draft-fluhrer-qr-ikev2-01

Authors: Scott Fluhrer, David McGrew, Panos Kampanakis

Problem to be solved

IKEv1 (with preshared keys) is quantum resistant; IKEv2 is not

> "CSfC deployments involving an IKE/IPsec layer may use RFC 2409-conformant implementations of … IKEv1… RFC 2409 is the only version of the IKE standard that leverages symmetric pre-shared keys in a manner that may achieve quantum resistant confidentiality" --- NSA web site.

We don't want to give people a reason to use IKEv1

Our Solution

Give each sides of IKEv2 a ‘postquantum preshared key’ (PPK)

- This is a password just like the IKE preshared keys.

Have both sides stir it in as a part of the key (SK) generation

- This is the same method that IKEv1 uses
- If the PPK has 256 bits of entropy, the IKE keys are quantum secure.

This PPK does not replace the authentication that IKEv2 uses.

The major complexity of the design is having both sides agree on which PPK to use (as key generation happens before the responder learns the initiator’s identity).

Modification to SK Generation

We modify the SKEYSEED/SK computation to be:

SKEYSEED = prf+(prf(PPK, Ni) | prf(PPK, Nr), g^{ir})

(SK_d, ai, ar, er, pi, pr) = prf+(SKEYSEED,
prf(PPK, Ni) | prf(PPK, Nr) | SPIi | SPIr)

Whereever Ni, Nr appear, we replace it with prf(PPK, Nx)

If PPK has 256 bits of entropy, recovering anything would take an expected $O(2^{128})$ operations with Grover's algorithm

We believe that to be infeasible.

PPK Agreement

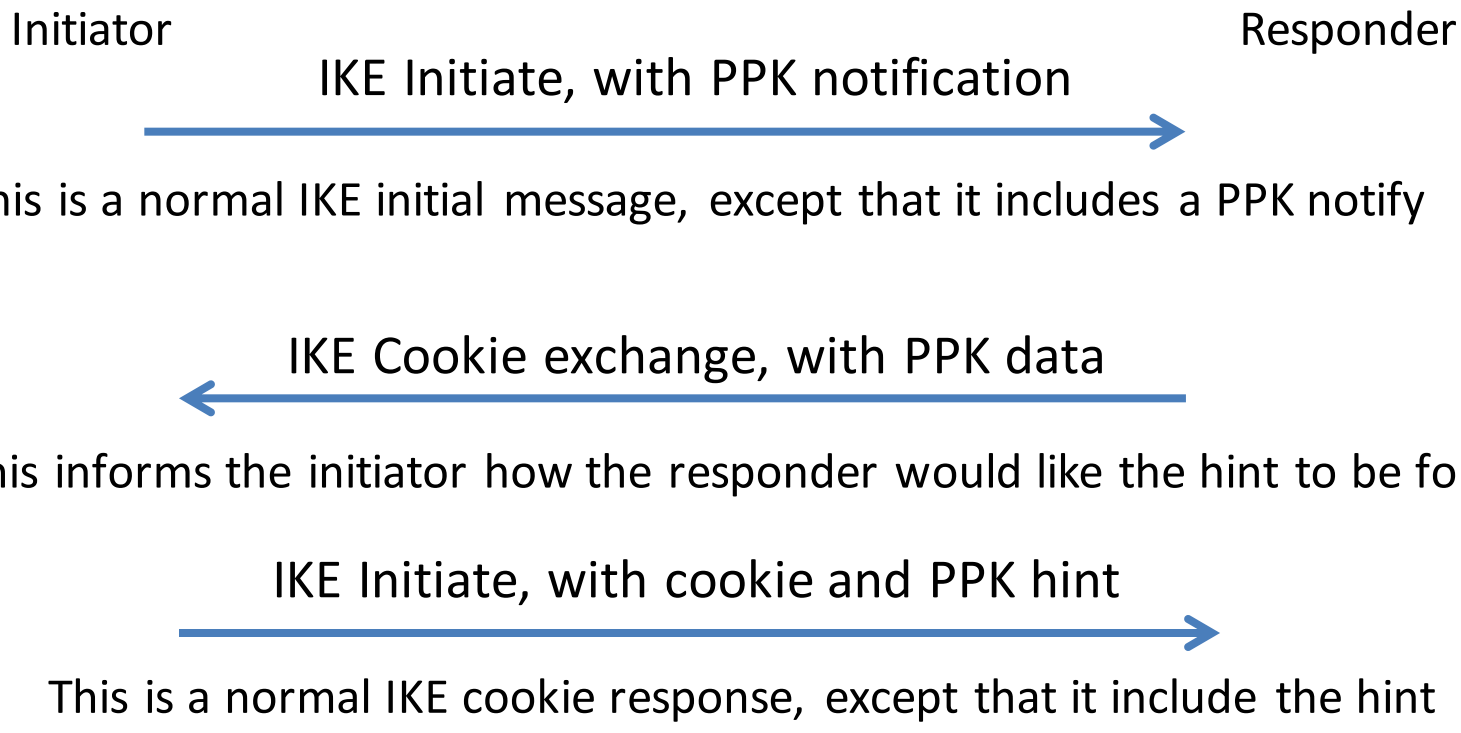
Problem: which PPK we use depends on the identity of the peer, but the initiator needs to use the PPK before it learns it.

Solution: the initiator gives a ‘hint’ about which PPK to use.

- Constraint: this ‘hint’ shouldn’t leak any information about who we’re connecting to
- So, we give an encoded version of the PPK;
  - Someone who knows the PPK can recognize it
  - Someone who doesn’t know it learns nothing

PPK Agreement

We have a three pass protocol:



PPK Agreement

Why we do this three pass method:

- It allows the responder to specify how the hint is to be specified
  - Large IKE responders may have thousands of PPKs; making them totally nondetermanistic may lead to large search times
  - An IKE responder may decide to make the hint deterministic, making for fast search times.
- The initiator can also specify the hint algorithm

Indicator Algorithm

The draft defines an initial algorithm of:

Indicator = AES256(HMACSHA256(PPK, “A”), Input)

Reasons:

- Efficiency; we want responders to scan through their list of PPKs quickly
  - They could precompute the values based on a limited number of inputs, but that would leak anonymity
- Security
  - Given R, AES256(PPK1, R), S, AES256(PPK2, S), it is infeasible to determine whether PPK1 == PPK2

Questions?