Aalto University

Network Security: Internet Key Exchange IKEv2

Tuomas Aura

CS-E4300 Network security

Aalto University

Internet Key Exchange (IKE)

- IKEv2 [RFC 7296]: authenticated key exchange for IPsec
  - Diffie-Hellman or ECDH, SIGMA (sign and MAC) protocol
  - Minimum two request-response exchanges (4 messages, 2 RTT)
  - Works over UDP port 500
- Initial exchanges create the IKE security association (IKE SA) for (re)keying and one IPsec SA pair for session data
  - CREATE_CHILD_SA exchange for later rekeying
- Endpoints: initiator I and responder R
  - Initiator can be the client or server (why?)

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, 0, SA_{i1}, g^x, N_i
2. R → I: SPI_i, SPI_r, SA_{r1}, g^y, N_r, $CERTREQ_r$
3. I → R: SPI_i, SPI_r, $E_{SK}(ID_i, CERT_i, CERTREQ_i, ID_r,$
   $Sign_i (Message1, N_r, MAC_{SK}(ID_i)), SA_{i2}, TS_i, TS_r, MAC_{SK}(\ldots))$
4. R → I: SPI_i, SPI_r, $E_{SK}(ID_r, CERT_r,$
   $Sign_R (Message2, N_i, MAC_{SK}(ID_r)), SA_{r2}, TS_i, TS_r, MAC_{SK}(\ldots))$

SPI_x = two values that together identify the protocol run and the created IKE SA

SA_{x1} = offered and chosen algorithms, DH or ECDH group

SK = h(Ni, Nr, g^{xy}) — actually, many different keys are derived from this

$Sign_x (Message_x, N_y, MAC_{SK}(ID_x))$ – SIGMA authentication

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted root CAs

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

$E_{SK}(\ldots, MAC_{SK}(\ldots))$ = Authenticated encryption for identity protection

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, 0, SA_{i1}, g^x, N_i
2. R → I: SPI_i, SPI_r, SA_{r1}, g^y, N_r, $CERTREQ_r$
3. I → R: SPI_i, SPI_r, $E_{SK}(ID_i, CERT_i, CERTREQ_i, ID_r,$
   $Sign_i (Message1, N_r, MAC_{SK}(ID_i)), SA_{i2}, TS_i, TS_r, MAC_{SK}(\ldots))$
4. R → I: SPI_i, SPI_r, $E_{SK}(ID_r, CERT_r,$
   $Sign_R (Message2, N_i, MAC_{SK}(ID_r)), SA_{r2}, TS_i, TS_r, MAC_{SK}(\ldots))$

SPI_x = two values that together identify the protocol run and the created IKE SA

SA_{x1} = offered and chosen algorithms, DH or ECDH group

SK = h(Ni, Nr, g^{xy}) — many different keys are derived from the key material

$Sign_x (Message_x, N_y, MAC_{SK}(ID_x))$ – SIGMA authentication

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted root CAs

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

$E_{SK}(\ldots, MAC_{SK}(\ldots))$ = Authenticated encryption for identity protection

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, 0, SA_{i1}, g^x, N_i
2. R → I: SPI_i, SPI_r, SA_{r1}, g^y, N_r, $CERTREQ_r$
3. I → R: SPI_i, SPI_r, $E_{SK}(ID_i, CERT_i, CERTREQ_i, ID_r,$
   $Sign_i (Message1, N_r, MAC_{SK}(ID_i)), SA_{i2}, TS_i, TS_r, MAC_{SK}(\ldots))$
4. R → I: SPI_i, SPI_r, $E_{SK}(ID_r, CERT_r,$
   $Sign_R (Message2, N_i, MAC_{SK}(ID_r)), SA_{r2}, TS_i, TS_r, MAC_{SK}(\ldots))$

SPI_x = two values that together identify the protocol run and the created IKE SA

SA_{x1} = offered and chosen algorithms, DH or ECDH group

$SK = h(Ni, Nr, g^{xy})$ — actually, many different keys are derived from this

$Sign_x (Message_x, N_y, MAC_{SK}(ID_x))$ – SIGMA authentication

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted root CAs

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

$E_{SK}(\ldots, MAC_{SK}(\ldots))$ = Authenticated encryption for identity protection

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, 0, SA_{i1}, g^x, N_i
2. R → I: SPI_i, SPI_r, SA_{r1}, g^y, N_r, $CERTREQ_r$
3. I → R: SPI_i, SPI_r, $E_{SK}(ID_i, CERT_i, CERTREQ_i, ID_r,$
   $Sign_i(Message1, N_r, MAC_{SK}(ID_i)), SA_{i2}, TS_i, TS_r, MAC_{SK}(...))$
4. R → I: SPI_i, SPI_r, $E_{SK}(ID_r, CERT_r,$
   $Sign_R(Message2, N_i, MAC_{SK}(ID_r)), SA_{r2}, TS_i, TS_r, MAC_{SK}(...))$

SPI_x = two values that together identify the protocol run and the created IKE SA

SA_{x1} = offered and chosen algorithms, DH or ECDH group

SK = $h(Ni, Nr, g^{xy})$ — actually, many different keys are derived from this

$Sign_x(Message_x, N_y, MAC_{SK}(ID_x))$ – SIGMA authentication

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted root CAs

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

$E_{SK}(..., MAC_{SK}(...))$ = Authenticated encryption for identity protection

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, 0, SA_{i1}, g^x, N_i
2. R → I: SPI_i, SPI_r, SA_{r1}, g^y, N_r, $CERTREQ_r$
3. I → R: SPI_i, SPI_r, $E_{SK}(ID_i, CERT_i, CERTREQ_i, ID_r,$ $Sign_i(Message1, N_r, MAC_{SK}(ID_i))$, SA_{i2}, TS_i, TS_r, $MAC_{SK}(...))$
4. R → I: SPI_i, SPI_r, $E_{SK}(ID_r, CERT_r,$ $Sign_R(Message2, N_i, MAC_{SK}(ID_r))$, SA_{r2}, TS_i, TS_r, $MAC_{SK}(...))$

SPI_x = two values that together identify the protocol run and the created IKE SA

SA_{x1} = offered and chosen algorithms, DH or ECDH group

$SK = h(Ni, Nr, g^{xy})$ — actually, many different keys are derived from this

$Sign_x(Message_x, N_y, MAC_{SK}(ID_x))$ – SIGMA authentication

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted root CAs

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

$E_{SK}(..., MAC_{SK}(...))$ = Authenticated encryption for identity protection

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, 0, SA_{i1}, g^x, N_i
2. R → I: SPI_i, SPI_r, SA_{r1}, g^y, N_r, $CERTREQ_r$
3. I → R: SPI_i, SPI_r, $E_{SK}(ID_i, CERT_i, CERTREQ_i, ID_r,$ $Sign_i(Message1, N_r, MAC_{SK}(ID_i))$, SA_{i2}, TS_i, TS_r, $MAC_{SK}(...))$
4. R → I: SPI_i, SPI_r, $E_{SK}(ID_r, CERT_r,$ $Sign_R(Message2, N_i, MAC_{SK}(ID_r))$, SA_{r2}, TS_i, TS_r, $MAC_{SK}(...))$

SPI_x = two values that together identify the protocol run and the created IKE SA

SA_{x1} = offered and chosen algorithms, DH or ECDH group

$SK = h(Ni, Nr, g^{xy})$ — actually, many different keys are derived from this

$Sign_x(Message_x, N_y, MAC_{SK}(ID_x))$ – SIGMA authentication

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted root CAs

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

$E_{SK}(..., MAC_{SK}(...))$ = Authenticated encryption for identity protection

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, 0, SA_{i1}, g^x, N_i
2. R → I: SPI_i, SPI_r, SA_{r1}, g^y, N_r, $CERTREQ_r$
3. I → R: SPI_i, SPI_r, $E_{SK}(ID_i, CERT_i, CERTREQ_i, ID_r,$
   $Sign_i(Message1, N_r, MAC_{SK}(ID_i)), SA_{i2}, TS_i, TS_r, MAC_{SK}(...))$
4. R → I: SPI_i, SPI_r, $E_{SK}(ID_r, CERT_r,$
   $Sign_R(Message2, N_i, MAC_{SK}(ID_r)), SA_{r2}, TS_i, TS_r, MAC_{SK}(...))$

SPI_x = two values that together identify the protocol run and the created IKE SA

SA_{x1} = offered and chosen algorithms, DH or ECDH group

$SK = h(Ni, Nr, g^{xy})$ — actually, many different keys are derived from this

$Sign_x(Message_x, N_y, MAC_{SK}(ID_x))$ – SIGMA authentication

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted root CAs

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

$E_{SK}(..., MAC_{SK}(...))$ = Authenticated encryption for identity protection

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, 0, SA_{i1}, g^x, N_i
2. R → I: SPI_i, SPI_r, SA_{r1}, g^y, N_r, $CERTREQ_r$
3. I → R: SPI_i, SPI_r, $E_{SK}(ID_i, CERT_i, CERTREQ_i, ID_r,$ $Sign_i(Message1, N_r, MAC_{SK}(ID_i)), SA_{i2}, TS_i, TS_r, MAC_{SK}(...))$
4. R → I: SPI_i, SPI_r, $E_{SK}(ID_r, CERT_r,$ $Sign_R(Message2, N_i, MAC_{SK}(ID_r)), SA_{r2}, TS_i, TS_r, MAC_{SK}(...))$

SPI_x = two values that together identify the protocol run and the created IKE SA

SA_{x1} = offered and chosen algorithms, DH or ECDH group

$SK = h(Ni, Nr, g^{xy})$ — actually, many different keys are derived from this

$Sign_x(Message_x, N_y, MAC_{SK}(ID_x))$ – SIGMA authentication

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted root CAs

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

$E_{SK}(..., MAC_{SK}(...))$ = Authenticated encryption for identity protection

IKEv2 notation in RFC 7296

Initial exchanges in the notation of the standard:

1. I → R: HDR(A,0), SAi1, KEi, Ni
2. R → I: HDR(A,B), SAr1, KEr, Nr, [CERTREQ]

} IKE_SA_INIT exchange

3. I → R: HDR(A,B), SK { IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr }
4. R → I: HDR(A,B), SK { IDr, [CERT,] AUTH, SAr2, TSi, TSr }

} IKE_AUTH exchange

SPI_x = two values that together identify the protocol run and the created IKE SA

Nx = nonces

SAx1 = offered and chosen algorithms, DH or ECDH group

KEx = Diffie-Hellman or ECDH key shares

IDx, CERT, CERTREQ = accepted root CAs, identity, certificate

AUTH = SIGMA authentication (signature and MAC)

SK = key material for deriving shared keys

SK { … } = authenticated encryption for identity protection

SAx2, TSx = parameters for the first IPsec SA (algorithms, SPIs, traffic selectors)

IKEv2 with pre-shared key

1. I → R: HDR(A,0), SAi1, KEi, Ni
2. R → I: HDR(A,B), SAr1, KEr, Nr
3. I → R: HDR(A,B), SK { IDi, [IDr,] AUTH, SAi2, TSi, TSr }
4. R → I: HDR(A,B), SK { IDr, AUTH, SAr2, TSi, TSr }

- Authentication with a pre-shared key between initiator and responder: AUTH is a MAC instead of a signature
  - Receiver selects the shared key based on the sender identity IDx
  - Only strong keys, no passphrases

IKEv2 with EAP

- IKEv2 supports EAP authentication

1. I → R: HDR(A,0), SAi1, KEi, Ni
2. R → I: HDR(A,B), SAr1, KEr, Nr
3. I → R: HDR(A,B), SK { IDi, [IDr,] [CERTREQ,] SAi2, TSi, TSr }
4. R → I: HDR(A,B), SK { IDr, [CERT,] AUTH, EAP }
5. I → R: HDR(A,B), SK { EAP }
6. R → I: HDR(A,B), SK { EAP(success) } // or send more EAP requests
7. I → R: HDR(A,B), SK { AUTH, }
8. R → I: HDR(A,B), SK { AUTH, SAr2, TSi, TSr }

- EAP is a framework with many authentication methods, e.g., password and SIM
- EAP for only the initiator [RFC 7296] or mutual authentication [RFC 5998]
- AUTH in messages 7-8 contains a MAC computed with the EAP MSK