Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security

draft-smyslov-ipsecme-ikev2-qr-alt

Valery Smyslov
svan@elvis.ru

IETF 116
PPK for IKEv2

Defined in **RFC 8784**:

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IKE_SA_INIT</strong></td>
<td><strong>IKE_SA_INIT</strong></td>
</tr>
<tr>
<td>HDR,SAi1,KEi,Ni,N(USE_PPK)</td>
<td>HDR,SAr1,KEr,Nr,N(USE_PPK)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IKE_AUTH</strong></td>
<td><strong>IKE_AUTH</strong></td>
</tr>
<tr>
<td>HDR,SK{IDi,AUTH,SAi2,TSi,TSr,N(PPK_IDENTITY)[,N(NO_PPK_AUTH)]}</td>
<td>HDR,SK{IDr,AUTH,SAr2,TSi,TSr,N(PPK_IDENTITY)}</td>
</tr>
</tbody>
</table>
The Problem

- Initial IKE SA is not protected with PPK (WG decision)
  - it was assumed that no sensitive information was transferred over initial IKE SA, and one could immediately rekey it to get protection
- G-IKEv2 (draft-ietf-ipsecme-g-ikev2) uses initial IKE SA to immediately transfer session keys from Group Controller/Key Server (GCKS) to Group Member (GM)
  - these keys are not protected with PPK

\[
\begin{align*}
\text{GM} & \quad \text{GCKS} \\
\text{IKE_SA_INIT} & \quad \text{IKE_SA_INIT} \\
\text{HDR,SAi1,KEi,Ni,N}(&\text{USE_PPK}) & \quad \text{HDR,SAr1,KEr,Nr,N}(&\text{USE_PPK}) \\
\text{GSA_AUTH} & \quad \text{GSA_AUTH} \\
\text{HDR,SK\{IDi,AUTH,IDg,SAg,} & \quad \text{HDR,SK\{IDr,AUTH,N}(&\text{PPK_IDENTITY}), \\
\text{N(PPK_IDENTITY)[,N(NO_PPK_AUTH)]}\} & \quad \text{GSA,KD}}
\end{align*}
\]
Current Use of PPK with G-IKEv2

Currently G-IKEv2 draft suggests the following sequence of exchanges to get the protection with PPK:

<table>
<thead>
<tr>
<th>GM</th>
<th>GCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IKE_SA_INIT</strong>&lt;br&gt; HDR, SAi1, KEi, Ni, N(USE_PPK)</td>
<td><strong>IKE_SA_INIT</strong>&lt;br&gt; HDR, SAr1, KEr, Nr, N(USE_PPK)</td>
</tr>
<tr>
<td><strong>GSA_AUTH</strong>&lt;br&gt; HDR, SK{IDi, AUTH, IDg, SAg, N(PPK_IDENTITY) [, N(NO_PPK_AUTH)]}</td>
<td><strong>GSA_AUTH</strong>&lt;br&gt; HDR, SK{IDr, AUTH, N(PPK_IDENTITY), N(REKEY_IS_NEEDED)}</td>
</tr>
<tr>
<td><strong>CREATE_CHILD_SA</strong>&lt;br&gt; HDR, SK{SAi, KEi, Ni}</td>
<td><strong>CREATE_CHILD_SA</strong>&lt;br&gt; HDR, SK{SAr, KEr, Nr}</td>
</tr>
<tr>
<td><strong>INFORMATIONAL</strong>&lt;br&gt; HDR, SK{D}</td>
<td><strong>INFORMATIONAL</strong>&lt;br&gt; HDR, SK{}</td>
</tr>
<tr>
<td><strong>GSA_REGISTRATION</strong>&lt;br&gt; HDR, SK{IDg, SAg}</td>
<td><strong>GSA_REGISTRATION</strong>&lt;br&gt; HDR, SK{GSA, KD}</td>
</tr>
</tbody>
</table>
Alternative Approach

Proposed in draft-smyslov-ipsecme-ikev2-qr-alt:

IKE_SA_INIT
HDR,SAi1,KEi,Ni,N(USE_PPK),
N(INTERMEDIATE_EXCHANGE_SUPPORTED)

IKE_SA_INIT
HDR,SAr1,KEr,Nr,N(USE_PPK),
N(INTERMEDIATE_EXCHANGE_SUPPORTED)

IKE_INTERMEDIATE
HDR,SK{...N(PPK_IDENTITY)
[,N(PPK_IDENTITY)++]}

IKE_INTERMEDIATE
HDR,SK{...N(PPK_IDENTITY)}

GSA_AUTH
HDR,SK{IDi,AUTH,IDg,SAg}

GSA_AUTH
HDR,SK{IDr,AUTH,GSa,Kd}
Fallback to RFC 8784

- If the responder doesn’t support this extension, then it doesn’t respond with any PPK_IDENTITY in IKE_INTERMEDIATE
  - the initiator MAY fallback to RFC 8784 in this case
  - the same situation happens if the responder isn’t configured with any of the proposed PPK_IDs
    - no need to fallback to RFC 8784 in this case, but allowed in the draft for simplicity
- It is possible to modify draft to distinguish between these two cases and disallow fallback if extension is supported, but no PPK found
Double PPK

• Do we need to support using both RFC 8784 and this draft’s approaches for a single SA?
  – Currently is not supported in the draft
  – It seems that this is too complex with no benefits
    • Should be explicitly prohibited in the draft?
Session Keys Calculation

• RFC 8784:

\[
\text{SKEYSEED} = \text{prf}(\text{Ni} \mid \text{Nr}, g^{ir})
\]

\[
\{\text{SK}_d', \text{SK}_ai, \text{SK}_ar, \text{SK}_ei, \text{SK}_er, \text{SK}_pi', \text{SK}_pr'}\} = \\
\text{prf} + (\text{SKEYSEED}, \text{Ni} \mid \text{Nr} \mid \text{SPIi} \mid \text{SPIr})
\]

\[
\text{SK}_d = \text{prf} + (\text{PPK}, \text{SK}_d')
\]

\[
\text{SK}_pi = \text{prf} + (\text{PPK}, \text{SK}_pi')
\]

\[
\text{SK}_pr = \text{prf} + (\text{PPK}, \text{SK}_pr')
\]

• This proposal

\[
\text{SKEYSEED'} = \text{prf} + (\text{PPK}, \text{SK}_d)
\]

\[
\{\text{SK}_d, \text{SK}_ai, \text{SK}_ar, \text{SK}_ei, \text{SK}_er, \text{SK}_pi, \text{SK}_pr} = \\
\text{prf} + (\text{SKEYSEED'}, \text{Ni} \mid \text{Nr} \mid \text{SPIi} \mid \text{SPIr})
\]
Mismatched PPK

• If PPKs with the same PPK_ID are different, then we run into the problem that the responder cannot decrypt and authenticate IKE_AUTH messages and will drop them
  – MUST be fixed in the next version of the draft
    • need to have key confirmation payload in the IKE_INTERMEDIATE exchange, perhaps prf(PPK, Ni | Nr | SPIi | SPIr)
    • Who should send it – initiator or responder? Seems like more appropriate for initiator
Comparison

• For G-IKEv2:
  – 3 exchanges instead of 5
  – 1 DH shared key computation instead of 2
  – 1 computation of AUTH in case of optional PPK instead of 2
  – initiator can propose several PPK_IDs

• Can also be used in IKEv2:
  – 3 exchanges instead of 2
    • but PPK_ID can be piggybacked if IKE_INTERMEDIATE is also used for other purposes
  – 1 computation of AUTH instead of 2 if PPK is optional
  – initiator can propose several PPK_IDs
Coexistence

- The proposed approach is **not intended to replace** the existing one, both can co-exist:
  - for G-IKEv2 the proposed approach can be a primary one (or the only one?)
  - for IKEv2 the proposed approach can be an alternative one (e.g. if IKE identities need to be protected)
Implementations

• At least 2 implementations of -06 draft exists:
  – ELVIS-PLUS
  – libreswan

• Successfully interoperated during hackathon
Thanks

- Comments? Questions?
- More details in the draft
- WG adoption?