Tweaking MACsec Cipher Suites

Mick Seaman
mickseaman@gmail.com

Improved performance/reduced effort for 802.1X MKA[1] participants as Group CA membership changes by substituting explicit nonce block use for fresh SAK[2] distribution by existing Key Server. Retains replay protection, interface (SA rx/tx) operational indications, as well avoiding nonce reuse by restarted participants. Retains current threat model – attacker can selectively (per participant) remove, copy, delay, repeat MKPDUs – and no new dependencies added.

[1] MKA: MACsec Key Agreement

[2] SAK: Dta Key

Agenda

Currently:

- When are fresh SAKs distributed
- What does SAK distribution achieve/guard against
- How does SAK distribution work

Tweaking Cipher Suites:

- {SAK, nonce space} distribution to existing objectives
- Minimizing MKPDU transmissions

Fresh SAK currently after..

1. Reauthentication – to provide PFS.
2. Pending nonce exhaustion – to guard against nonce repetition/misuse.
3. Changed Key Server – new Key Server not guaranteed to have record of prior, potentially restarted, participants use of nonce space.[1]
4. Resumption after suspension – being cautious.
5. Participants join – potentially reusing or reallocating nonce space.[2]
6. Participants leave – may be stolen for intrusive analysis.[3]

Goal of Cipher Suite tweaking is to improve performance/reduce effort for #5 when #6 is not a threat. May address #4 but not covered by this presentation, not intended to change fresh SAK requirement for #1, #2, #3.

[1] A given SAK's potential nonce space is allocated between participants. For GCM-AES-128/256, nonce includes transmitter's SCI (MAC Address.PortNum). For GCM-AES-XPN-128/256 Key Server/rules allocate SSCI (Short SCI). CA participants only need to use one SAK (two during rollover) at a time, irrespective of group size.

[2] Restarted participants may have no record of their past participation.

[3] CAK, ICK, KEK may be held in secure module. Unlikely that SAK will be. Not a realistic threat in application environments targeted by this presentation, and SAK change omitted for this case.

Current SAK distribution

SAK distribution specified by the CP state machine.

- Distributes SAK to new CA participants and to existing participants
- Supports lossless rollover old SAK to new SAK
- Only ‘Live List’ participants can transmit with new SAK
  - Guards against duplicate SCI (MAC Address.Port Number) nonce part
- Key Server decides/communicates when tx with new SAK begins
  - New participant only receives with new SAK, guarding against replay
  - Participants indicate readiness to receive, but Key Server can timeout

Tweaking MACsec Cipher Suites

For the purpose of this presentation a “tweakable Cipher Suite” is one for which there is a clear and efficient mechanism for allocating blocks of the nonce to the set of CA participants.

This presentation considers Cipher Suites that are tweakable by changing the Key Number (KN) while retaining the existing SAK.

The existing GCM-AES-XPN Cipher Suites can be tweaked, as the KN is included in the (public) Salt.

The current Ascon proposal also includes a KN Salt component, with a view to tweakability.

If the cryptographic element of any given Cipher Suite is potentially tweakable but requires particular attention to SAK.nonce distribution rules this could be signaled as an additional TLV in MKA. It could also be signaled by assignment of a further Cipher Suite Identifier (identifying both the cryptography and its use).

Tweakable SAK.KN distribution

Retaining the CP state machine externally observable behavior, but with additional intelligent Key Server and Participant behavior:

- Key Server knows all participants tweak-capable.
- On SAK.KN distribution participants also install the SA for SAK.KN+1 for receive (no need to signal rx on for that SA).
- Key Server does not distribute a fresh SAK on participant leaving [1]
- Key Server distributes SAK.KN+1 on participant(s) joining [2]
- Joining participants send MKPDU with rx enabled for SAK.KN+1 SA
- Key Server sends MKPDU with tx enabled for SAK.KN+1 SA, all participants can now tx and rx with SAK.KN+1 SA. [3]

[1] Participant leaving likely not detected by timeout, but by new participant with the same SCI. The Key Server should distinguish a participant restart from true SCI duplication.

[2] Subject to timing, allowing for multiple participants joining, possible using supplementary information as to expected participants. Proof of liveness required from each participant to Key Server: worst case, one joining, three MKPDU exchange prior to SAK distribution; best case one MKPDU from Key Server, one MKPDU from each joining participant. Then MKPDU from Key Server to distribute SAK.

[3] Continuing participants only need to validate 1 MKPDU, enabling tx with SAK.KN+1.

SAK.KN distribution impact (summary)

New participants:

- Have to prove liveness to Key Server (as always)
- Receive distributed SAK (as always)
- Signal readiness to receive (send MKPDU)
- Receive transmit permission from Key Server (receive MKPDU)

Existing participants:

- Receive distributed SAK with KN+1, tx disabled: iff SCs for new participants not already installed;
  - validate MKPDU, check SAK wrap unchanged, install missing SCs (needs MKPDU from each new participant)
  - transmit MKPDU with rx enabled for KN+1 SA
- Receive distributed SAK with KN+1, tx enabled
  - validate MKPDU, check SAK wrap unchanged
  - start transmission with KN+1 SA

Cipher Suite specific considerations

Each participant's nonce space (excluding KN refinement) needs to remain unchanged for optimal tweaking:

- Not a problem when full SCI included in nonce.
- For CGM-AES-XPN, each participant's SSCI to remain unchanged.
  - Specify 'unused MI' value and SSCI allocation per Live List ordering. Accommodates both pre-allocation of SSCIs and cumulative allocation.