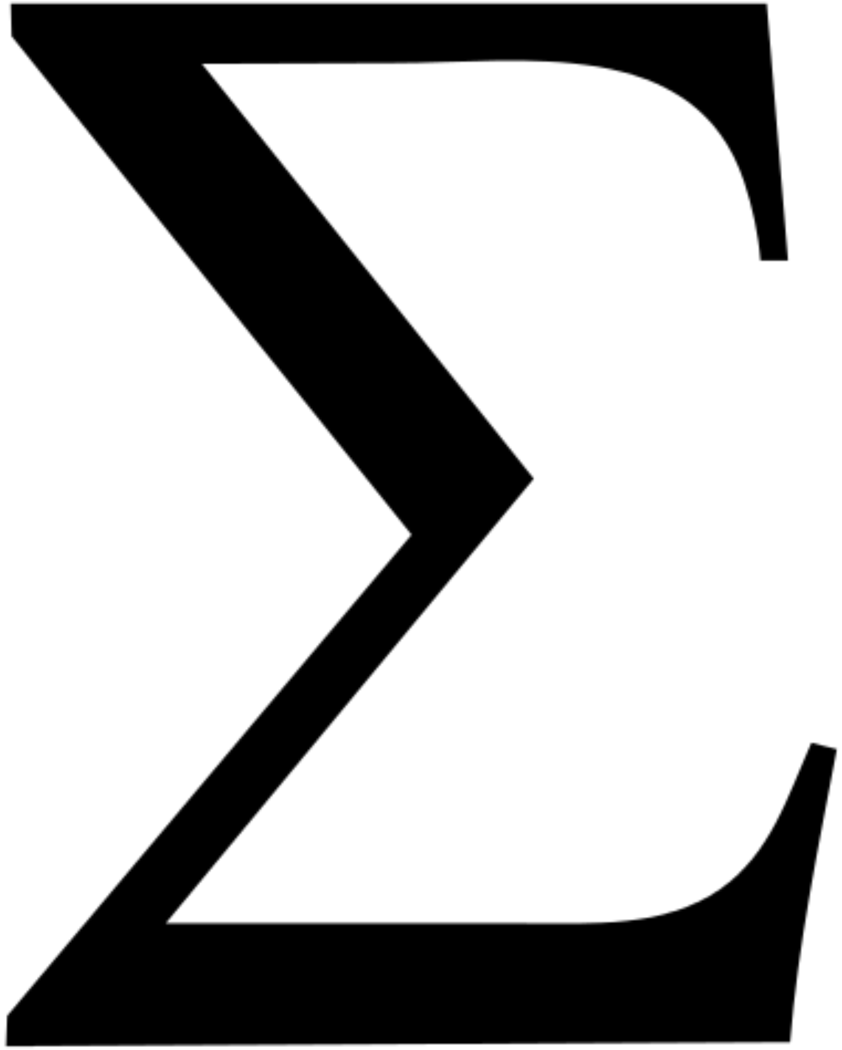
ERICSSON

# EPHEMERAL DIFFIE-HELLMAN OVER COSE (EDHOC)

DRAFT-SELANDER-ACE-COSE-ECDHE-04
SELANDER, MATTSSON, PALOMBINI
IETF97 ACE, NOV 17 2016

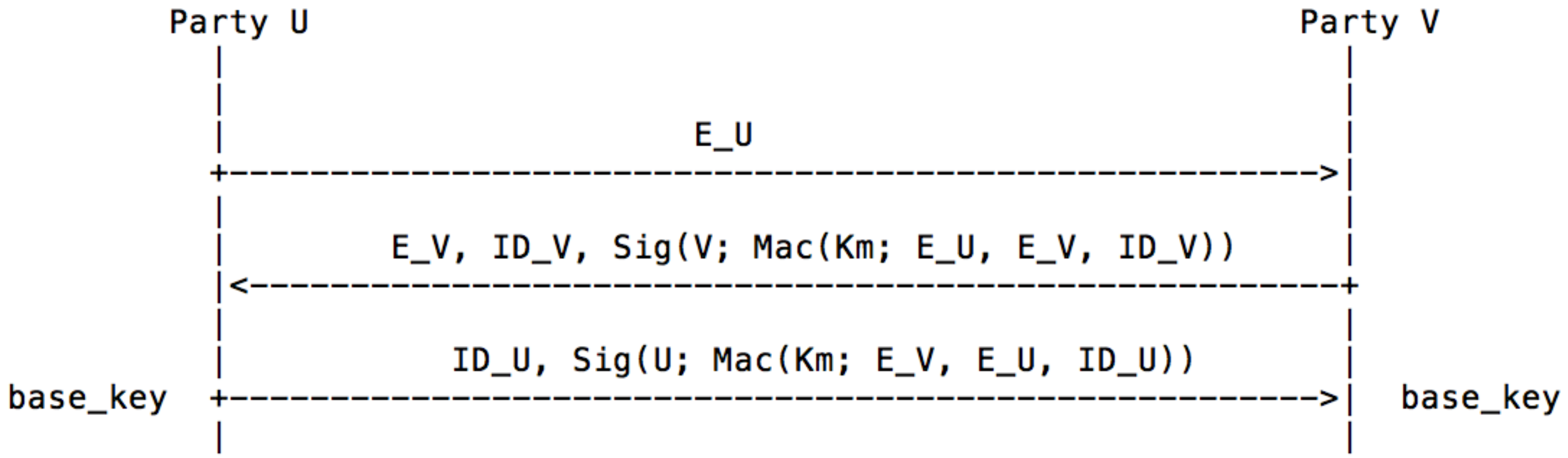
# NEW VERSION -04

- Built on the SIGMA family of key exchange protocols
  - Aligning with state-of-the-art security protocols
  - Has better security properties.
  - IKEv2 and TLS 1.3 are also based on SIGMA.
- 3 messages instead of 2
  - But no extra round-trips. Application data can be sent together with message 3 (similar to TLS 1.3)
- Still implemented using CBOR and COSE
- Still Diffie-Hellman (DH) key exchange protocol with ephemeral keys



# THE BASIC SIGMA PROTOCOL

- 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.

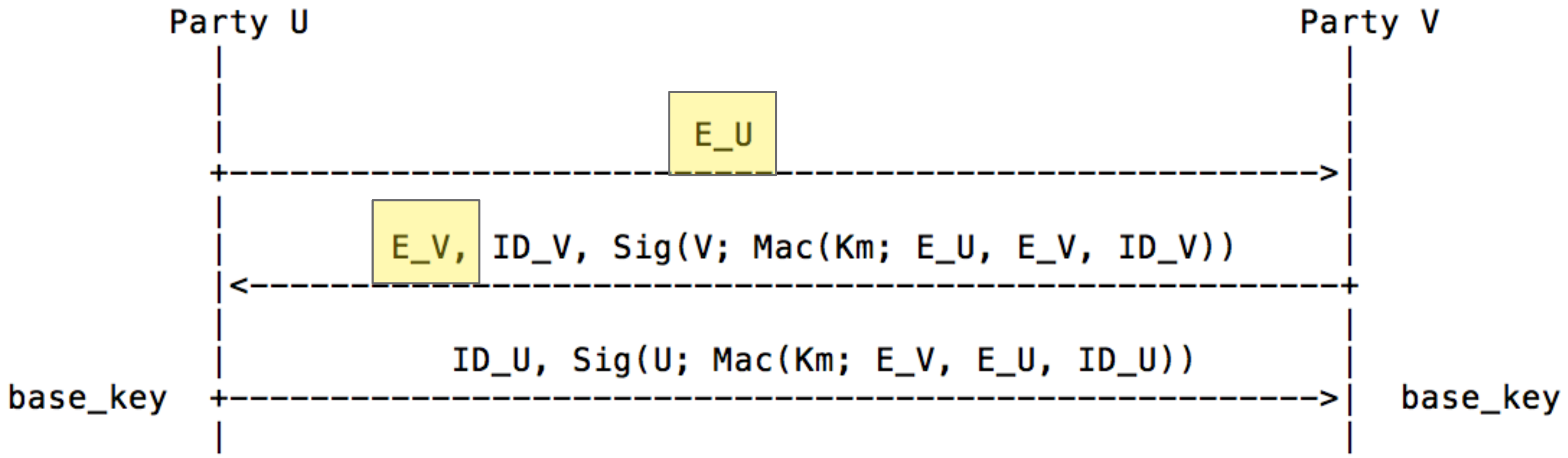


```
          Party U                                   Party V
             |                                         |
             |                                         |
             |                  E_U                    |
             +---------------------------------------->|
             |                                         |
             |    E_V, ID_V, Sig(V; Mac(Km; E_U, E_V, ID_V))
             |<----------------------------------------+
             |                                         |
             |      ID_U, Sig(U; Mac(Km; E_V, E_U, ID_U))
   base_key  +---------------------------------------->|  base_key
             |                                         |
```

Figure 1: The basic SIGMA protocol

# THE BASIC SIGMA PROTOCOL

- 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.



```
          Party U                                  Party V
             |                                        |
             |                                        |
             |                 E_U                    |
             +--------------------------------------->|
             |                                        |
             |  E_V, ID_V, Sig(V; Mac(Km; E_U, E_V, ID_V))  |
             |<---------------------------------------+
             |                                        |
             |   ID_U, Sig(U; Mac(Km; E_V, E_U, ID_U))      |
 base_key    +--------------------------------------->|  base_key
             |                                        |
```

Figure 1: The basic SIGMA protocol

# THE BASIC SIGMA PROTOCOL

- 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.

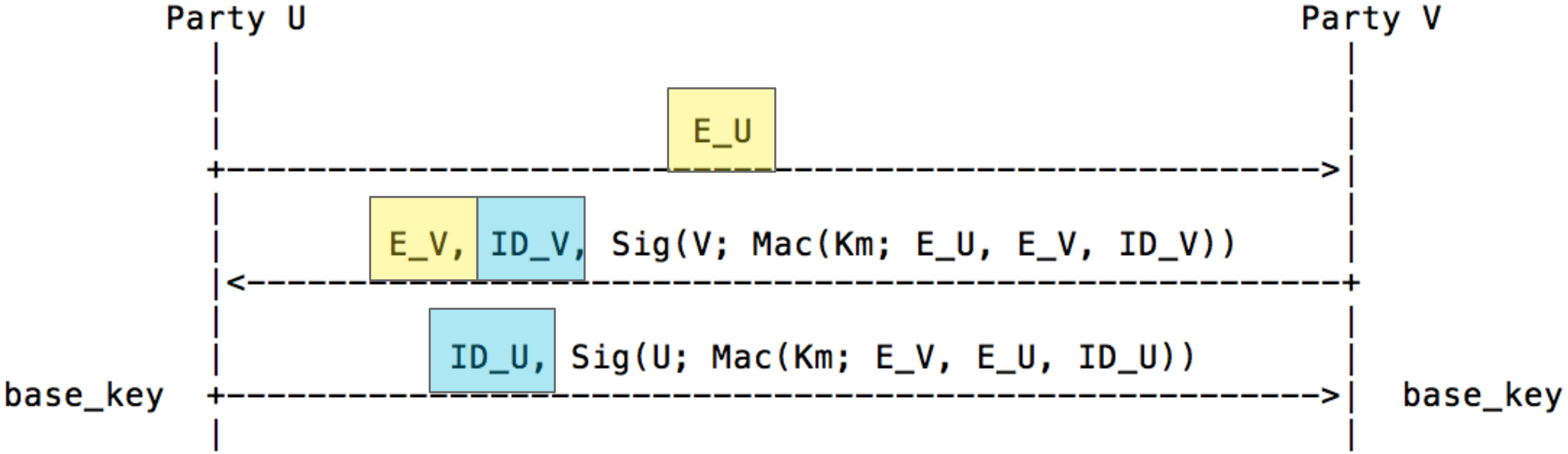

Party U
Party V
E_U
E_V, ID_V, Sig(V; Mac(Km; E_U, E_V, ID_V))
ID_U, Sig(U; Mac(Km; E_V, E_U, ID_U))
base_key
base_key

Figure 1: The basic SIGMA protocol

# THE BASIC SIGMA PROTOCOL

- 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.

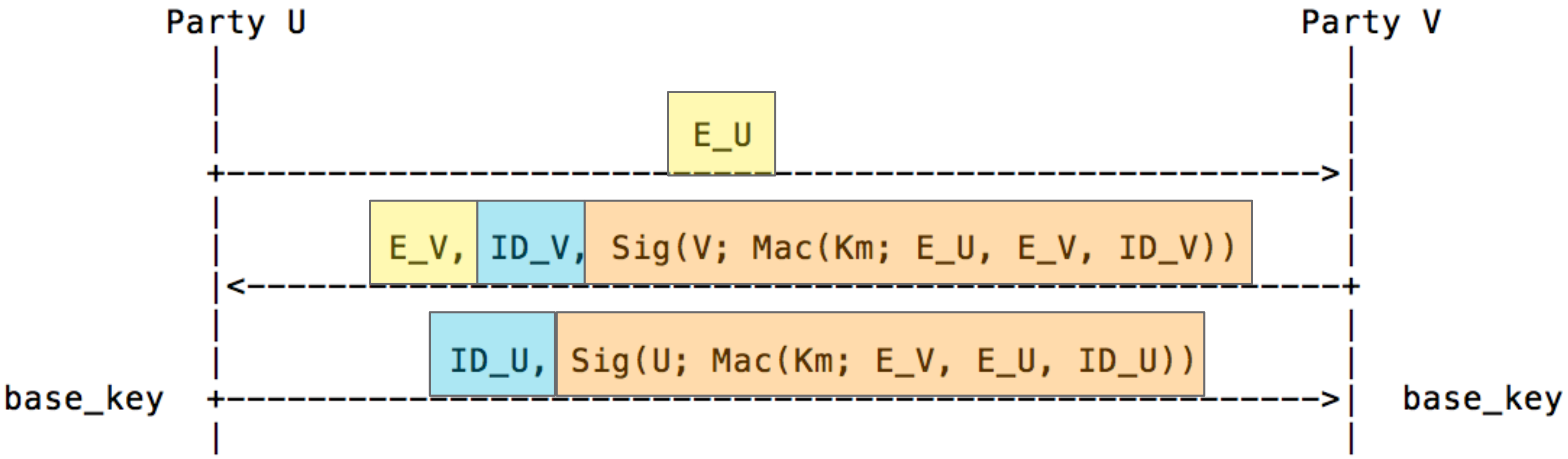

Party U
Party V
E_U
E_V,
ID_V,
Sig(V; Mac(Km; E_U, E_V, ID_V))
ID_U,
Sig(U; Mac(Km; E_V, E_U, ID_U))
base_key
base_key

Figure 1: The basic SIGMA protocol

# THE BASIC SIGMA PROTOCOL

- 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.

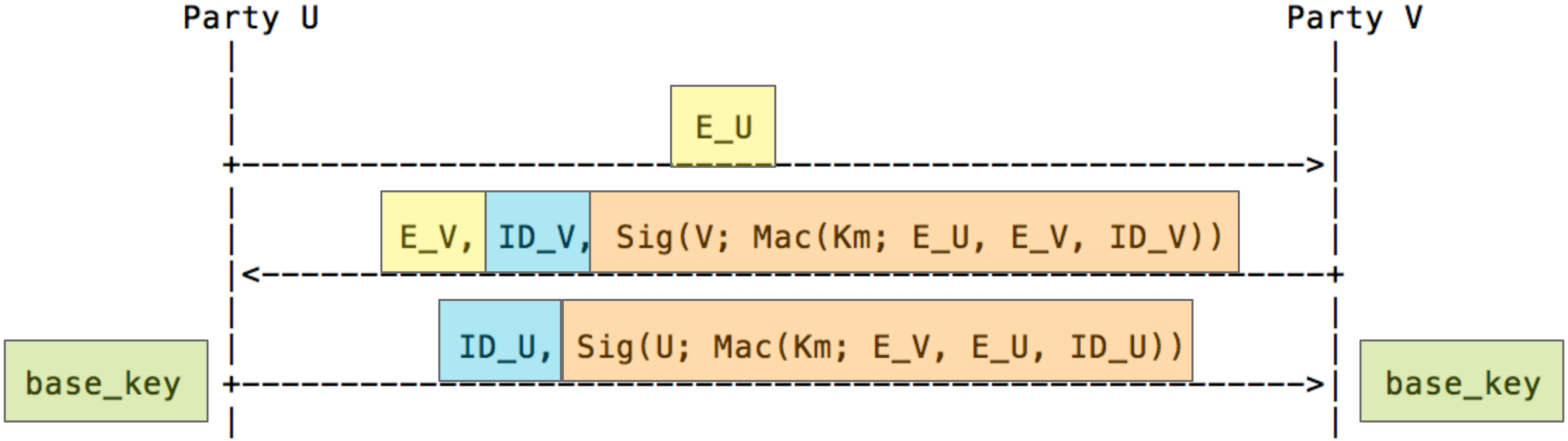
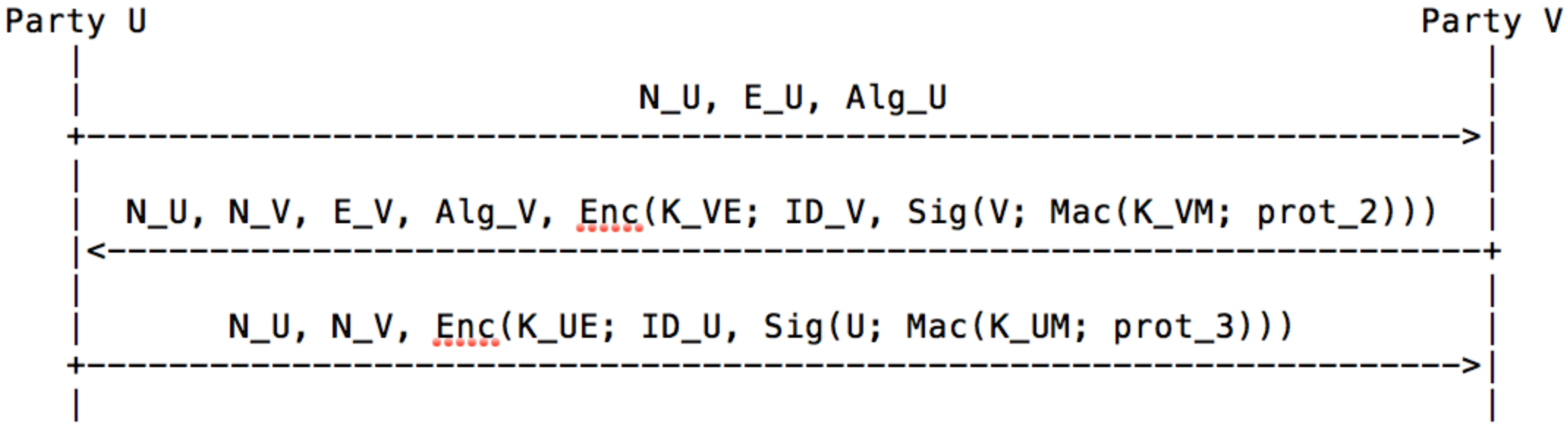

Party U
Party V
E_U
E_V,
ID_V,
Sig(V; Mac(Km; E_U, E_V, ID_V))
ID_U,
Sig(U; Mac(Km; E_V, E_U, ID_U))
base_key
base_key

Figure 1: The basic SIGMA protocol

# EDHOC

- Based on the SIGMA-I protocol that includes encryption. Adds nonces, explicit key derivation, and algorithm negotiation. Realized using CBOR and COSE.
- The DH key exchange messages may be authenticated using either pre-shared keys (PSK), raw public keys (RPK) or X.509 certificates (Cert).

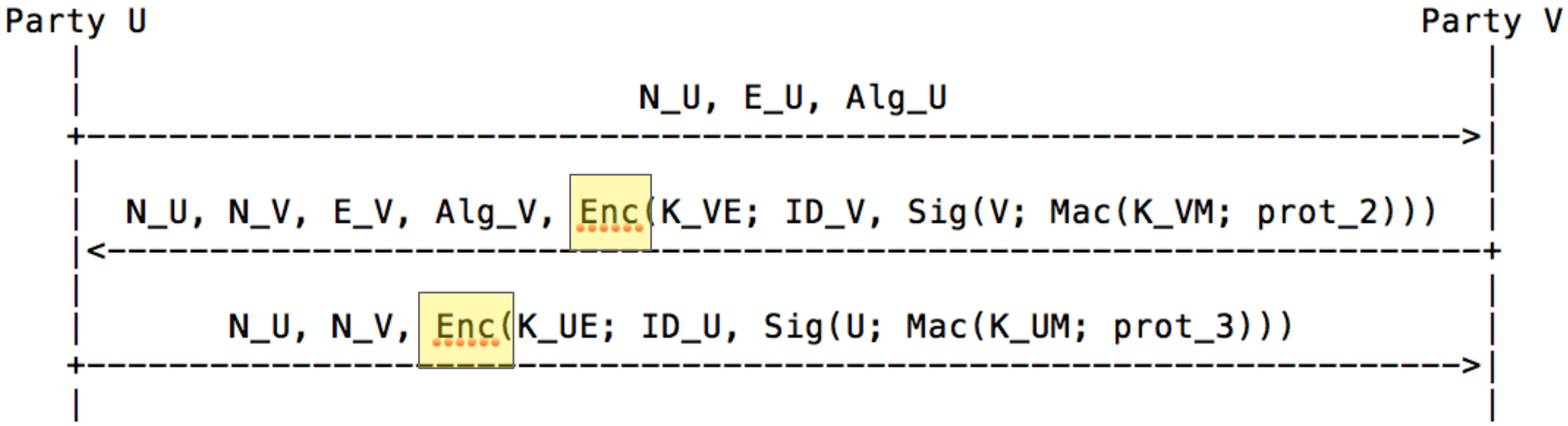


```
Party U                                                     Party V
   |                                                           |
   |                    N_U, E_U, Alg_U                        |
   +---------------------------------------------------------->|
   |                                                           |
   |  N_U, N_V, E_V, Alg_V, Enc(K_VE; ID_V, Sig(V; Mac(K_VM; prot_2)))  |
   |<----------------------------------------------------------+
   |                                                           |
   |      N_U, N_V, Enc(K_UE; ID_U, Sig(U; Mac(K_UM; prot_3)))       |
   +---------------------------------------------------------->|
   |                                                           |
```

EDHOC with asymmetric keys.

# EDHOC

- Based on the SIGMA-I protocol that includes encryption. Adds nonces, explicit key derivation, and algorithm negotiation. Realized using CBOR and COSE.
- The DH key exchange messages may be authenticated using either pre-shared keys (PSK), raw public keys (RPK) or X.509 certificates (Cert).

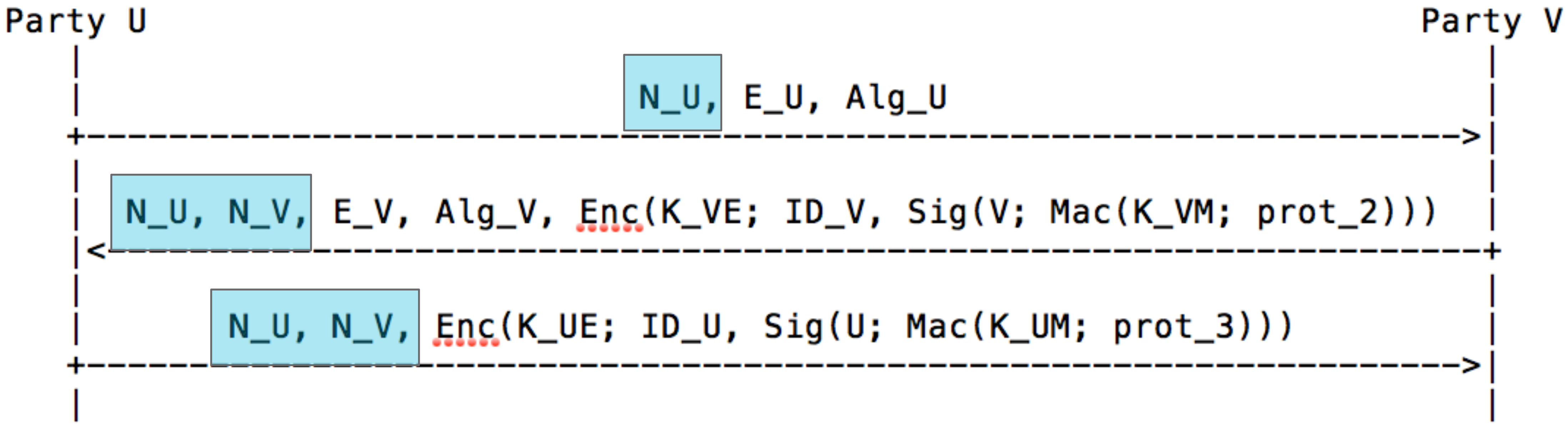


```
Party U                                                    Party V
   |                                                          |
   |                     N_U, E_U, Alg_U                      |
   +--------------------------------------------------------->|
   |                                                          |
   |  N_U, N_V, E_V, Alg_V, Enc(K_VE; ID_V, Sig(V; Mac(K_VM; prot_2)))  |
   |<---------------------------------------------------------+
   |                                                          |
   |      N_U, N_V, Enc(K_UE; ID_U, Sig(U; Mac(K_UM; prot_3)))      |
   +--------------------------------------------------------->|
   |                                                          |
```

EDHOC with asymmetric keys.

# EDHOC

- Based on the SIGMA-I protocol that includes encryption. Adds nonces, explicit key derivation, and algorithm negotiation. Realized using CBOR and COSE.
- The DH key exchange messages may be authenticated using either pre-shared keys (PSK), raw public keys (RPK) or X.509 certificates (Cert).

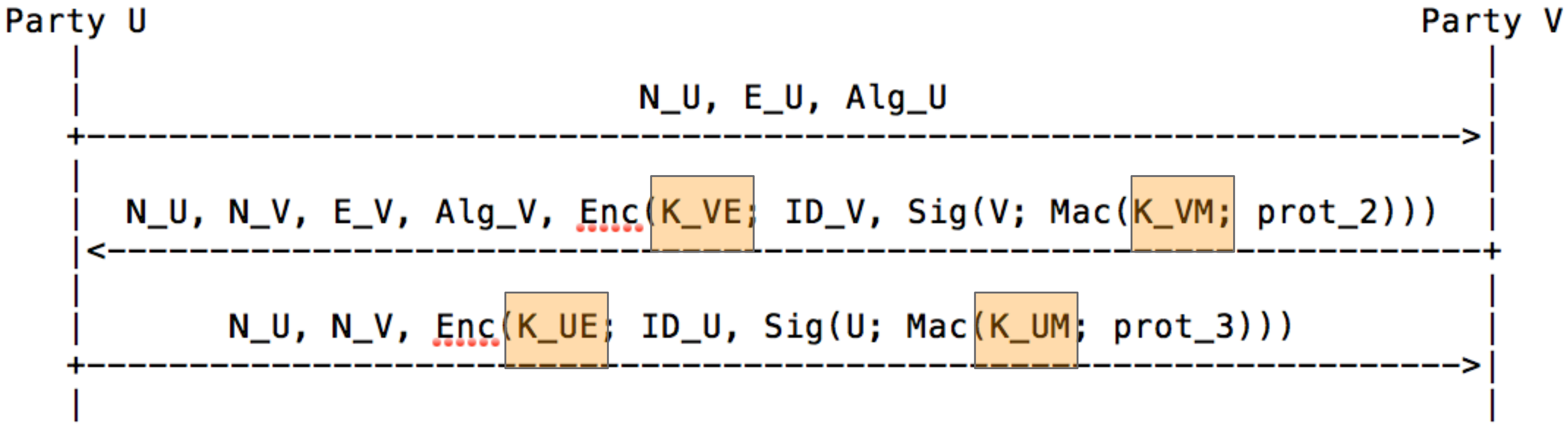


```
Party U                                                            Party V
   |                          N_U, E_U, Alg_U                         |
   |                                                                  |
   +---------------------------------------------------------------->|
   |                                                                  |
   | N_U, N_V, E_V, Alg_V, Enc(K_VE; ID_V, Sig(V; Mac(K_VM; prot_2)))  |
   |<-----------------------------------------------------------------+
   |                                                                  |
   |      N_U, N_V, Enc(K_UE; ID_U, Sig(U; Mac(K_UM; prot_3)))         |
   +---------------------------------------------------------------->|
   |                                                                  |
```

EDHOC with asymmetric keys.

# EDHOC

- Based on the SIGMA-I protocol that includes encryption. Adds nonces, explicit key derivation, and algorithm negotiation. Realized using CBOR and COSE.
- The DH key exchange messages may be authenticated using either pre-shared keys (PSK), raw public keys (RPK) or X.509 certificates (Cert).

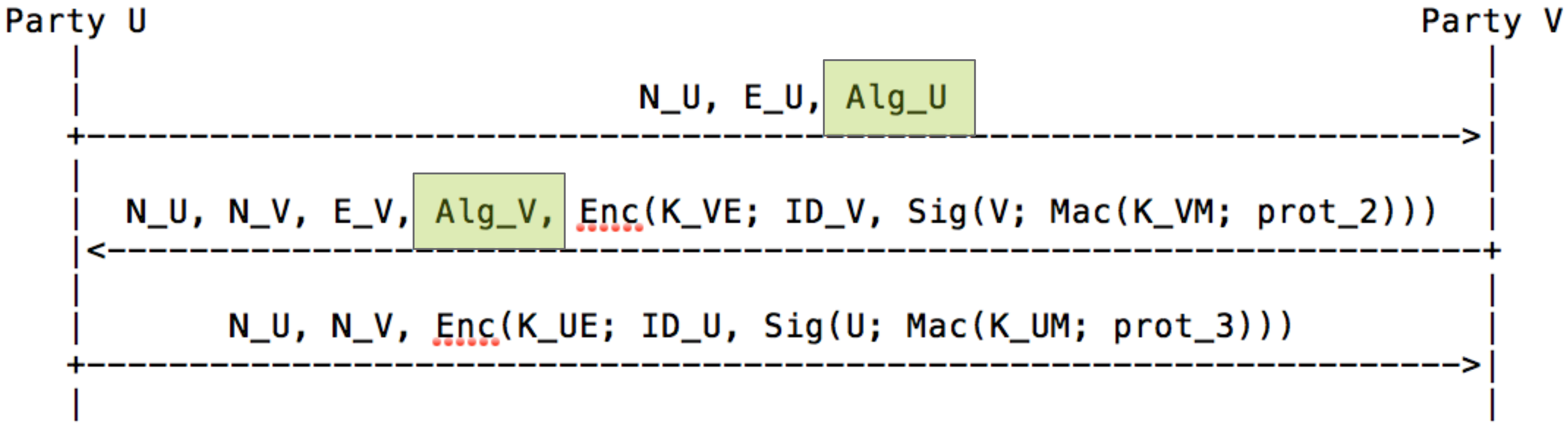


```
Party U                                                        Party V
   |                                                              |
   |                      N_U, E_U, Alg_U                          |
   +------------------------------------------------------------->|
   |                                                              |
   |  N_U, N_V, E_V, Alg_V, Enc(K_VE; ID_V, Sig(V; Mac(K_VM; prot_2)))  |
   |<-------------------------------------------------------------+
   |                                                              |
   |       N_U, N_V, Enc(K_UE; ID_U, Sig(U; Mac(K_UM; prot_3)))    |
   +------------------------------------------------------------->|
   |                                                              |
```

EDHOC with asymmetric keys.

# EDHOC

- Based on the SIGMA-I protocol that includes encryption. Adds nonces, explicit key derivation, and algorithm negotiation. Realized using CBOR and COSE.
- The DH key exchange messages may be authenticated using either pre-shared keys (PSK), raw public keys (RPK) or X.509 certificates (Cert).

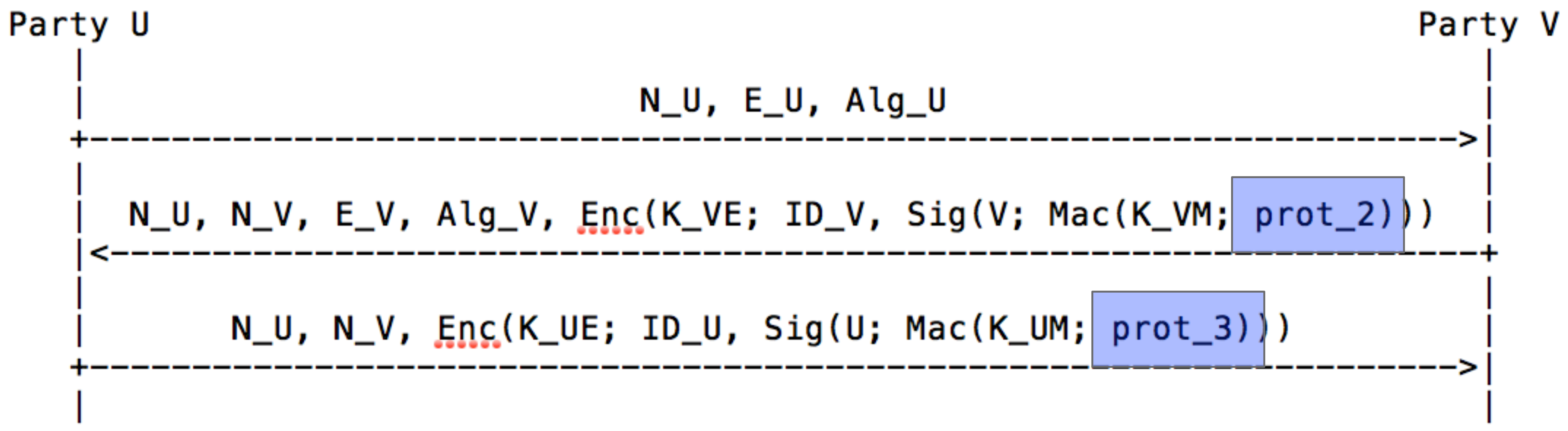


```
Party U                                                        Party V
   |                                                              |
   |                      N_U, E_U, Alg_U                         |
   +------------------------------------------------------------->|
   |                                                              |
   | N_U, N_V, E_V, Alg_V, Enc(K_VE; ID_V, Sig(V; Mac(K_VM; prot_2))) |
   |<-------------------------------------------------------------+
   |                                                              |
   |     N_U, N_V, Enc(K_UE; ID_U, Sig(U; Mac(K_UM; prot_3)))     |
   +------------------------------------------------------------->|
   |                                                              |
```

EDHOC with asymmetric keys.

# EDHOC

- Based on the SIGMA-I protocol that includes encryption. Adds nonces, explicit key derivation, and algorithm negotiation. Realized using CBOR and COSE.
- The DH key exchange messages may be authenticated using either pre-shared keys (PSK), raw public keys (RPK) or X.509 certificates (Cert).

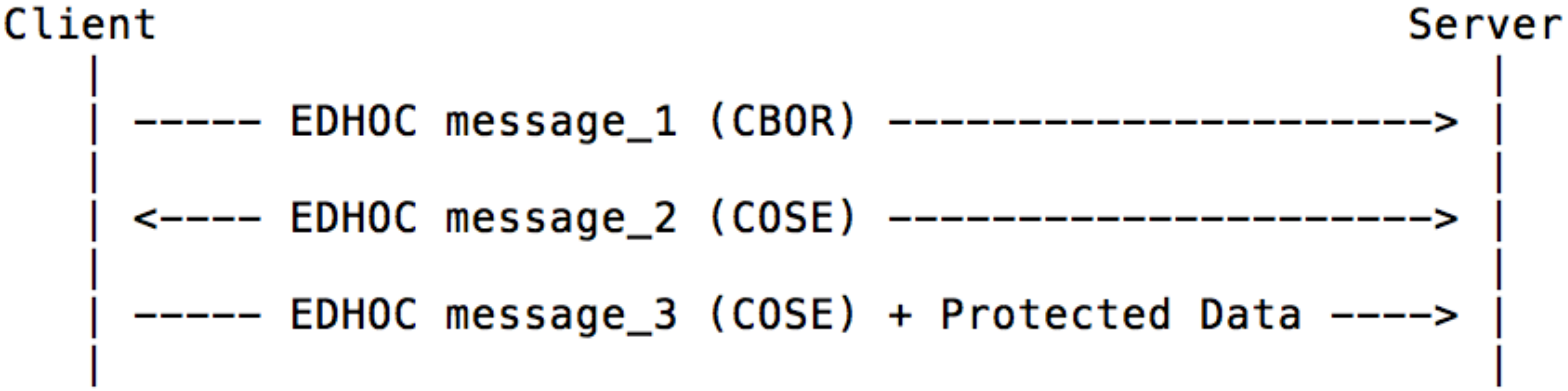


```
Party U                                                              Party V
   |                                                                    |
   |                           N_U, E_U, Alg_U                          |
   +------------------------------------------------------------------->|
   |                                                                    |
   |  N_U, N_V, E_V, Alg_V, Enc(K_VE; ID_V, Sig(V; Mac(K_VM; prot_2)))  |
   |<-------------------------------------------------------------------+
   |                                                                    |
   |       N_U, N_V, Enc(K_UE; ID_U, Sig(U; Mac(K_UM; prot_3)))         |
   +------------------------------------------------------------------->|
   |                                                                    |
```

EDHOC with asymmetric keys.

# EDHOC MESSAGE FLOW

- All EDHOC messages are encoded with CBOR
- EDHOC message_2 and message_3 uses COSE
- Protected application data can be sent together with message 3



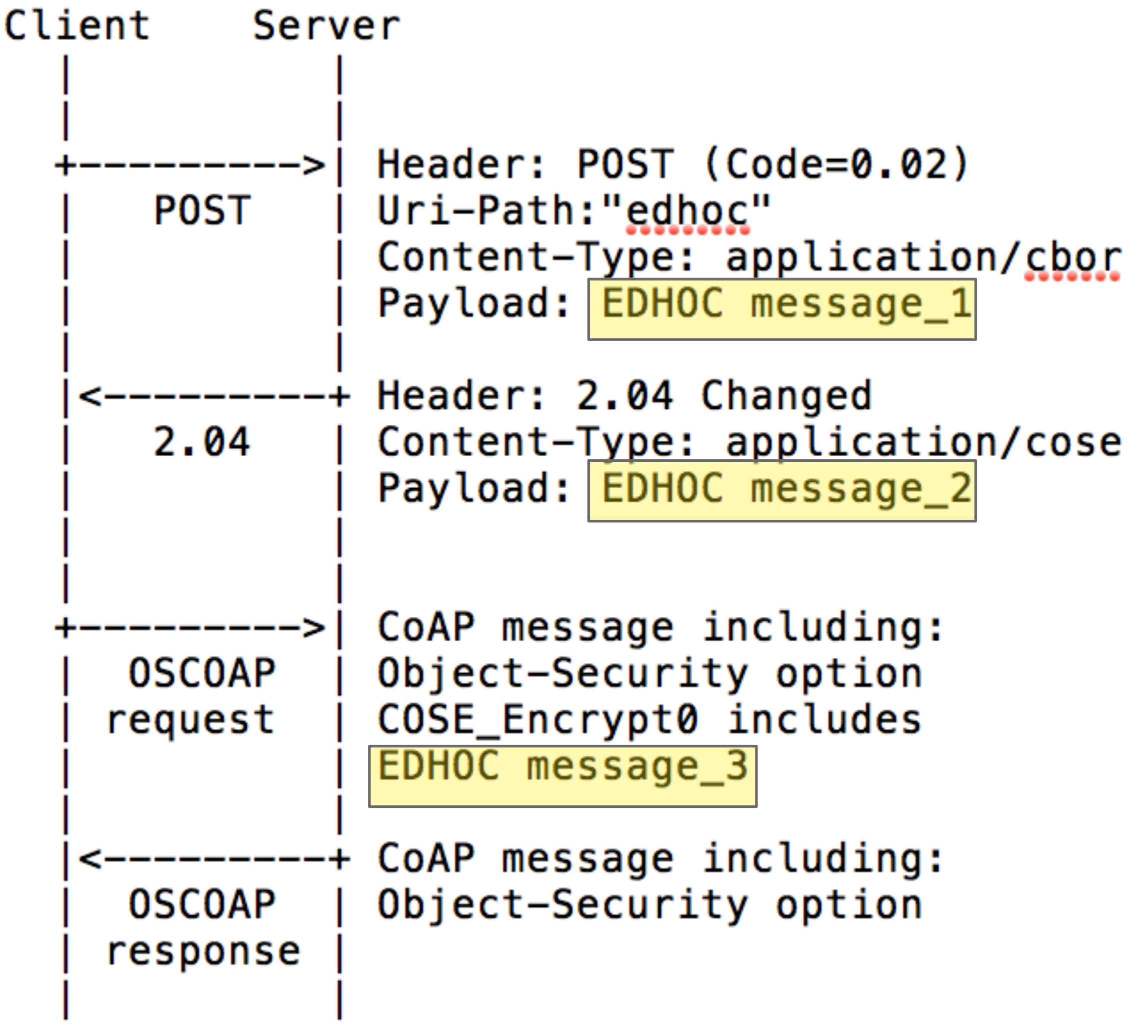
```
Client                                                Server
   |                                                     |
   | ----- EDHOC message_1 (CBOR) ---------------------> |
   |                                                     |
   | <---- EDHOC message_2 (COSE) ---------------------> |
   |                                                     |
   | ----- EDHOC message_3 (COSE) + Protected Data ----> |
   |                                                     |
```

# EXAMPLE

- Can e.g. be implemented as CoAP message exchanges with the CoAP client as party U and the CoAP server as party V.

- EDHOC and OSCOAP can be run in sequence embedded in a 2-round trip message exchange, where the base_key used in OSCOAP is obtained from EDHOC.

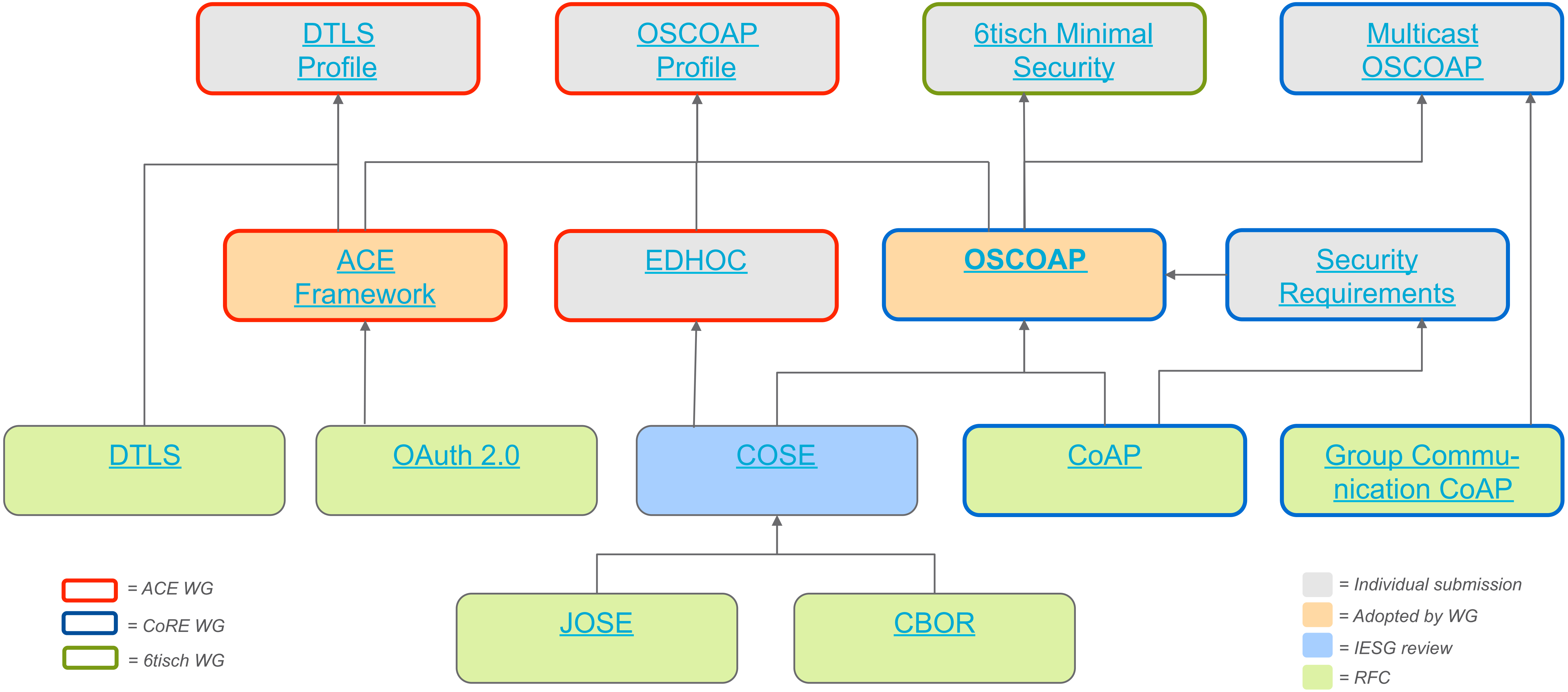
This is how EDHOC is use in the OSCOAP profile of ACE
draft-seitz-ace-oscoap-profile



```
Client      Server
   |           |
   |           |
   +---------->| Header: POST (Code=0.02)
   |   POST    | Uri-Path:"edhoc"
   |           | Content-Type: application/cbor
   |           | Payload: EDHOC message_1
   |           |
   |<----------+ Header: 2.04 Changed
   |   2.04    | Content-Type: application/cose
   |           | Payload: EDHOC message_2
   |           |
   |           |
   +---------->| CoAP message including:
   |  OSCOAP   | Object-Security option
   |  request  | COSE_Encrypt0 includes
   |           | EDHOC message_3
   |           |
   |<----------+ CoAP message including:
   |  OSCOAP   | Object-Security option
   | response  |
   |           |
```

Detail of EDHOC and OSCOAP

# RELATED WORK



# NEXT STEPS

- Two implementations underway
  - SICS
  - Jim Schaad

- Minor updates based on review comments

- Ask for CFRG review

ERICSSON