

## SIG-SIG-KEM (simplified)



Choose METHHOD and SUITES

 $Msg_1 = (METHOD, SUITES, G_X, [EAD_1])$ 

- If agree on METHOD and SUITES. (ct\_kem, k\_kem) = KEM.Encap(G\_X)
   G Y = ct kem, G XY = k kem
- TH\_2 = H(G\_Y, H(Msg\_1))
- PRK\_2e = HKDF\_Extract(TH\_2, G\_XY), PRK\_3e2m = PRK\_2e
- MAC\_2 = mac(PRK\_3e2m, 2, C\_R || ID\_CRED\_R || TH\_2 || CRED\_R || EAD\_2)
- SIG\_2 = Sign(sk\_R, (ID\_CRED\_R, (TH\_2 || CRED\_R || [EAD\_2]), MAC\_2))
- Ptxt\_2 = (C\_R, ID\_CRED\_R, SIG\_2, [EAD\_2])

 $Msg_2 = (G_Y, Ptxt_2 \oplus KDF(PRK_2e, 0, TH_2, Ptxt_2_length))$ 

- Verify Sig\_2 (verify Responder), if it fails then abort. G\_XY = KEM.Decap(sk\_kem, G\_Y)
- TH\_3 = H(TH\_2, , Ptxt\_2, ), PRK\_4e3m = PRK\_3e2m
- MAC\_3 = mac(PRK\_4e3m, 6, ID\_CRED\_I || TH\_3 || CRED\_I || [EAD\_3])
- SIG\_3 = Sign(sk\_I, (ID\_CRED\_I, (TH\_3 || CRED\_I || [EAD\_3]), MAC\_3))
- Ptxt\_3 = (ID\_CRED\_I, SIG\_3, [EAD\_3])
- K\_3 = KDF(PRK\_4e3m, 3, TH\_3, key\_length)
- TH\_4 = H(TH\_3, Ptxt\_3, CRED\_I), PRK\_out = KDF(PRK\_4e3m, 7, TH\_4, hash\_lenght)

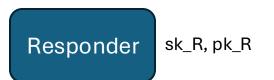
 $Msg_3 = AEAD.Enc(K_3, \{IV_3\}, Ptxt_3, \{AD_3\})$ 

- Decrypt Msg\_3 then Verify Sig\_3 (verify Initiator), if it fails then abort
- TH\_4 = H(TH\_3, Ptxt\_3, CRED\_I), PRK\_out = KDF(PRK\_4e3m, 7, TH\_4, hash\_lenght)

Shared Key = KDF(PRK\_out, 10, " ", hash\_length)



## KEM-KEM (simplified)



(pk\_kem, sk\_kem) = KEM.KeyGen(), G\_X = pk\_kem
(ct\_auth\_R, K\_auth\_R) = KEM.Encap(pk\_R)
Enc auth R = AEAD.Enc(K auth R, ID CRED I)

Msg\_1 = (METHOD, SUITES, G\_X, [EAD\_1]) + ct\_auth\_R + Enc\_auth\_R

Choose METHHOD and SUITES

Should have IV, Nonce to use with AEAD.Enc (YES):  $TH_1 = H(Msg_1)$ 

 $AAD = TH_1$ 

 $IV_1 = KDF(K_auth_R, -1, TH_1, IV_length)$ 

If agree on METHOD and SUITES.

K\_auth\_R = KEM.Decap(sk\_R, ct\_auth\_R), ID\_CRED\_I = AEAD.Dec(K\_auth\_R, Enc\_auth\_R)
Verify CRED\_I (verify Initiator), (ct\_auth\_I, K\_auth\_I) = KEM.Encap(pk\_I)
(ct\_kem, k\_kem) = KEM.Encap(G\_X), G\_Y = ct\_kem, G\_XY = k\_kem

- TH 2 = H(G Y, H(Msg 1))

- PRK 2e = HKDF Extract(TH 2, G XY), PRK 3e2m = HKDF Extract(SALT 3e2m, K auth R)
- MAC\_2 = mac(PRK\_3e2m, 2, C\_R || ID\_CRED\_R || TH\_2 || CRED\_R || EAD\_2)
- Ptxt 2=(C R, ID CRED R, MAC 2, [EAD 2])

 $Msg_2 = (G_Y, Ptxt_2 \oplus KDF(PRK_2e, 0, TH_2, Ptxt_2_length)) +$   $ct_auth_1$ 

- Verify MAC\_2 and CRED\_R if it fails then abort. G\_XY = KEM.Decap(sk\_kem, G\_Y)
   K auth\_I = KEM.Decap(sk\_I, ct\_auth\_I)
- TH\_3 = H(TH\_2, , Ptxt\_2, ), PRK\_4e3m = HKDF\_Extract(SALT\_4e3m, K\_auth\_I)
- MAC\_3 = mac(PRK\_4e3m, 6, ID\_CRED\_I || TH\_3 || CRED\_I || [EAD\_3])
- Ptxt\_3 = (ID\_CRED\_I, MAC\_3, [EAD\_3])
- K 3 = KDF(PRK 4e3m, 3, TH 3, key length)
- TH\_4 = H(TH\_3, Ptxt\_3, CRED\_I), PRK\_out = KDF(PRK\_4e3m, 7, TH\_4, hash\_lenght)

 $Msg_3 = AEAD.Enc(K_3, \{IV_3\}, Ptxt_3, \{AD_3\})$ 

- Decrypt **Msg\_3**, if it fails then abort
- TH\_4 = H(TH\_3, Ptxt\_3, CRED\_I), PRK\_out = KDF(PRK\_4e3m, 7, TH\_4, hash\_lenght)

Shared Key = KDF(PRK\_out, 10, " ", hash\_length)