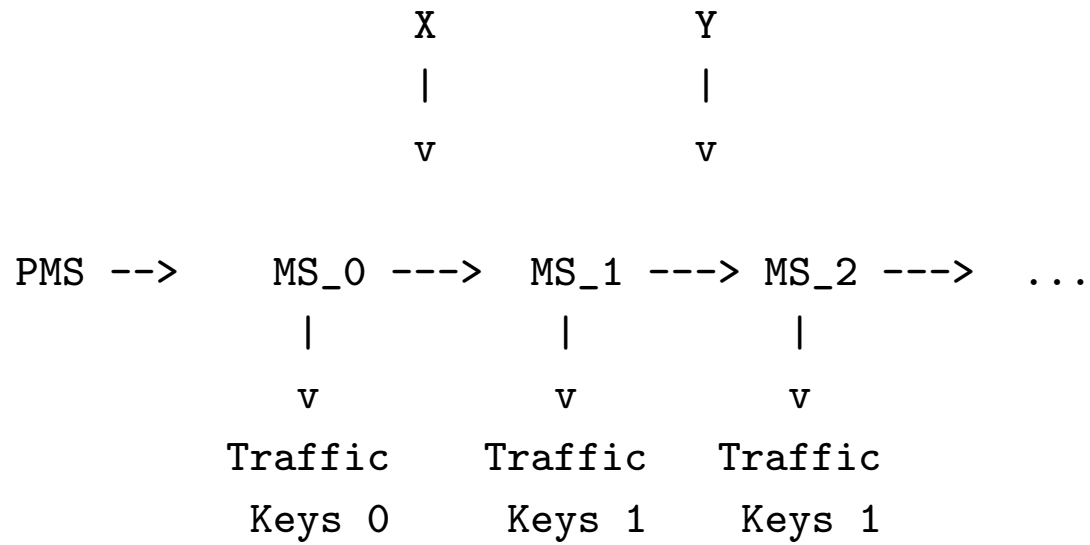
Variant of Hugo's proposal (server side only)

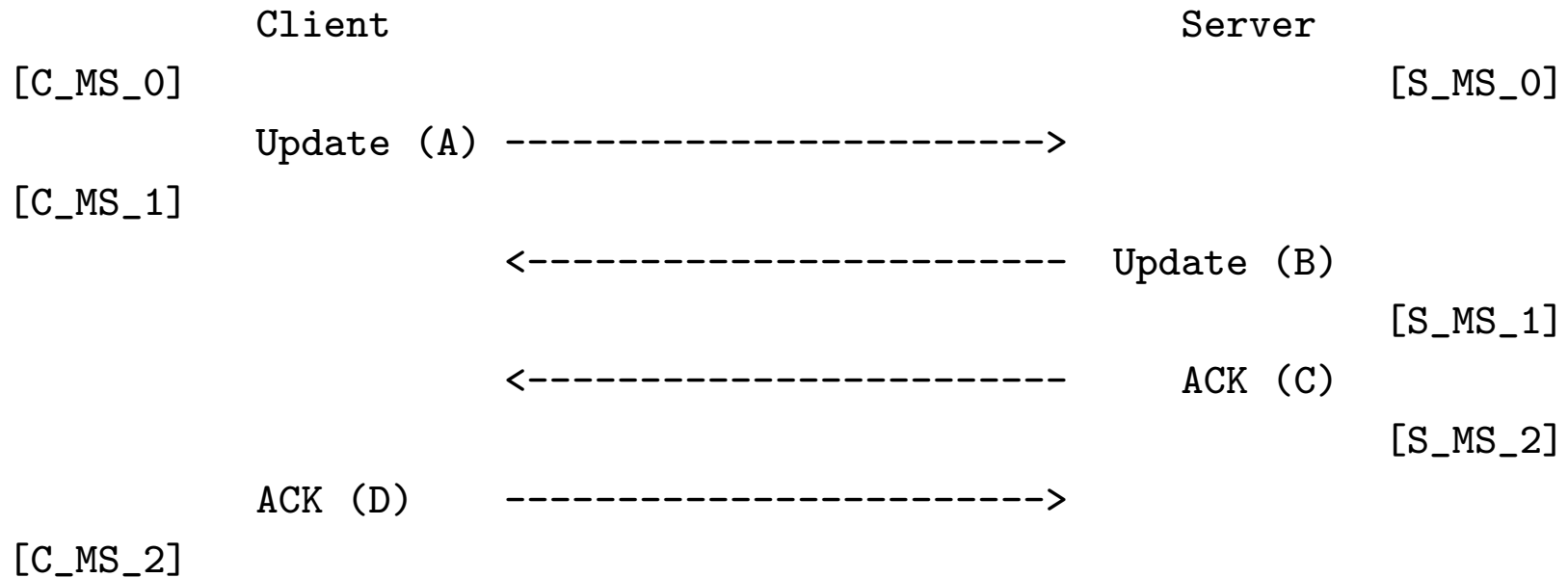


Server CertificateVerify is computed as  $\text{HMAC}(g^{xs}, \text{transcript})$

{ } encrypted under the handshake master secret ( $g^{xy}$ )

[] encrypted under the master secret ( $g^{xy} + g^{xs}$ )





The key computations are as follows:

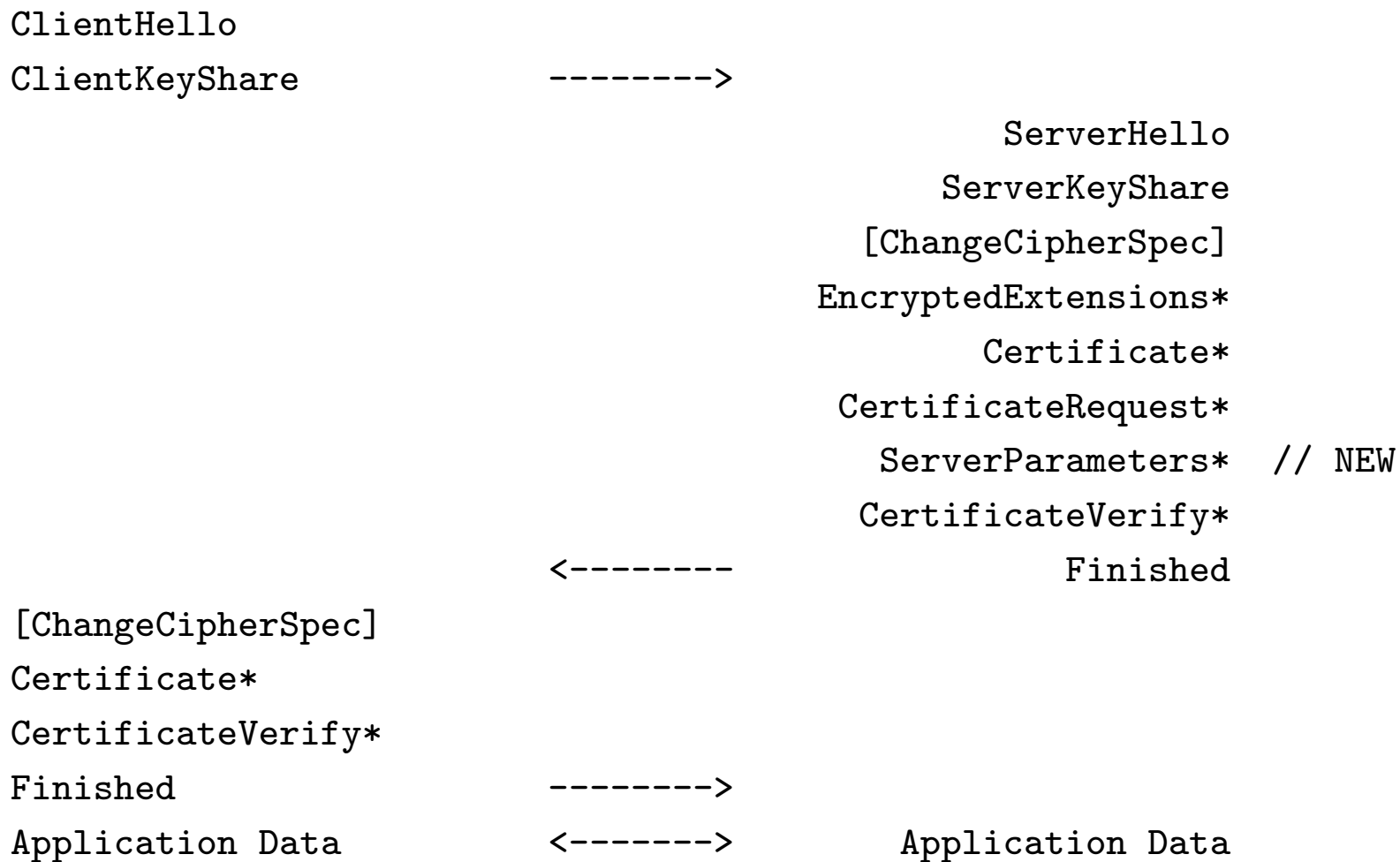
$$C\_MS\_1 = \text{PRF}(C\_MS\_0, A)$$

$$S\_MS\_1 = \text{PRF}(S\_MS\_0, B)$$

$$C\_MS\_2 = \text{PRF}(C\_MS\_1, D) \quad [D \text{ depends only on } B]$$

$$S\_MS\_2 = \text{PRF}(S\_MS\_1, C) \quad [C \text{ depends only on } A, \text{ but } S\_MS\_2 \text{ depends on } S\_MS\_1 \text{ which depends on } B].$$

## First Handshake for 0-RTT





0-RTT Handshake (very Sketchy, no client auth)

ClientHello

+ PredictedParameters

ClientKeyExchange

{EncryptedExtensions + AntiReplayToken}

{Finished}

{ApplicationData} ----->

ServerHello

ServerKeyExchange

[ChangeCipherSpec]

{EncryptedExtensions  
+ AntiReplayToken}

{Certificate\*}

{CertificateRequest\*}

{ServerParameters\*}

{CertificateVerify\*}

<----- {Finished}

{Finished???

----->

Application Data

<----->

Application Data