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

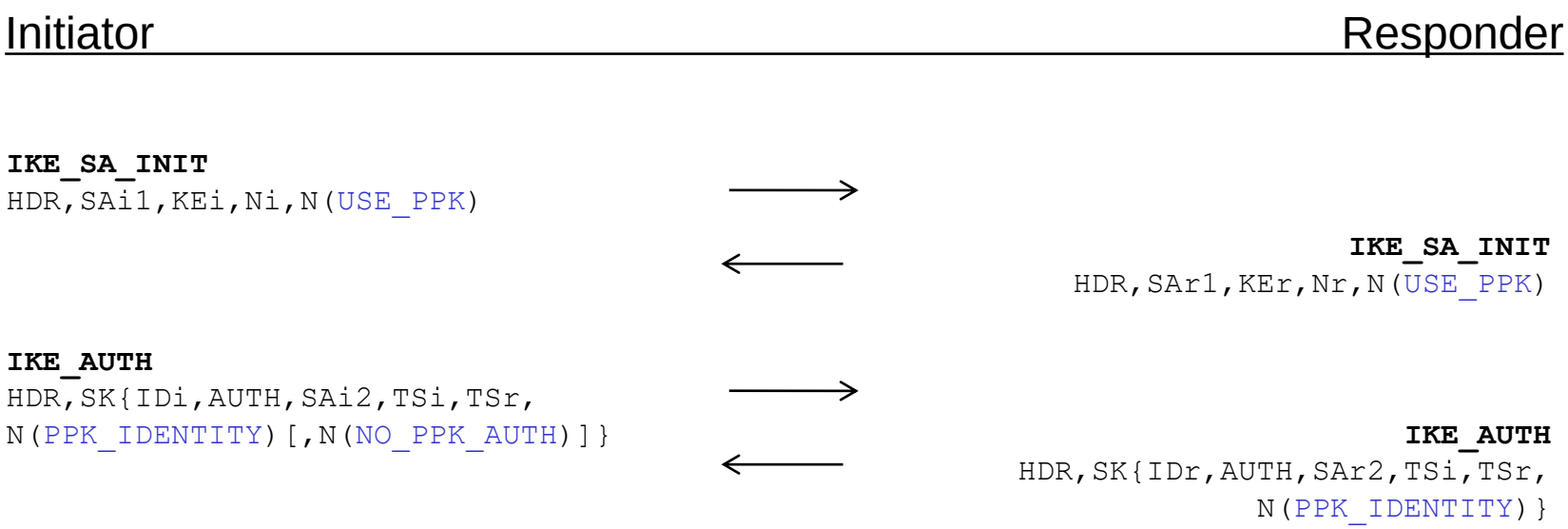
draft-smyslov-ipsecme-ikev2-qr-alt

Valery Smyslov
svan@elvis.ru

IETF 116

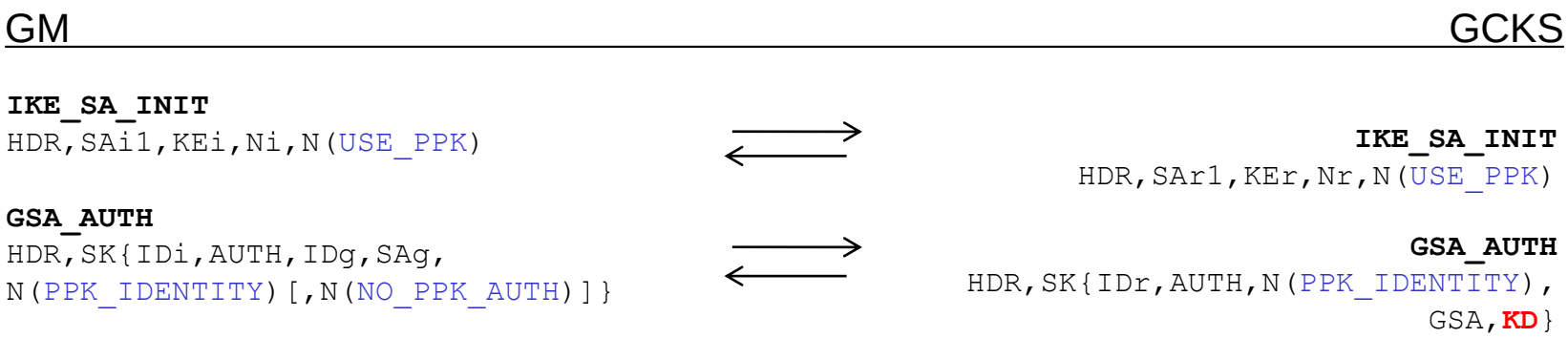
PPK for IKEv2

Defined in RFC 8784:



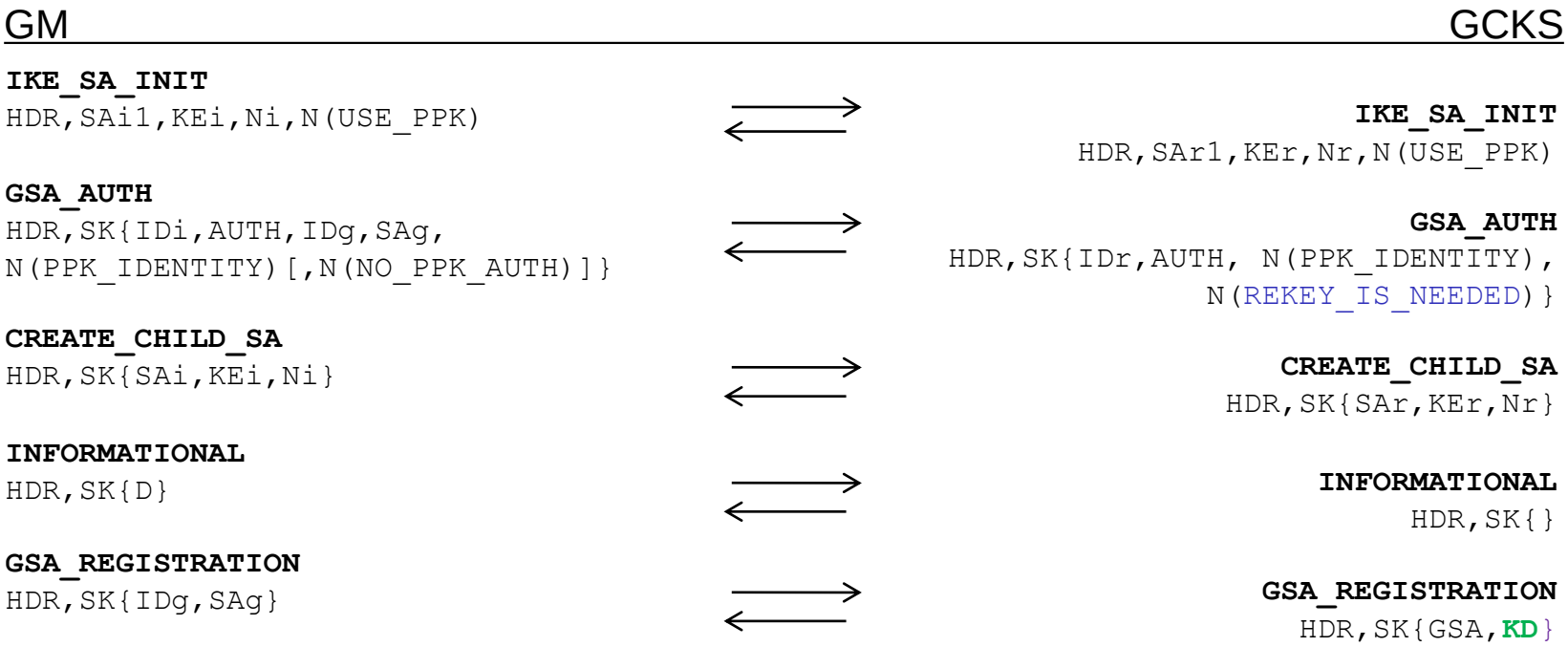
The Problem

- 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



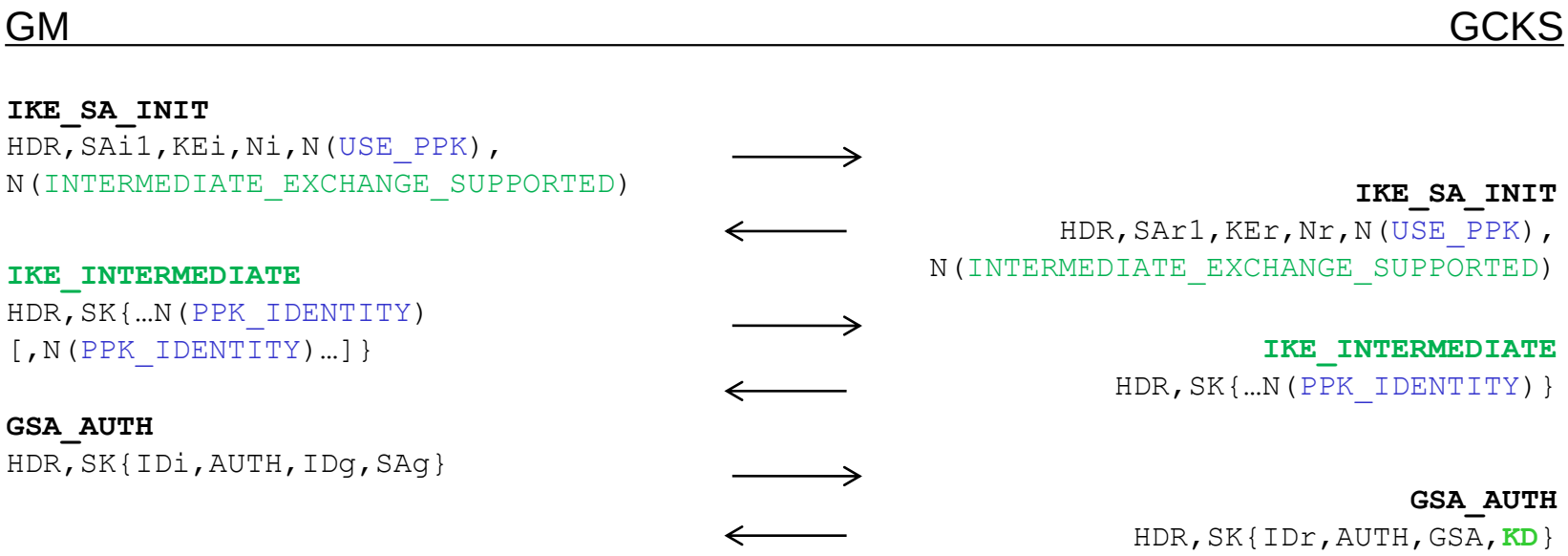
Current Use of PPK with G-IKEv2

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



Alternative Approach

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



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:

```
SKEYSEED = prf(Ni | Nr, g^ir)
{SK_d` | SK_ai | SK_ar | SK_ei | SK_er | SK_pi` | SK_pr`} =
                                    prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)
SK_d = prf+ (PPK, SK_d')
SK_pi = prf+ (PPK, SK_pi')
SK_pr = prf+ (PPK, SK_pr')
```

- This proposal

```
SKEYSEED' = prf+ (PPK, SK_d)
{SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr} =
                                    prf+ (SKEYSEED', Ni | Nr | SPIi | 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_INTERMRDIATE 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?