Framework to Integrate Post-Quantum Key Exchanges into IKEv2


IETF 102
Agenda

- Overview of problem to be solved
- Version 02
- Questions for the WG
Overview of the Problem

- Add postquantum key exchanges to IKE
- Allow multiple key exchanges
  - So we rely on both standard DH/ECDH, and on these new fangled postquantum key exchanges
  - So we can rely on multiple postquantum key exchanges
- Deal with fragmentation
Version 02 - Strategy

- It implements ‘hybrid key exchanges’ by performing multiple consecutive exchanges
  - Uses the IKE_AUX exchanges proposed by Valery
  - The final IKE keys is secure if any of the key exchanges are secure
- All key exchanges except the first are encrypted
  - Standard IKE fragmentation applied to the later key exchanges
Version 02 - Protocol

- The initial exchange is the standard IKEv2 IKE_INIT exchange, with postquantum policy attached.

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SAi1, KEi, Ni, N(Policy)</td>
<td>HDR, SAr1, KE-Hybrid, Nr, [CERTREQ], N(AcceptedPolicy)</td>
</tr>
</tbody>
</table>

- The initiator lists what additional key exchanges it would accept
- The responder lists which set of key exchanges it agrees to
The IKE_AUX exchanges iterate through the negotiated key exchanges.

Initiator: HDR, {KEi2, Ni2} -->

Responder: HDR, {KEr2, Nr2} <--

- Each exchange is encrypted with keys based on all previous key exchanges
- Each exchange updates the keys for the next exchange

We then complete it with a standard IKE_AUTH exchange
The proposed policy format is simply a list of the sets of additional key exchanges (DH transform id's) acceptable to the initiator.

Example:

- ROUND2 + SIKE
- NTRU + SIKE
- ROUND2
- NTRU

This is in addition to the initial KE performed in the IKE_INIT.

The responder replies with the set that matches his policy.

Example:

- ROUND2 + SIKE
Comments and Suggestions are Welcome

- This is a work in progress
- We tried to keep things simple
- We felt it would easier to add new requirements from the working group, than it would be to take them out
Open Questions for the Group

- Do you agree with this general approach?
- Do you agree with the strategy of treating classical and postquantum key exchanges equivalently?
- Should we allow multiple key exchanges per exchange?
- How do we encode the policy?
- No native support for key shares > 64k
  - Only about 25 out of 175 NIST submissions would require that.
Thank You