Overview of IKEv2

Dan Harkins

Charlie Kaufman

Radia Perlman

What we were trying to do

- Consolidate RFCs 2407, 2408, and 2409 in one document
- Not make gratuitous changes
- Simplify
- Fix ambiguities (commit bit, meaning of major/minor version numbers)
- Fix bugs (reflection attacks, lost messages)
- Add flexibility where it seemed necessary (e.g., traffic selectors, critical bit)
- Reduce latency
- Allow stateless cookies

Basic IKEv2

- IKE SA+IPsec SA established in 4 messages
- Exchange based on public signature keys
- Hides both identities from passive attacker
- 1st child-SA (ESP, AH, IPcomp) established during messages 3 and 4 of the IKE SA
- Future child-SAs (new IPsec SA, or rekeying of IKE SA) established in 2 messages

Forward Compatibility

- Version numbers
  - minor v# informational only. Ignored by node with smaller v#
  - major changed if protocol incompatible. Reject message if v# not supported
  - Rejection is unauthenticated
  - Major v# in header is v# of *packet*
  - Bit in header “I could do higher version”
- Critical flag in payloads (so can add new payloads and decide if it’s appropriate to reject message with those, or skip that payload)
- Critical bit only relevant for unknown payloads. All the ones in the IKEv2 draft are required to be known.

Reliability

- All messages request/response
- Messages have sequence numbers (not, as in IKEv1, random message IDs)
- Initiator is responsible for retransmission if it doesn’t receive a reply
- Multiple requests allowed in transit (e.g. in parallel setting up a bunch of child-SAs)
- Window size stated (not negotiated) in SA payload, can be different in the two directions

Traffic Selectors in v2

- “ID” payload only for IKE SA
- Child-SA uses “traffic selector” payload
- Allows lists of IP address ranges, port ranges
- Responder can narrow choice. Not just reject
- Can choose subset of ranges, or subset within a range, or say “no, must be single address pair”

Cookies

- Rather than defining IKE-SA by (c_i, c_r), treat each side’s cookie like an SPI
- Both appear in the header, so can reply to the other side’s SPI (can’t do that with ESP/AH)
- Only difference on wire from v1 is order of cookies is reversed in the two directions
- v1’s (c_i, c_r):
  - potential collision (unlikely *unless malice*)
  - Only unlikely because cookies are required to be randomly chosen (but makes stateless choice impossible)
  - “must be unique” (also prevents stateless)

Dead Peers, SA Lifetimes

- Always allowed to forget IKE-SA and all child-SAs at any time (what you’d do if you crash)
- Unauthenticated messages (ICMP, IKE “no such SPI”) raise suspicion about dead peer
- If suspicious (rate-limited) send reliable IKE message. If no reply, then delete SA
- No reason to negotiate lifetime
- If delete, send (reliable IKE) delete notification
- Deleting IKE SA automatically deletes all child-SAs
- Deleting child-SA just deletes that child-SA

Rekeying

- Either side can rekey at any time
- Rekeying of either child-SA or IKE-SA is done by creating new SA, and then deleting the old one
- Rekeyed IKE-SA inherits all the child-SAs

Encryption/Integrity Protection Format

- Complex in IKEv1 and different from anything else, weird IV calculation
- We liked the “encrypt and integrity protect this blob” syntax from the ESP spec better

IV	length depends on crypto alg, usually 8 bytes
data	encrypted
padding	encrypted
pad length	encrypted
reserved	1 byte, must be zero
integrity	includes IKE header

Negotiating Security Parameters

- SA payload in IKEv1
  - very complex
  - exponential explosion
- v2:
  - Simpler
  - Allows a proposal with “any of these algorithms for, say, encryption, with any of these algorithms for, say integrity”. Responder chooses one of each type of algorithm when accepting the P
  - I wanted to change the name from “SA” but got outvoted

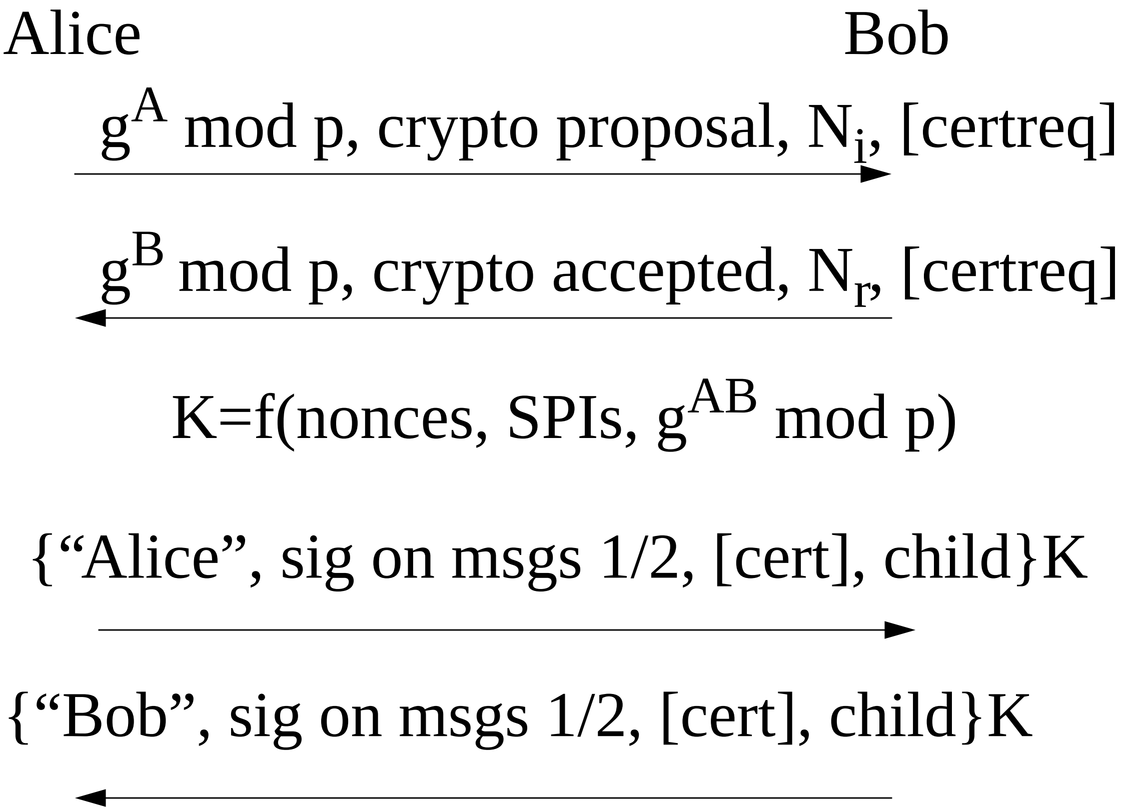
Negotiating Traffic Restrictions

- An IPsec policy thing: say “I want this SA to only carry traffic from these sources to these destinations, using these ports, etc
- IKEv1: Responder can just say “no”
- IKEv2: We added ability for responder to give subset, or say “single pair”
- Also allows sets of ranges of addresses, ports

The Exchange

- Our paper from a year ago recommended
  - have Bob prove ID first
  - and a 3-message exchange for public signature keys
- Decided instead Alice should prove ID first
  - Else trivial to poll to see who is at an address
- Decided 4 msgs better
  - piggybacking child-SA: Alice has better idea of appropriate policy
  - initiator has data to send. If no 4th msg, can’t know when OK to send the data
  - spec easier: reliability burden on initiator
  - can do stateless cookie without extra 2 msgs

The Exchange



Alice Bob

g^A mod p, crypto proposal, N_i, [certreq]

g^B mod p, crypto accepted, N_r, [certreq]

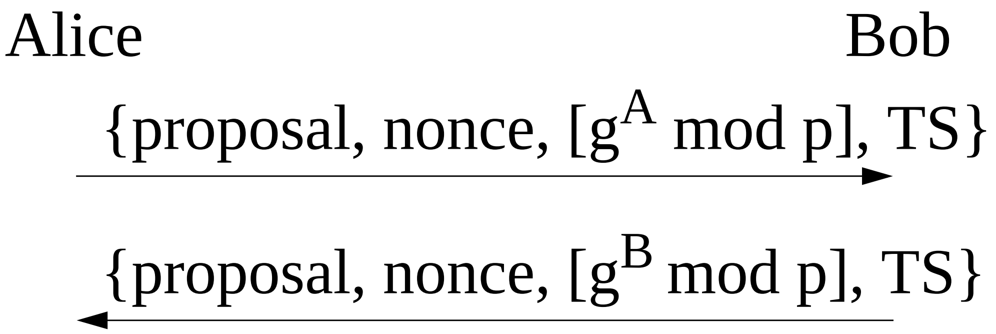
K=f(nonces, SPIs, g^{AB} mod p)

{“Alice”, sig on msgs 1/2, [cert], child}K

{“Bob”, sig on msgs 1/2, [cert], child}K

- Bob can optionally refuse 1st message and require return of stateless cookie, extra 2 msgs
- If Alice repeats info in msg 3, can avoid extra 2 msgs

Create Child-SA



Alice Bob

{proposal, nonce, [g^A mod p], TS}

{proposal, nonce, [g^B mod p], TS}

- proposal = crypto suites, SPI, protocol (ESP, AH, and/or IPcomp)
- TS=description of traffic to be sent
- Derived keys=function of IKE keying material plus nonces in this exchange, plus output of optional Diffie-Hellman

Variants

- Now that spec written, easy to modify
- The exchange is easily changed
- Things to consider
  - Bill Sommerfeld’s “birth certificate”
  - Different keys in the two directions for IKE
  - Specifying encryption/integrity format explicitly
  - Making stateless 4-message exchange
  - Preshared secret keys...weak secrets (SRP)?