What Happened?

- Draft -07 completed WG last call following Beijing
  - Vincent provided us with a comprehensive review
  - IANA discovered some minor issues with the GDOI Payloads registry
  - Authors discovered some issues with the counter mode logic
  - Cleaned up the Security Considerations text
- Draft -08 was published in early March
Miscellaneous

• Added an *Acronyms and Abbreviations* section to increase reading comprehension.

• Moved sections not critical to understanding the protocol to Appendices
  – Requirements on extending the protocol
  – Discussion of applications

• Made terminology consistent
  – (GM, GCKS) replaces (Initiator, Receiver) everywhere now
Additions

- Centralized the GCKS counter modes procedures in Section 3.5 Counter-modes of operation
- Added Section 7.4.2 Backward Access Control Requirements section
- Added Section 7.5 Derivation of keying material section clarifying requirements on keying material
IANA Changes

- Improper IANA terms used
- Many namespaces are 2-byte values, yet only values 0-255 was described.

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
</tr>
<tr>
<td>KEK_ALG_DES</td>
<td>1</td>
</tr>
<tr>
<td>KEK_ALG_3DES</td>
<td>2</td>
</tr>
<tr>
<td>KEK_ALG_AES</td>
<td>3</td>
</tr>
<tr>
<td>Standards Action</td>
<td>4-127</td>
</tr>
<tr>
<td>Private Use</td>
<td>128-255</td>
</tr>
<tr>
<td>Unassigned</td>
<td>256-32767</td>
</tr>
</tbody>
</table>
Allocation of SIDs (old)

• In -07 a GM would request Sender ID (SID) values in the 2\textsuperscript{nd} message using a GAP payload
  – This was before it knew whether or not there would be counter modes in the policy
  – This is awkward: does it predict that needs SIDs, and if so how many?

• There are two cases where a GM might want more than 1 SID
  – It has a high-speed interface and will burn through its sequence number too quickly
  – It will be installing SAs in >1 encryption engine
Allocation of SIDs (new)

- Upon receipt of the SA payload, the GM now detects the use of a counter mode. It then can determine how many SIDs it might need. If it needs more than 1, it will add a GAP payload requesting that many.
- Upon receipt of the GAP payload, the GCKS allocates the requested # of SIDs, and returns them in the KD payload. Otherwise, it returns one SID in the KD payload.

<table>
<thead>
<tr>
<th>Group Member</th>
<th>GCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>HDR*, HASH(1), Ni, ID</td>
<td>--&gt;</td>
</tr>
<tr>
<td>HDR*, HASH(2), Nr, SA</td>
<td>&lt;--</td>
</tr>
<tr>
<td>HDR*, HASH(3) [,GAP]</td>
<td>--&gt;</td>
</tr>
<tr>
<td>HDR*, HASH(4), [SEQ,] KD</td>
<td>&lt;--</td>
</tr>
</tbody>
</table>
Allocation of SIDs (GCKS)

- Recall: an SID *must* be allocated to one GM only, and the GCKS *must* do so reliably.
- We clarified the allocation method in the draft, *keeping it simple*.
- Claim:
  “Using the method [on the next slide], at no time can two group members use the same IV values with the same Data-Security SA key.”
Allocation of SIDs (GCKS details)

1. Initialize a counter to 0
2. Increment the counter once per SID.
3. Give each sender 1 SID, or as many as they require
4. Allocate an SID in every GROUPKEY-PULL
5. When the SID counter reaches its last value, reset to 0, create new SAs, delete old SAs, distribute new SAs
6. In a rekey, send a DELETE to delete all old SAs, which causes GMs to re-register and get new SIDs and new SAs.
Next Steps

• Re-review
• Send to AD