Digital Signature Authentication for ESP/AH

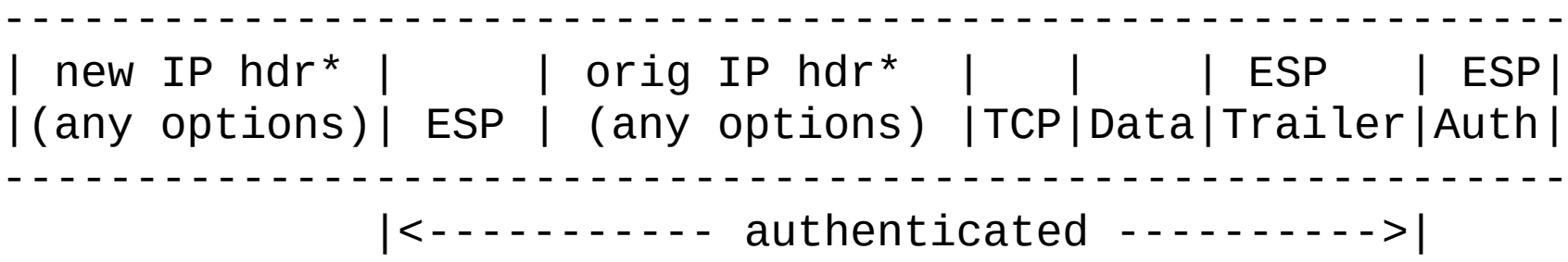
<draft-bew-ipsec-signatures-00.txt>

Brian Weis

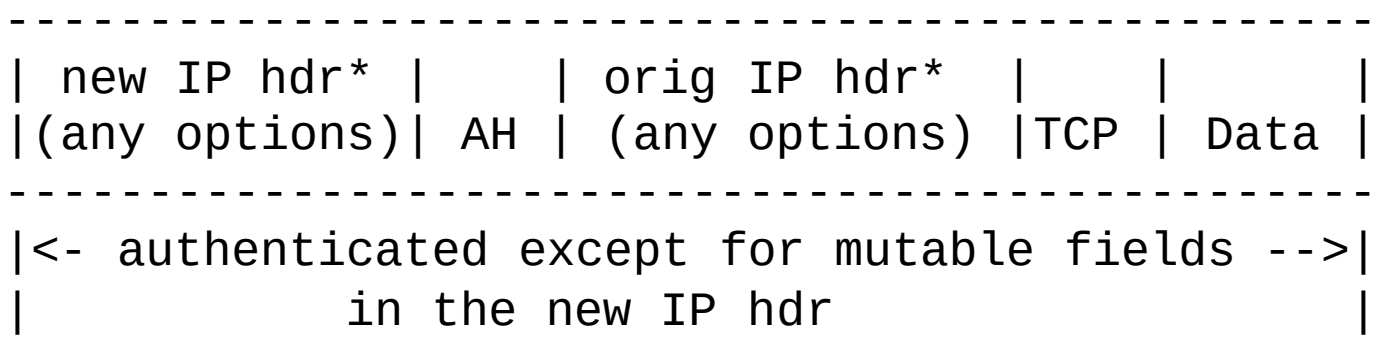
ESP/AH Authentication Overview

- RFC 2406 (ESP) and 2402 (AH) don't specify any particular authentication mechanisms
- RFC 2403 and RFC 2404 define HMAC authentication transforms
  - A key is hashed with the packet
  - The first 12 bytes of the hash are placed in the Authentication Data field

ESP (tunnel mode)



AH (tunnel mode)



Adding Digital Signatures to ESP/AH

- Replacing HMAC with a digital signature is straightforward
  - Take a hash over the ESP or AH authenticated area
  - Encrypt the hash with a private key
  - Put the ciphertext in the Authentication Data field

RSA Algorithm

- Widely implemented
  - Algorithm is freely available
  - RFC 2437
- Relatively fast verification
  - Useful in minimizing processing for a group
- RSA Parameters chosen:
  - RSAES-OAEP raw RSA scheme with default parameters
  - SHA-1 hash algorithm

RSA Modulus Size

- Variable sized modulus
  - Passed by key management
- Minimum size: 496 bits
  - Function of the size of the data (160 bits) and OAEP padding
  - Sounds small, but attacker can only use the key to inject/modify packets. The attacker must find the private key before the session terminates.
- The actual size depends on the application
  - Public keys used for a long period of time should be larger.

RSA Exponent

- The draft does not currently specify the size of the exponent. Does it need to be?
  - The exponent must be passed to receivers along with the modulus. This is a key management issue.
  - Performance issues with a larger exponent?

Key Management support

- Authentication type for ESP/AH
  - The draft proposes a new Authentication Algorithm called SIG-RSA
- Modulus size
  - The draft proposes a new “authentication key size” RFC 2407 attribute

GDOI support

- No changes to ESP SA_TEK
  - Specify the new authentication type in the ESP SA_TEK
  - Send the modulus size sent in the ESP SA_TEK
- The KD payload must pass a TEK_ALGORITHM_KEY attribute with the public key (modulus, exponent)
  - The format of the key will be PKCS#1

SA Lookup Logic

- Each SA must be a single sender SA
- Group Controller (e.g., GDOI) coordinates the SPIs for the SAs.
- SA Lookup still conforms to RFC 2401:

  (Destination address, protocol, SPI)

Issues

- Size of the Authentication Data
  - 61 to 256 bytes of ciphertext
  - Packet fragmentation more likely
- Performance
  - Need to set expectations properly
  - Need some implementation experience with various h/w and s/w implementations

Issues (cont.)

- DoS vulnerability
  - RSA verification is relatively slow in comparison
  - MESP solves the problem by wrapping with an HMAC but that adds complexity and may not always be feasible
  - For example, consider the AH transform used to protect neighbor discovery messages. A long-term public key may be provisioned to the device, and no pairwise session key is possible.

Comparison to MESP

- ESP/AH Signatures is *simpler* than MESP
  - Protocols which specify use of AH or ESP to protect their protocols can take advantage of digital signatures, if appropriate
- ESP/AH Signatures *cannot provide* all of the features of MESP, and may be more vulnerable to a DoS attack.

Questions?