Aalto University

Network Security: IKEv2 discussion

Tuomas Aura

CS-E4300 Network security
Aalto University

Internet Key Exchange (IKEv2)

1. I → R: SPI_i, SPI_r, 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 ru

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

SK = $h(Ni, Nr, g^{xy})$ — actually, 7 different keys are deriv

ID_x, $CERT_x$, $CERTREQ_x$ = identity, certificate, accepted r

SA_{x2}, TS_x = parameters for the first IPsec SA (algorithm

$E_{SK}(\ldots, MAC_{SK}(\ldots))$ = HMAC and encryption, or authenti

Which security properties?

- Secret, fresh session key
- Mutual or one-way authentication
- Entity authentication, key confirmation
- Perfect forward secrecy (PFS)
- Contributory key exchange
- Downgrading protection
- Identity protection
- Non-repudiation
- Plausible deniability
- DoS resistance

Privacy properties

- Identity protection
  - All identifiers and certificates are encrypted with the DH secret
  - Initiator reveals its identity first → vulnerable to active attacks
  - Responder authenticates initiator before revealing its identity → Responder identity protected also against impersonation attacks.
  - Why protect the responder better? Because the attacker can initiate IKEv2 key exchange with any target IP address. The target then becomes the responder
  - Special case: In mutual authentication with EAP, identity protection against active attackers depends on the EAP method
- Plausible deniability
  - Neither endpoint signs anything that would bind it to the other endpoint's identity

IKEv2 with a cookie exchange

- Responder may send a cookie (a random number) to the initiator
- Goal: verify initiator IP address; prevent DoS attacks from a spoofed IP address

1.	I → R:	HDR(A,0), SAi1, KEi, Ni	
2.	R → I:	HDR(A,0), N(COOKIE)	// R stores no state
3.	I → R:	HDR(A,0), N(COOKIE), SAi1, KEi, Ni	
4.	R → I:	HDR(A,B), SAr1, KEr, Nr, [CERTREQ]	// R creates a state
5.	I → R:	HDR(A,B), SK{ IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr }	
6.	R → I:	HDR(A,B), E_{SK} (IDr, [CERT,] AUTH, SAr2, TSi, TSr)	

How to bake a good cookie? Example:

$$COOKIE = h(K_{R\text{-periodic}}, ipaddr_I, ipaddr_R)$$

where $K_{R\text{-periodic}}$ is a periodically changing secret key know only by the responder R

Negotiated parameters

Many options add complexity and reduce inter-operability

- NAT traversal:
  - NAT detection IKE_SA_INIT exchange
  - If NAT detected, IKEv2 and IPsec are encapsulated in UDP with port 4500

- Parameters for the key exchange:
  - Protocol version and authentication method (signatures, PSK, or EAP)
  - A, B = each endpoint chooses a locally unique SPI for the IKE SA
  - SAi1, SAr1 = cryptographic algorithms for the key exchange and IKE SA (responder chooses from initiator's offer)
  - CERTREQ = sender's supported trust anchors (CAs)
  - IDr = responder identity which the initiator wants to authenticate
- Parameters for the IPsec SA pair:
  - SAi2, SAr2 = cryptographic algorithms for protecting session data SA (responder chooses from initiator's offer)
  - TSi, TSr = traffic selectors i.e. which packets to protected (responder can choose a subset of the offer)

IKE versions

- IKE(v1) [RFC 2407, 2408, 2409]
  - Framework for authenticated key-exchange protocols, typically DH
  - Multiple authentication methods: certificates, pre-shared key, Kerberos
  - Two phases: Main Mode (MM) or Aggressive Mode creates an ISAKMP SA (i.e., IKE SA) and Quick Mode (QM) creates IPsec SAs
  - Interoperability issues, complex to implement and test, incomplete spec
  - Still used, but no reason to use for anything new
- IKEv2 [RFC 7296]
  - Redesign of IKE: fewer modes and messages, simpler to implement
  - Interoperability still requires careful configuration of the endpoints