Just Fast Keying (JFK)

Angelos Keromytis (Columbia University)

Bill Aiello (AT&T Labs - Research)
Matt Blaze (AT&T Labs - Research)
Steve Bellovin (AT&T Labs - Research)
Ran Canetti (IBM T.J. Watson Research Center)
John Ioannidis (AT&T Labs - Research)
Omer Reingold (AT&T Labs - Research)

Draft

- draft-ietf-ipsec-jfk-01.txt

JFKi Protocol

I->R: Ni, g^i

R->I: Ni, Nr, GRPINFO, IDr, g^r, Sig(g^r, GRPINFO),
HMAC{Hkr}(Nr, g^r, Ni, g^i, IPi)

I->R: Ni, Nr, g^i, g^r, CK,
HMAC{Hkr}(Nr, g^r, Ni, g^i, IPi)
E{Ke}(IDi, sa, Sig(Ni, Nr, g^i, g^r, IDr, sa))

R->I: E{Ke}(Sig(Ni, Nr, g^i, g^r, IDr, sa, sa'), sa')

JFKr Protocol (LBJ)

I->R: Ni, g^i

R->I: Ni, Nr, GRPINFO, g^r,
HMAC{Hkr}(Nr, g^r, Ni, IPi)

I->R: Ni, Nr, g^i, g^r, HMAC{Hkr}(Nr, g^r, Ni, IPi),
C = E{Ke1}(IDi, sa, Sig(g^i, g^r, Ni, Nr, GRPINFO)),
HMAC{Ka}('I', C)

R->I: D = E{Ke2}(IDr, sa', Sig(g^i, g^r, Ni, Nr)),
HMAC{Ka}('R', D)

Changes

- JFKr seems to be preferable
- ID protection
- Proof of security
- SA deletion (0 lifetime SAs)
- Phase 2 (none)
- SAs
  - Ciphersuites
  - Ranges for TS

To Come

- IPi added in authenticator
- Cert verification result caching
- Jane/Tarzan (no JFK policy)
- Reuse of same g^i/g^r means faster (re)keying
  - No need to negotiate
  - 1 RSA sign/verify operation in each direction
- Fragmentation attack avoidance possible with 4 messages

To Come (cont.)

- Easy computation DoS/flash-crowd management
  - Queue of exponentials (generate 1/N secs)
  - Keep using next-in-queue
  - If out of exponentials, reuse last one
  - DoS prevention through flow control
    - No need for detection

Last words

- Preshared key authentication is possible
- So is no-pubkey-op (re)keying

- Are these really needed ?