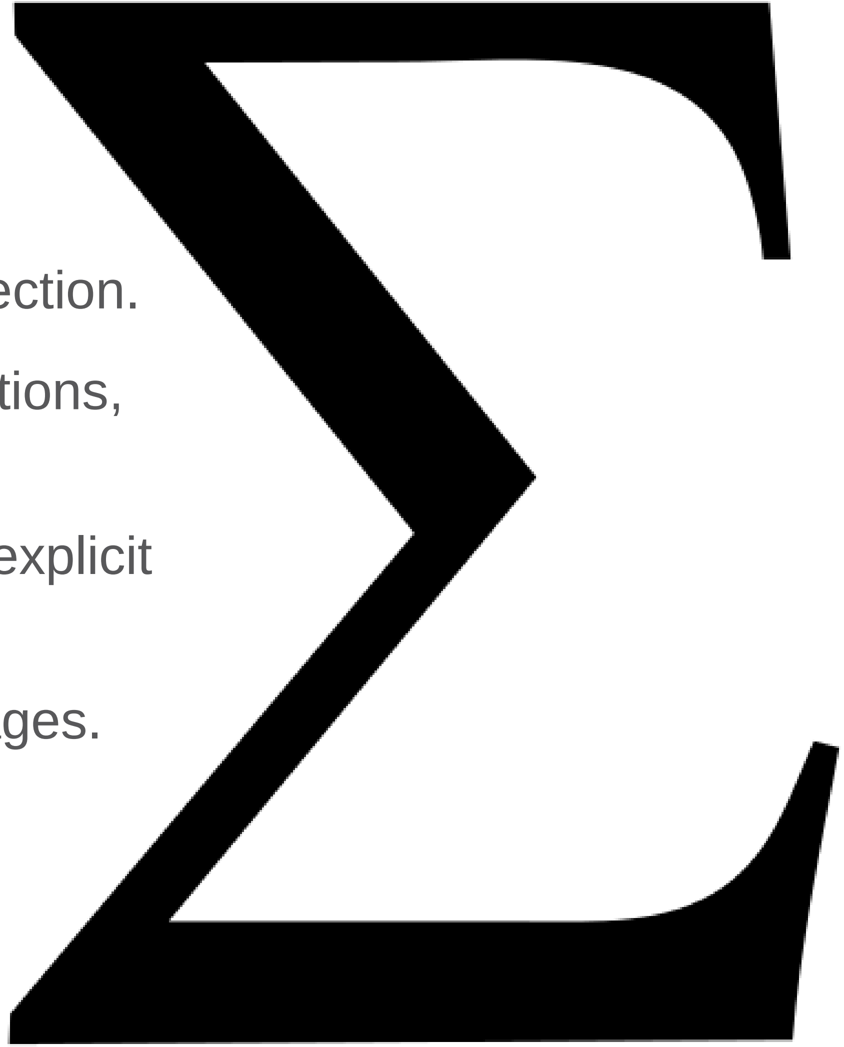
ERICSSON

Ephemeral Diffie-Hellman Over COSE (EDHOC)

draft-selander-ace-cose-ecdhe-05
SELANDER, MATTSSON, PALOMBINI
IETF98 ACE, MAR 27 2017

NEW IN VERSION -05

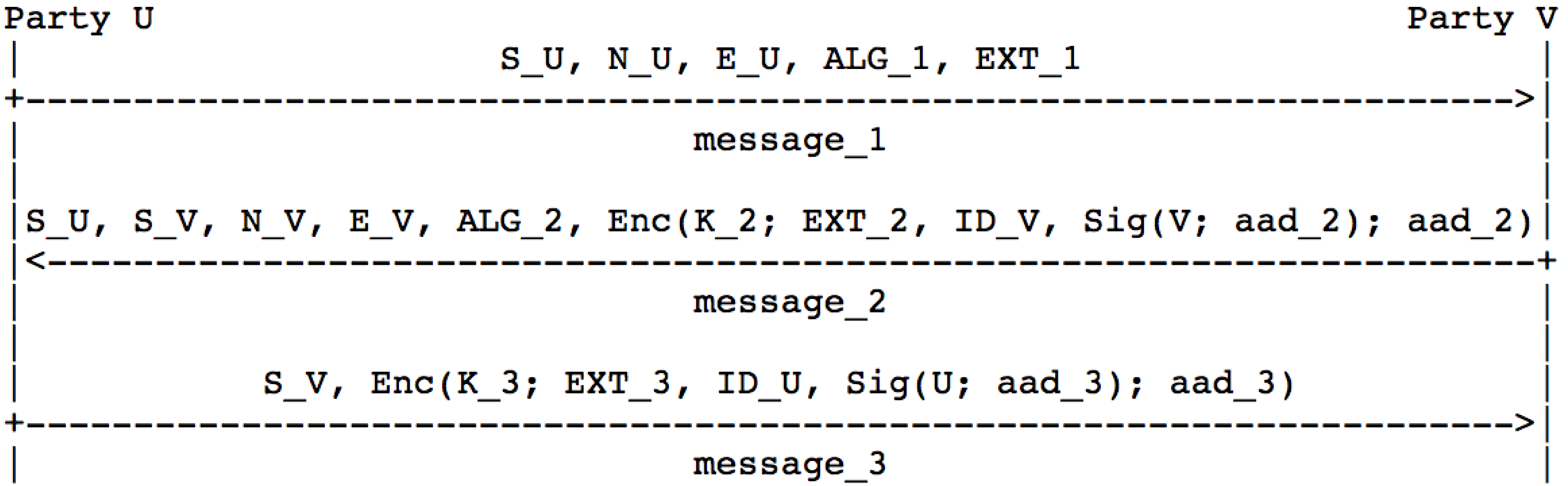
- Many smaller changes:
  - Simplified both protocol and protocol specification.
  - Added explicit session identifiers, different in each direction.
  - Added explicit extensions that can be used by applications, e.g. for authorization tokens.
  - All EDHOC messages are now CBOR arrays with an explicit message type.
  - MACs and key derivation bound to all previous messages.
  - Simplified and strengthened key derivation.
  - Hash previous messages to save memory.



EDHOC with Asymmetric Keys

- The parties exchanging messages are called "U" and "V". U and V exchange identities and ephemeral public keys. They compute the shared secret and derive the keying material.

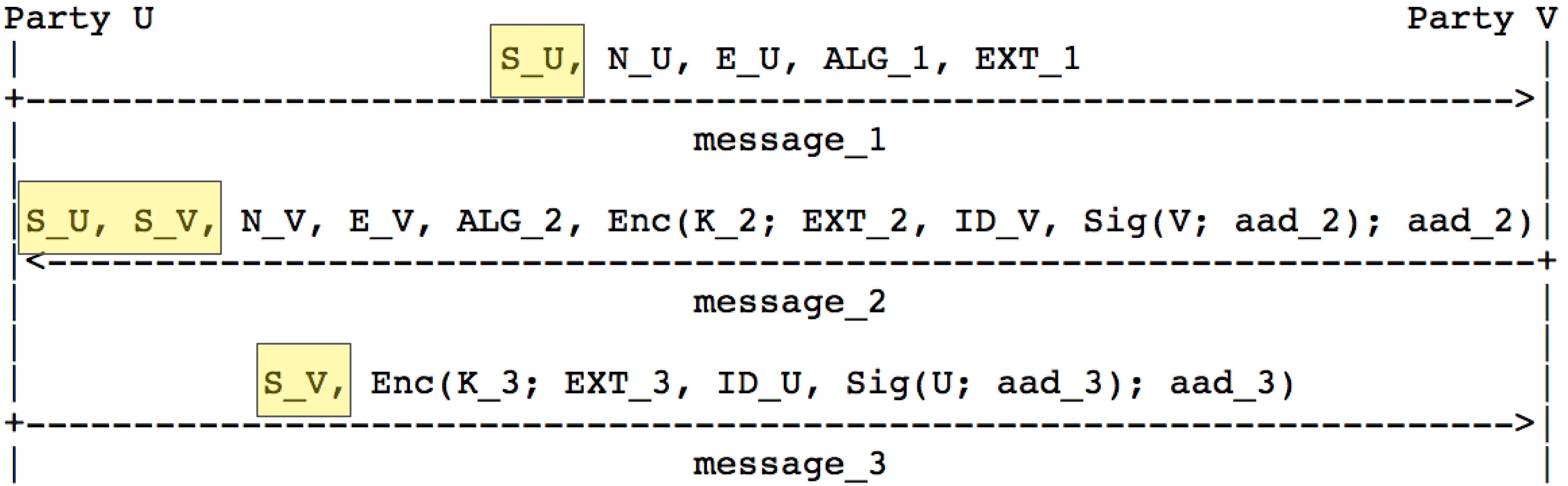
- All EDHOC messages are now CBOR arrays with an explicit message type.



```
Party U                                                   Party V
|                S_U, N_U, E_U, ALG_1, EXT_1                    |
+-------------------------------------------------------------->|
|                          message_1                            |
|                                                               |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<--------------------------------------------------------------+
|                          message_2                            |
|                                                               |
|       S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)        |
+-------------------------------------------------------------->|
|                          message_3                            |
```

EDHOC with Asymmetric Keys

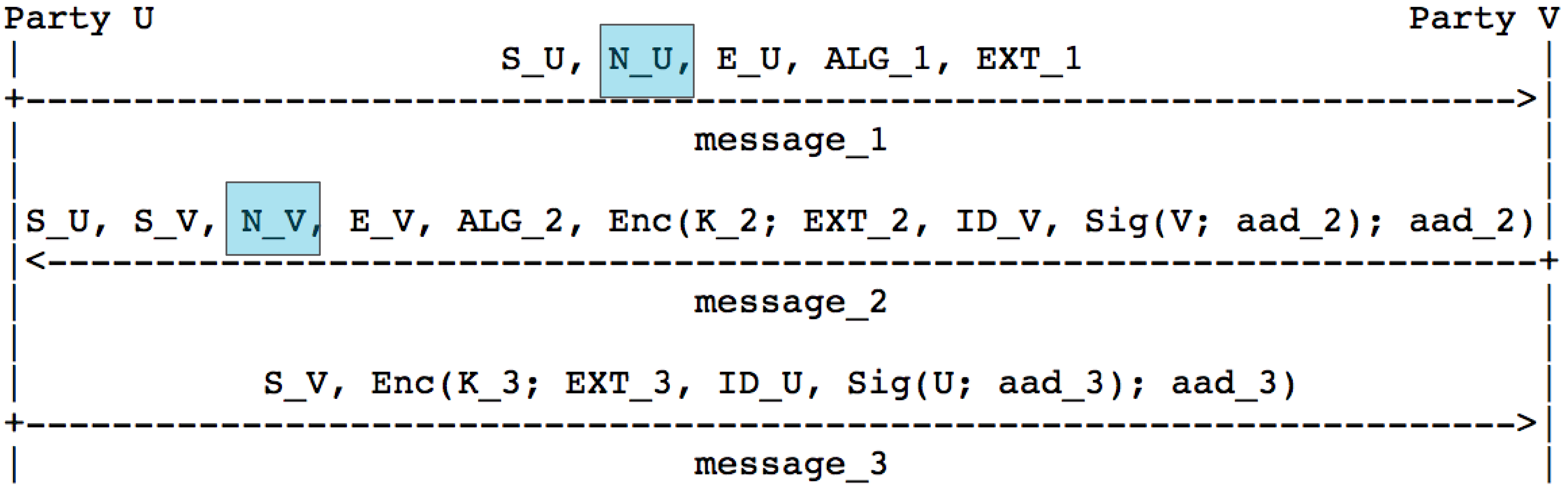
- Two explicit session identifiers S_U and S_V (one for each direction).
- If EDHOC is used for OSCOAP, S_U and S_V are reused as identifiers in OSCOAP.



```
Party U                                                           Party V
|                S_U, N_U, E_U, ALG_1, EXT_1                            |
+---------------------------------------------------------------------->|
|                          message_1                                    |
|                                                                       |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<----------------------------------------------------------------------+
|                          message_2                                    |
|                                                                       |
|          S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)             |
+---------------------------------------------------------------------->|
|                          message_3                                    |
```

EDHOC with Asymmetric Keys

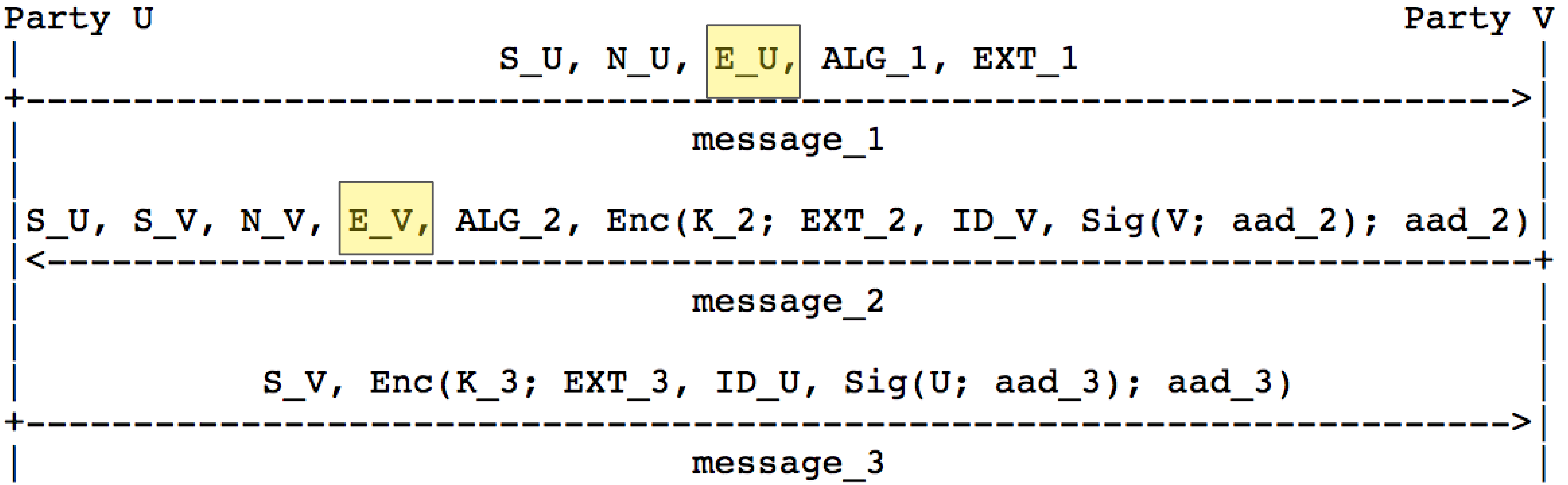
- Two explicit nonces N_U and N_V



```
Party U                                                       Party V
|                        S_U, N_U, E_U, ALG_1, EXT_1                |
+------------------------------------------------------------------>|
|                                message_1                          |
|                                                                   |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<------------------------------------------------------------------+
|                                message_2                          |
|                                                                   |
|           S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)        |
+------------------------------------------------------------------>|
|                                message_3                          |
```

EDHOC with Asymmetric Keys

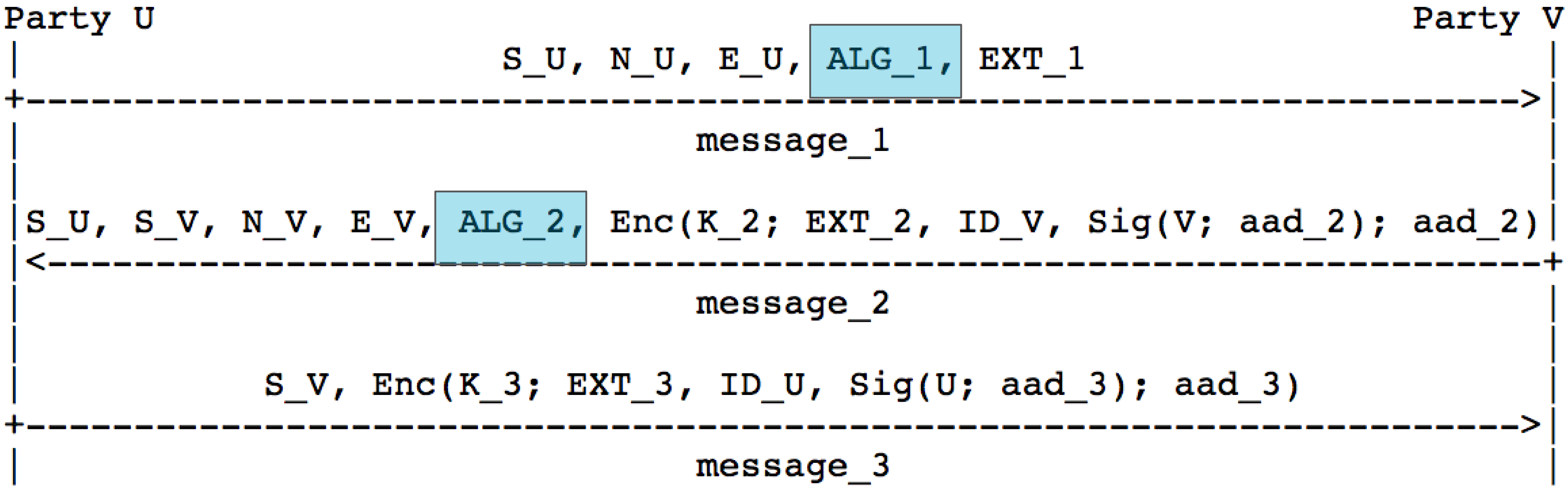
- Two ephemeral public keys E_U and E_V



```
Party U                                                      Party V
|              S_U, N_U, E_U, ALG_1, EXT_1                         |
+----------------------------------------------------------------->|
|                         message_1                                |
|                                                                  |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<-----------------------------------------------------------------+
|                         message_2                                |
|                                                                  |
|       S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)           |
+----------------------------------------------------------------->|
|                         message_3                                |
```

EDHOC with Asymmetric Keys

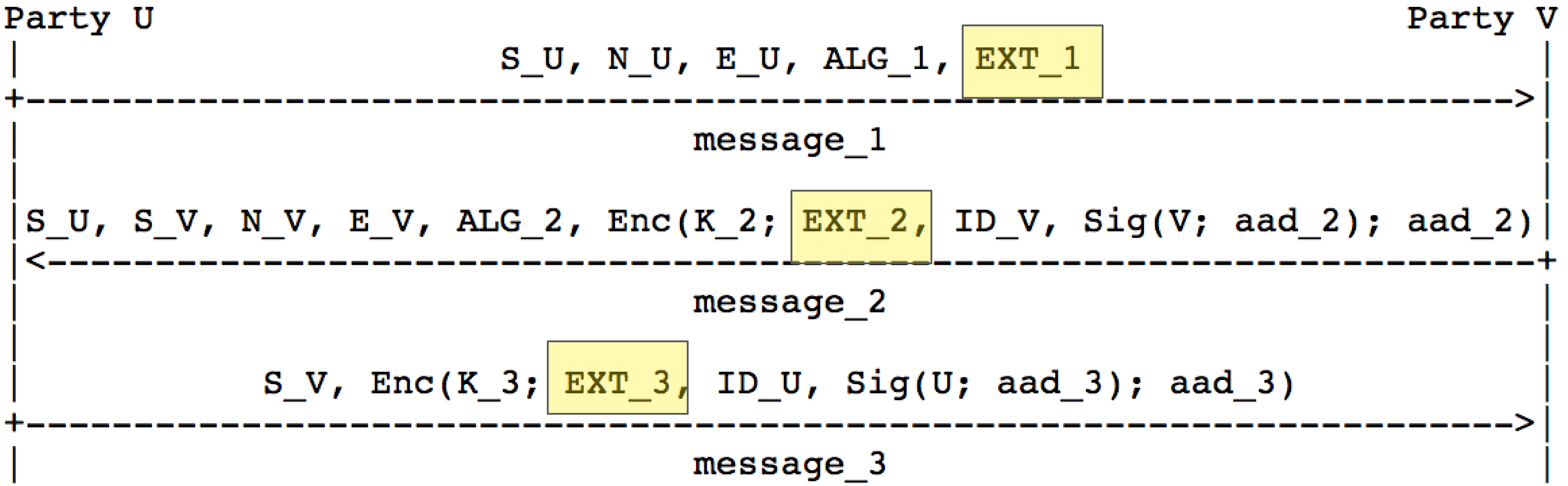
- Algorithm negotiation ALG_1 and ALG_2
- Four algorithms negotiated: HKDF, AEAD, and two signature algorithms.



```
Party U                                                          Party V
|                      S_U, N_U, E_U, ALG_1, EXT_1                     |
+--------------------------------------------------------------------->|
|                               message_1                              |
|                                                                      |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<---------------------------------------------------------------------+
|                               message_2                              |
|                                                                      |
|           S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)           |
+--------------------------------------------------------------------->|
|                               message_3                              |
```

EDHOC with Asymmetric Keys

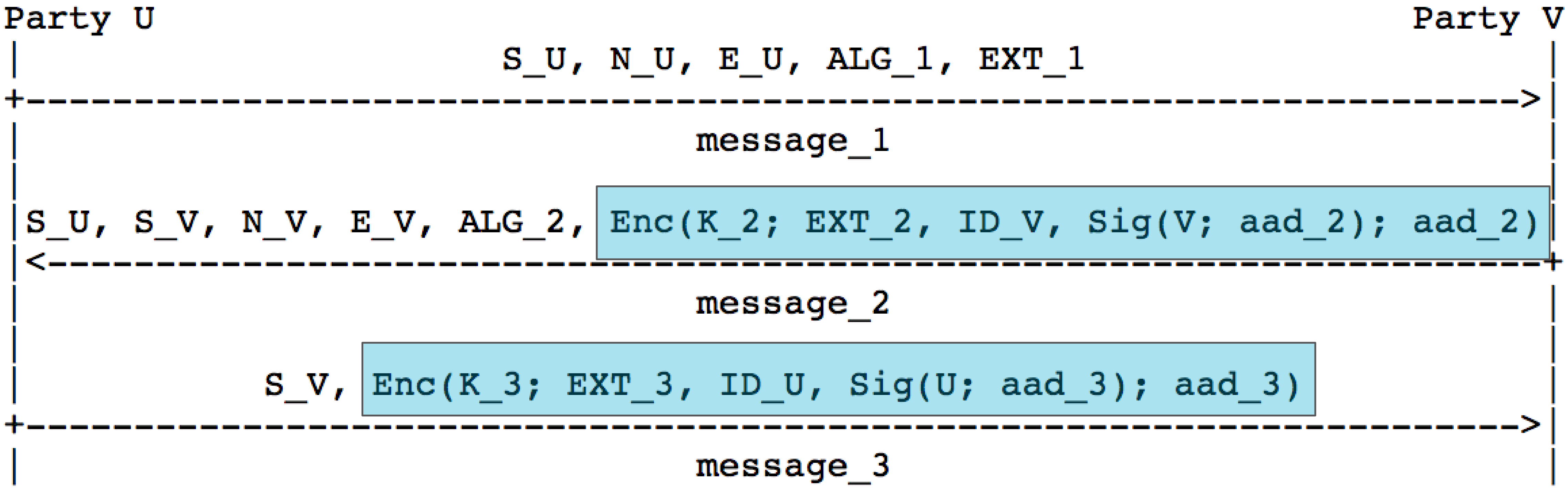
- Explicit application defined extensions, used e.g. authorization tokens.



```
Party U                                                         Party V
|                   S_U, N_U, E_U, ALG_1, EXT_1                       |
+-------------------------------------------------------------------->|
|                            message_1                                |
|                                                                     |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<--------------------------------------------------------------------+
|                            message_2                                |
|                                                                     |
|          S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)           |
+-------------------------------------------------------------------->|
|                            message_3                                |
```

EDHOC with Asymmetric Keys

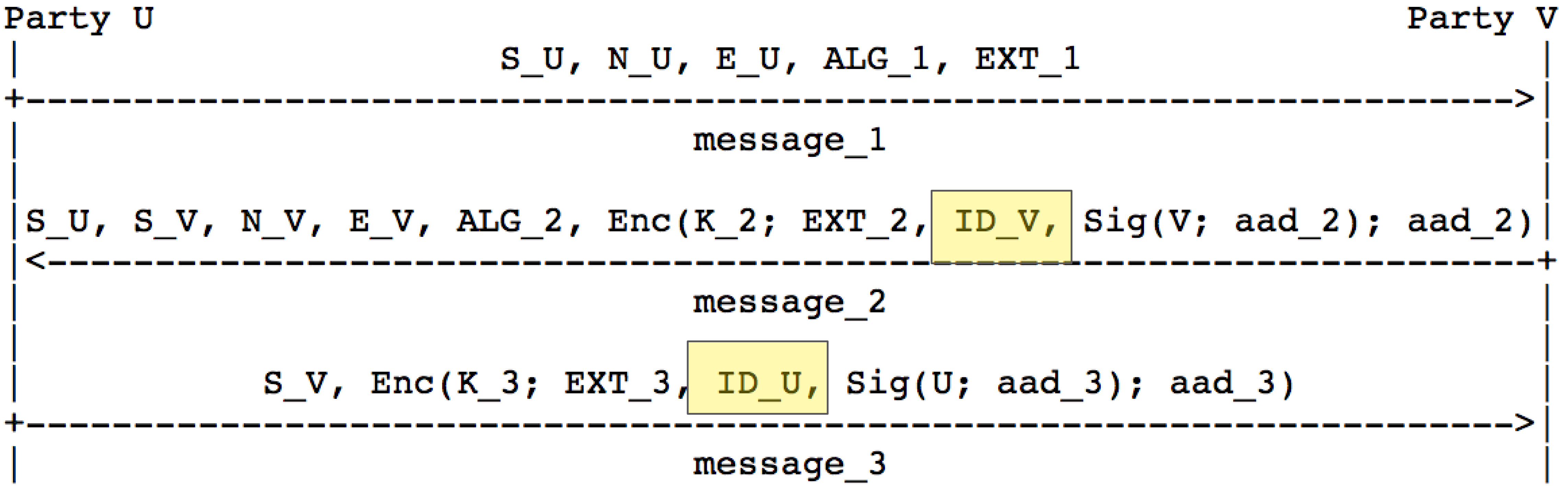
- Two COSE Encrypt0 object protected with two different keys K_2 and K_3



```
Party U                                                              Party V
|                   S_U, N_U, E_U, ALG_1, EXT_1                            |
+------------------------------------------------------------------------->|
|                             message_1                                    |
|                                                                          |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<-------------------------------------------------------------------------+
|                             message_2                                    |
|                                                                          |
|           S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)               |
+------------------------------------------------------------------------->|
|                             message_3                                    |
```

EDHOC with Asymmetric Keys

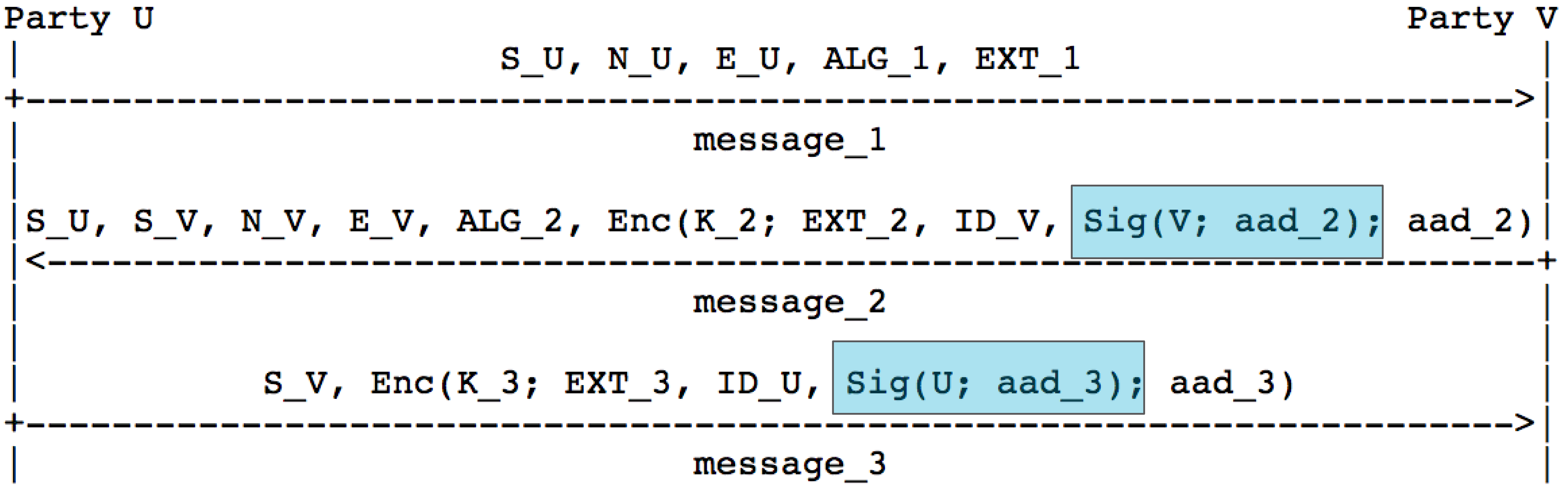
- Certificates or RPK identifiers are sent in ID_V and ID_U.
- Makes use of draft-schaad-cose-x509



```
Party U                                                         Party V
|                      S_U, N_U, E_U, ALG_1, EXT_1                     |
+--------------------------------------------------------------------->|
|                               message_1                              |
|                                                                      |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<---------------------------------------------------------------------+
|                               message_2                              |
|                                                                      |
|           S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)           |
+--------------------------------------------------------------------->|
|                               message_3                              |
```

EDHOC with Asymmetric Keys

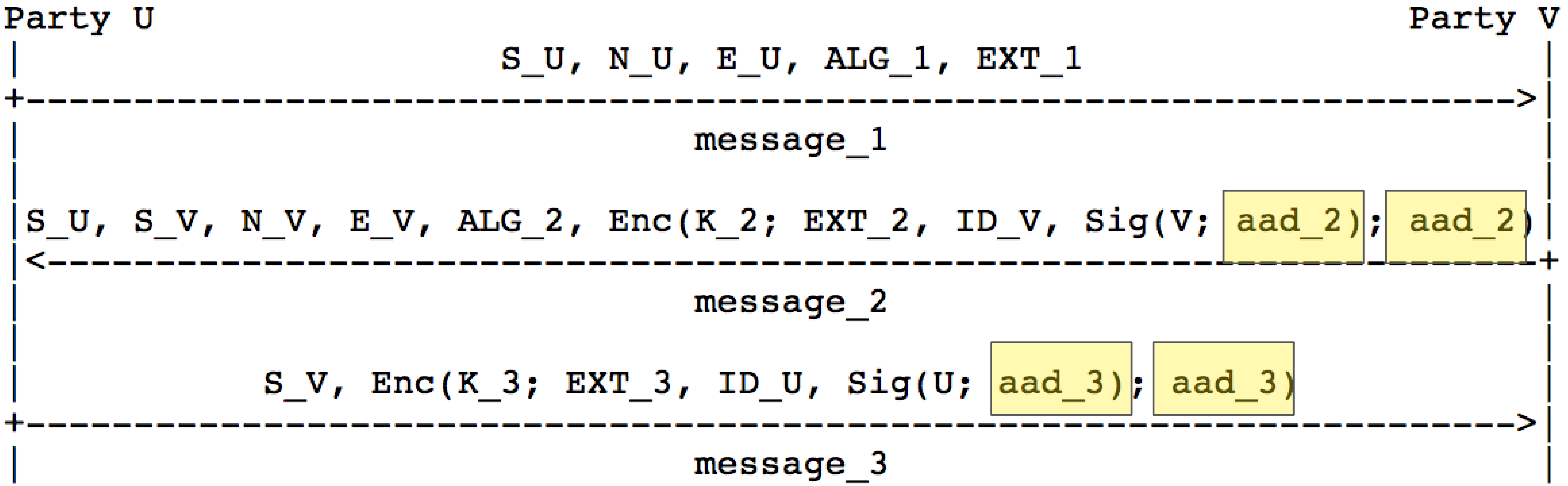
- Two COSE Sign1 object signed by Party V and Party U.
- Party U and Party V may use different signature algorithms.



```
Party U                                                          Party V
|                      S_U, N_U, E_U, ALG_1, EXT_1                      |
+--------------------------------------------------------------------->|
|                               message_1                               |
|                                                                       |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<---------------------------------------------------------------------+
|                               message_2                               |
|                                                                       |
|           S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)            |
+--------------------------------------------------------------------->|
|                               message_3                               |
```

EDHOC with Asymmetric Keys

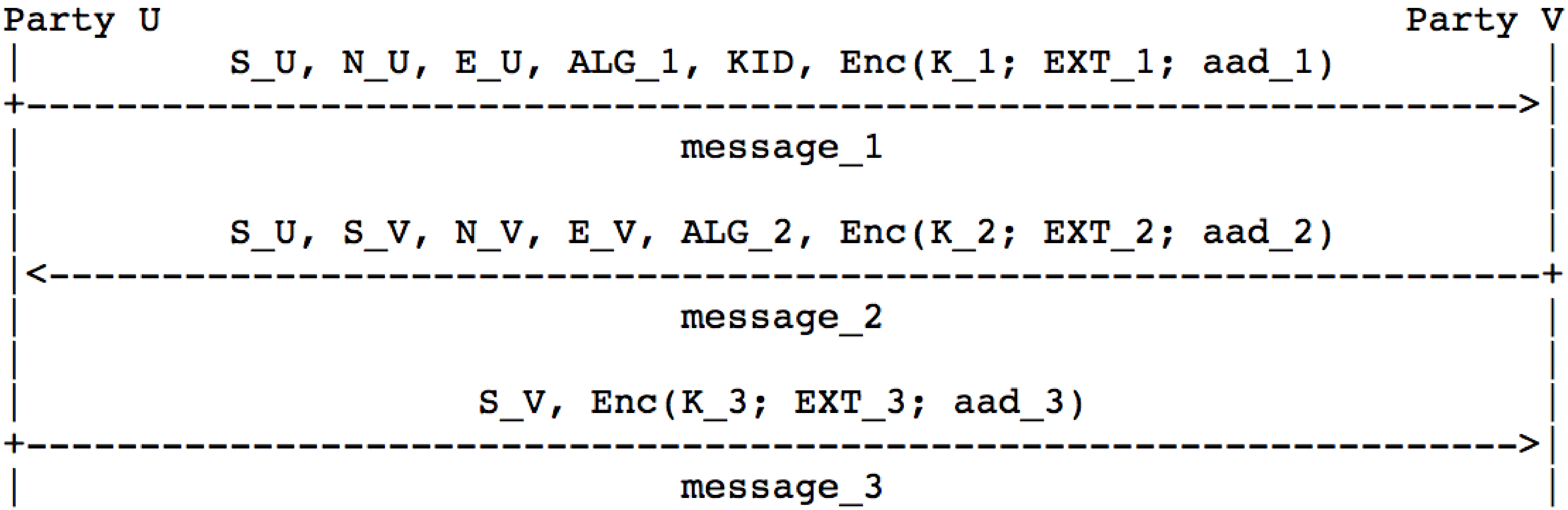
- Same AAD structure in MAC and Signature. Contains all previous messages.
- Previous messages are hashed to save memory.



```
Party U                                                        Party V
|                 S_U, N_U, E_U, ALG_1, EXT_1                        |
+------------------------------------------------------------------>|
|                          message_1                                 |
|                                                                    |
|S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2, ID_V, Sig(V; aad_2); aad_2)|
|<------------------------------------------------------------------+
|                          message_2                                 |
|                                                                    |
|           S_V, Enc(K_3; EXT_3, ID_U, Sig(U; aad_3); aad_3)         |
+------------------------------------------------------------------>|
|                          message_3                                 |
```

EDHOC with symmetric Keys

- Similar to the asymmetric case but without COSE_Sign0 with an COSE_Encrypt0 in message_1 to encrypt EXT_1 and get PSK proof-of-possession already in message_1 (may be used for DoS protection).
- Keys K_2 and K_3 derived from both PSK and the Diffie-Hellman secret.

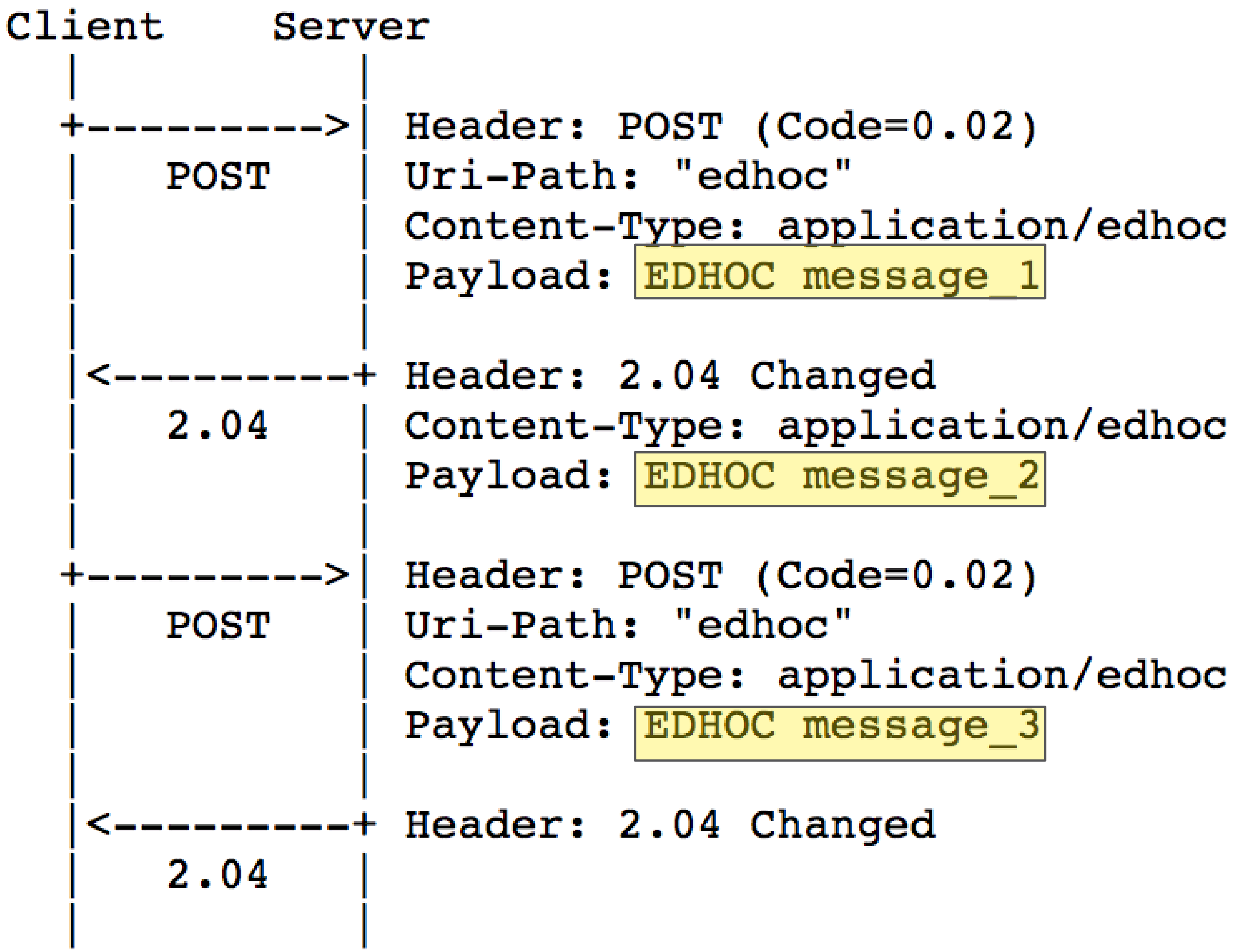


```
Party U                                                       Party V
|         S_U, N_U, E_U, ALG_1, KID, Enc(K_1; EXT_1; aad_1)         |
+------------------------------------------------------------------>|
|                             message_1                             |
|                                                                   |
|         S_U, S_V, N_V, E_V, ALG_2, Enc(K_2; EXT_2; aad_2)         |
|<------------------------------------------------------------------+
|                             message_2                             |
|                                                                   |
|                    S_V, Enc(K_3; EXT_3; aad_3)                    |
+------------------------------------------------------------------>|
|                             message_3                             |
```

EXAMPLE

- Sending EDHOC embedded in OSCOAP has been removed. EDHOC is now sent as payload.

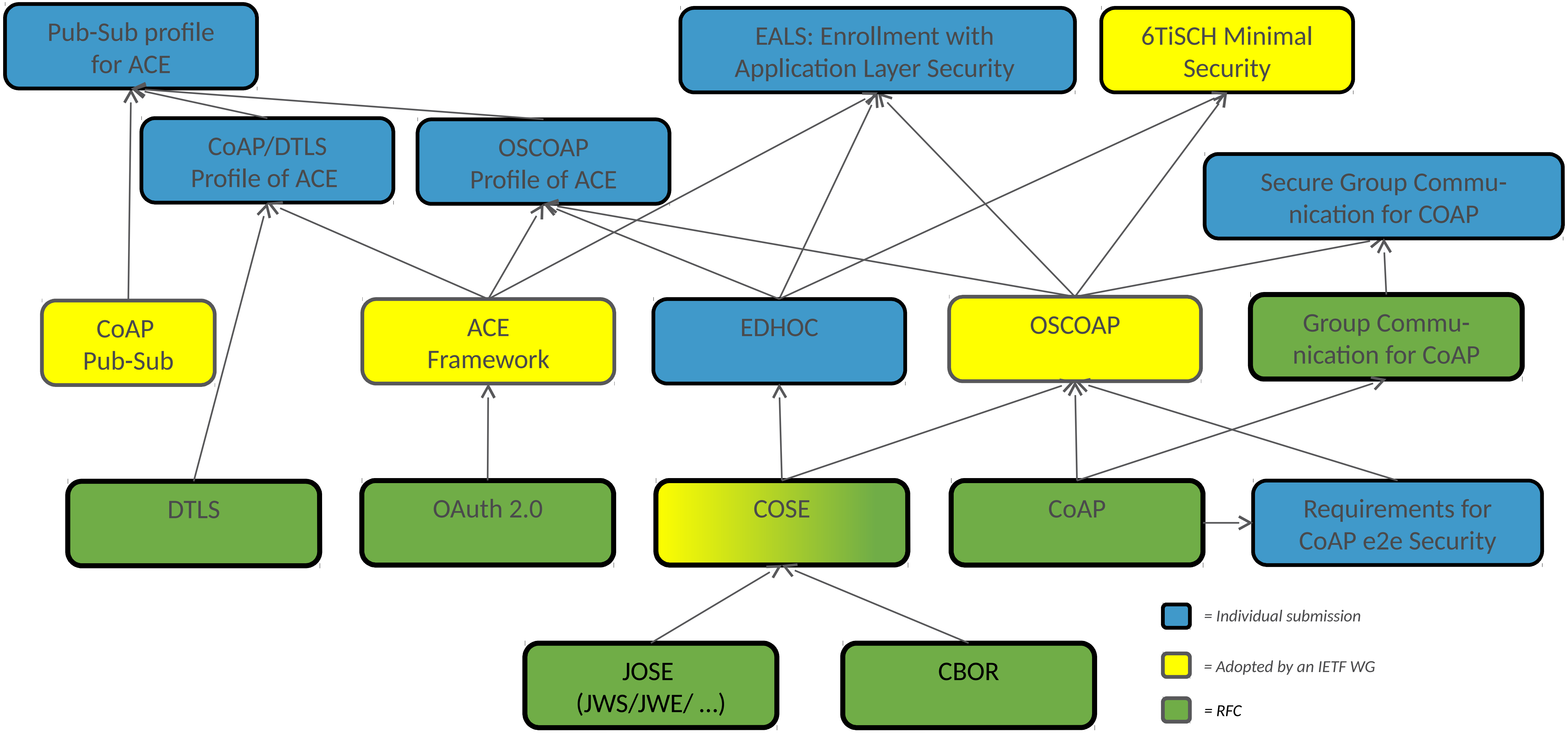
- OSCOAP Master Secret, Master Salt, and identities can be obtained from EDHOC.



```
Client    Server
  |          |
  +--------->| Header: POST (Code=0.02)
  |   POST   | Uri-Path: "edhoc"
  |          | Content-Type: application/edhoc
  |          | Payload: EDHOC message_1
  |          |
  |<---------+ Header: 2.04 Changed
  |   2.04   | Content-Type: application/edhoc
  |          | Payload: EDHOC message_2
  |          |
  +--------->| Header: POST (Code=0.02)
  |   POST   | Uri-Path: "edhoc"
  |          | Content-Type: application/edhoc
  |          | Payload: EDHOC message_3
  |          |
  |<---------+ Header: 2.04 Changed
  |   2.04   |
  |          |
```

Figure 5: Transferring EDHOC in CoAP

Related Work



🟦 = Individual submission

🟨 = Adopted by an IETF WG

🟩 = RFC

NEXT STEPS

- Already one implementation of -05 using asymmetric keys by Jim Schaad. Another implementation in progress by a master thesis student. Interop planned before the summer.

- Some specific proposed changes under consideration. Nothing major.

- Test vectors, error messages.

ERICSSON