LinkSec Frame Formats?

David Johnston
david.johnston@ieee.org
dj.johnston@intel.com

What is needed?

- A sequence counter
  - PN/IV
- An integrity check
  - MIC/MAC/ICV
- Ciphertext
  - ICV may be part of ciphertext
- SAID
  - If there is no other context (e.g. CID)
  - Must assume its presence for a general crypto protocol

In What Order (ICV)

- ICV is computed over transmitted packet
  - Easier at end, since ICV can then be computed on the fly
- ICV is computed on received packet and compared with received ICV
  - Easier at end, since the received ICV doesn't need to be stored during ICV computation
  - Inserting between PN and ciphertext can provide time for header and AD computation for CCM and OCB type algorithms
- On balance, ICV goes at end. No surprise..

In what Order (PN)

- Generally a nonce needs to be constructed before any crypto operations can happen
  - PN goes as early as possible in the frame

In what Order (SAID)

- Needed as early as possible to allow early retrieval of keys

In What Order (ciphertext)

- Ciphertext goes pretty much where the plain text goes.
  - Good for cipherstream and in-place (OCB like) modes.
  - Minimizes intermediate storage in datapath

In What Order (Header Data)

- Header data consists of authenticated and non authenticated parts
  - E.G. HCS might be non authenticated
- A gap in time between nonce construction and data encryption is useful in both CCM and OCB
  - Time to get that initial block encryption and AD processing complete before decryption

The Ideal Security Packet Format

- Doesn't map to the 802 encapsulation style
  - Need a compromise
  - SAID, PN and ICV go in data field
  - Header data is as defined in the MAC/PHY spec

- Again, no surprises

ICV Size

- Birthday attack susceptible modes
  - Strength proportional to sqrt(2^n) where n is the number of bits
- Non birthday attack susceptible modes
  - Strength proportional to 2^n
- Does not interact with data rate, rekeying rate etc
- If we avoid birthday attack susceptible modes, then can have half the ICV length for the same strength
  - 64 bits seems ok
  - You might notice 2^63 ICV guessing attempts before one succeeds
- 8 octet ICV

PN Size

- PN Strength is measured in time to rekey
  - Longer is good
- Inversely proportional to packet rate
  - 802 has variable packet rates due to variable packet size and asynchronous MAC service
- Minimum rate ≈ 10kbps (.15.4)
- Max rate ≈ 100Gbps (Arbitrary optical thing)
- 10^7 difference ≈ 23.5 bits
  - **No good PN size for all cases**
  - Slowest will want 24 fewer bits than fastest for optimal operation
  - Gap can only widen in the future

PN Size Options

- Pick suitable case for highest packet rate interface
  - Slower devices suffer the consequences
  - Poor adoption. We waste our time
- Tie PN size to PHY type
  - Those pesky provider bridges mess things up again
- Flexible PN length
  - PN representation size grows with PN magnitude
  - Bad frame formatting implementation ramification
  - Most time spent with large PN bit width anyway
- Code PN size in header
  - Bad security implications
  - Can spoof packet with weak crypto and crack it
- Negotiable PN size
  - Need to be careful about security implications
    - Potential for attacker to force negotiation of weaker mode
  - Could be negotiated at time of SA formation and remain constant for lifetime of the SA
- **Only reasonable option appears to be to negotiate a PN size**

PN Negotiation

- For a given technology, the PN size should have some default based on max packet rate
  - So in non provider bridge case, any negotiation should be very short – both sides will agree the same value
- In provider bridge case is it OK to default to smaller PN choice of the two ends?
  - Slower device limits the max packet rate
  - Probably should make sure that crypto in the negotiation leads to assurance that we in fact DO have nothing smaller than the minimum acceptable to the slower device
  - Maybe higher speed spoofing can be performed on the high speed side of the bridge, attacking the shorter PN
    - Needs rate limit detection. Starts getting messy

PN Negotiation

- At what level do we negotiate?
  - PN size as applied to a particular algorithm?
  - Between algorithms that offer a variety of PN sizes?
- PN size negotiation might lead to improper use of a crypto function. Security proof might assume PN size is constant
  - Not true for proposed ciphersuite, but maybe it applies somwhere in the general case
- Choosing between cipher suite entries would work since we are limited to known good cases
  - Exposes existence of provider bridge to ends performing security negotiation. The end would otherwise have no basis to enter into a negotiation

PN Negotiation Proposal

- When operating across provider bridges
  - The slower device should yield to the faster device PN length and suffer the throughput drop
    - Minimized security risk on the fast side
    - Makes impersonating a provider bridge pointless
- When on a vanilla link
  - Negotiation leads to default for that MAC/PHY

The Alternative

- When operating across provider bridges
  - The faster device should yield to the slower device PN length and suffer the potential increase in rekeying rate
    - Prevents imposing undue overhead on a slow link
- When on a non provider bridge link
  - Negotiation leads to default for that MAC/PHY or a shared enhance mode

PN Negotiation Proposal

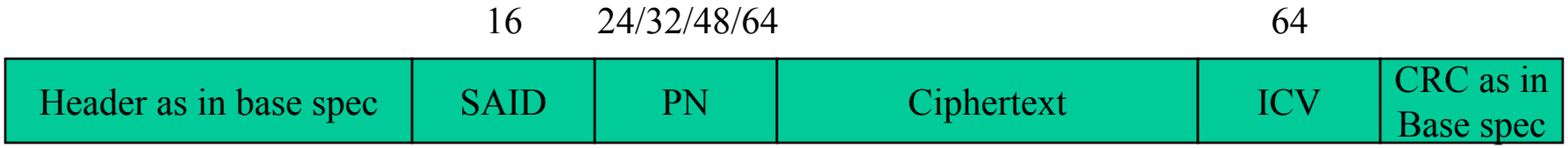
- E.G. CCM Mode cipher suite entry becomes 3…
  - 1:AES-CCM-128, 64 bit ICV, 32 bit PN
  - 2:AES-CCM-128, 64 bit ICV, 48 bit PN
  - 3:AES-CCM-128, 64 bit ICV, 64 bit PN
- 802.15.4 gets (1) as default
- 10G ethernet gets (3) as default
- Negotiation over provider bridge picks highest of 2 defaults within a cipher class (ccm, ocb etc)
- Devices implementing CCM must support all three
  - not a hard implementation issue in this case

SAID Format

- Does it include the cipher type?
  - No, unnecessary, will be set at SA creation
    - Why waste the bandwidth
- How many SAs to support?
  - 16 bits sounds like a lot of SAs
  - Need separate SAs for broadcast/multicast groups
  - Can we trim it to 8 bits?
    - 802.16 supports > 256 SSs.
    - So no – need 16 bits.

Proposed Security Packet Format

- 16 bit SAID
- Variable PN based on ciphersuite negotiation, generally 24 – 64 bits
- 64 bit ICV



Frame Expansion

- Additional 144 bits would exceed HW limitations in certain 802.3 implementations
- Could do MTU limitation
  - Wireless doesn't care. MTU limitation is normal. Frames never expand.
  - Some wired standards care. No means to signal MTU variation upwards
- Could fragment into 2 packets when MTU adjustment not supported, as per 802.10

Frame Expansion Proposals

- Copy what was done in 802.10
  - Contract MTU where standard or implementation supports it
  - Otherwise cause fragmentation into two (sf)MPDUs at security sub layer
- Alternative
  - Fix the MAC service to always make MTU information available at the MAC service, for all MACs supporting linksec.
  - Should have been in there from the start. 802 is more than just Ethernet

AAD

- AAD for different MACs and PHYs can vary
- Some modes need special treatment to support varying AAD
  - E.G. OCB is chained with PMAC in order to support greater AAD than the nonce.
  - CCM does it by default
- How do we deal with AAD? It differs over a provider bridge.

Summary of Proposal

- Adopt normal frame format
  - Header||SAID||PN||CT||ICV||CRC
- 16 bit SAID, 64 bit ICV
- Expose presence of provider bridge crossing to cipher suite negotiators
- Add multiple PN length cipher suite options
- Fix PN length (i.e. the cipher type) for SA lifetime
- Choose whether to
  - Use 802.10 style MTU/Fragment mechanism
  - Force MTU variation (and maybe fix MAC service)