

sk_I, pk_I

Initiator

SIG-SIG-KEM (simplified)

Responder

sk_R, pk_R

$(pk_kem, sk_kem) = KEM.KeyGen(), G_X = pk_kem$

Choose METHOD 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. $G_XY = KEM.Decap(sk_kem, G_Y)$
- $TH_3 = H(TH_2, , Ptxt_2,), PRK_4e3m = PRK_3e2m$
- $MAC_3 = mac(PRK_4e3m, 6, ID_CRED_I || TH_3 || CRED_I || [EAD_3])$
- $SIG_3 = Sign(sk_I, (ID_CRED_I, (TH_3 || CRED_I || [EAD_3]), MAC_3))$
- $Ptxt_3 = (ID_CRED_I, SIG_3, [EAD_3])$
- $K_3 = KDF(PRK_4e3m, 3, TH_3, key_length)$
- $TH_4 = H(TH_3, Ptxt_3, CRED_I), PRK_out = KDF(PRK_4e3m, 7, TH_4, hash_length)$

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

- Decrypt **Msg_3** then Verify **Sig_3 (verify Initiator)**, if it fails then abort
- $TH_4 = H(TH_3, Ptxt_3, CRED_I), PRK_out = KDF(PRK_4e3m, 7, TH_4, hash_length)$

Shared Key = KDF(PRK_out, 10, " ", hash_length)

sk_I, pk_I

Initiator

KEM-KEM
(simplified)

Responder

sk_R, pk_R

(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 METHOD 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_I

- 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_length)

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_length)

Shared Key = KDF(PRK_out, 10, " ", hash_length)