DH group  Key Exchange Method issues

- Initial Exchanges create a Child SA with DH of IKE SA setting.
  - No consistent behaviour if configuration IKE DH \(!=\) Child DH
  - Neither end knows if a rekey will require PFS.
    - and if so, which DH to use.

- Cause: IKE_AUTH should have contained Child DH proposal.
- Problem: Humans keep using different DH for IKE and ESP.
- Result: IPsec connection works, then fails hour(s) later.....
DH Issues: Initial Exchanges

- If negotiated IKE DH is not a valid configured Child DH:
  - On Responder:
    - If pfs=no, no issues
    - If pfs=yes, return NO_PROPOSAL_CHOSEN?
    - If pfs=yes, assume peer has same Child DH?
  - On Initiator:
    - If pfs=no, no issues
    - If pfs=yes, refuse configuration to load?
    - If pfs=yes, send Informational Delete after negotiation?
    - If pfs=yes, assume peer has same Child DH? (problematic)
DH Issues: REKEY of Initial Child

- Peer’s configured child DH group(s) has not been negotiated yet and is unknown
- Rekey with pfs SHOULD use same DH group
- If rekey proposed Child DH is not IKE SA DH:
  - On Responder:
    - Return INVALID_KE(dh) ? [wrong, rekey DH group matches KE payload]
    - Return NO_PROPOSAL_CHOSEN ? [right, but too confusing]
    - Return another new error code? UNEXPECTED_DH KE or something?
    - Accept any/none DH, immediately rekey IKE SA to gain pfs with IKE DH
DH Issues: Additional complications

- Microsoft Windows IKEv2 configured for DH14, rekeys with DH2
  - Probably thinks DH14 is for IKE SA, libreswan has ms-dh-downgrade=yes|no
- draft-ietf-ipsecme-ikev2-sa-ts-payloads-opt does not know PFS / DH settings
- Implementation interop issues with DH/KE transform “DH_NONE” vs no transform
- Implementation interop issues with initiator rekey using configured parameters instead of established parameters
Libreswan interoperability issues

<table>
<thead>
<tr>
<th>esp</th>
<th>pfs</th>
<th>ms-dh</th>
<th>w/o</th>
<th>AUTH CHILD</th>
<th>REKEY CHILD</th>
<th>v4.5 AUTH CHILD</th>
<th>v4.6 AUTH CHILD</th>
<th>second CHILD</th>
</tr>
</thead>
<tbody>
<tr>
<td>algs</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>algs</td>
<td>OLD-none</td>
<td>algs:REKEY none</td>
<td>algs:REKEY none</td>
<td>OLD:REKEY none</td>
</tr>
<tr>
<td>algs</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>algs</td>
<td>algs:none</td>
<td>algs:REKEY none</td>
<td>algs:REKEY none</td>
<td>OLD:REKEY none</td>
</tr>
<tr>
<td>algs</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>algs</td>
<td>OLD-none</td>
<td>algs:REKEY none</td>
<td>algs:REKEY none</td>
<td>OLD:REKEY none</td>
</tr>
<tr>
<td>algs</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>algs</td>
<td>algs:none</td>
<td>algs:REKEY none</td>
<td>algs:REKEY none</td>
<td>OLD:REKEY none</td>
</tr>
<tr>
<td>algs</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>algs</td>
<td>OLD-IKE</td>
<td>algs:REKEY none</td>
<td>algs:REKEY none</td>
<td>OLD:REKEY none</td>
</tr>
<tr>
<td>algs</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>algs</td>
<td>OLD-IKE, OLD-none</td>
<td>algs:REKEY none</td>
<td>algs:REKEY none</td>
<td>OLD:REKEY none</td>
</tr>
</tbody>
</table>
Possible solutions

- Disallow IKE DH != Child DH
  - does not fix install base, but will reduce problem over time.
  - Authors of RFC8247 already tried to suggest this to WG at the time :-)
- On responder, if Initial Exchange IKE DH != Child, return NO_PROPOSAL_CHOSEN
  - and use the Childless IKE SA to CREATE_CHILD_SA Child SA with proper DH
    - Causes race conditions and/or interoperability issues
- In Initial Exchanges, add DH to proposals if pfs=yes (will prob break things)
- On responder, do IKE rekey if Child DH insufficient (doesn’t help initiator case)
- In IKE_AUTH, exchange a new CHILD_SA_KE(dh,..) notify
  - Return INVALID_KE / NO_PROPOSAL_CHOSEN if new notify mismatched
Questions for the IPsecME WG

- Q1: Is a new Notify CHILD_SA_KEY(dh,...) worth publishing?
- Q2: Is a new Notify Error code useful?
- Q2: Is it useful to write up a “DH behaviour updates” doc, updating RFC 7296?
- Q3: Did we miss additional issues or other obvious solutions?