Notes on Network-level Security and IPSec

Antonio Carzaniga

May 15, 2020

1 Outline

- introduction: security at different layers
- basic concepts of internetworking
- architecture of IPSec
- security services: AH, and ESP
- AH format
- AH outbound traffic processing
- AH inbound traffic processing
- ESP format
- ESP outbound traffic processing
- ESP inbound traffic processing
- policy database
- security associations: parameters and setup methods
- Key exchange
  - Diffie-Hellman
  - problems with Diffie-Hellman
  - Oakley
- ISAKMP
  - general principles: generic framework
  - ISAKMP format: generic header
  - generic payload format
  - ISAKMP payloads

- ISAKMP exchanges

- IKE
  - two exchanges: “Main Mode” and “Aggressive Mode”.
  - example: IKE in main mode.

2 Introduction

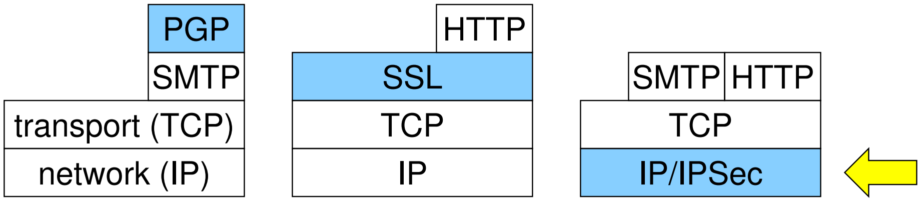
- e-commerce application and secure WEB access
  - one-time and/or short-lived associations
  - essentially one tool/protocol
  - somewhat “mobile” clients (DHCP, modems)

  Solution: TLS/SSL

- professor traveling abroad
  - short- and not-so-short-term associations
  - mobility
  - more than one tool/protocol: e-mail, file access (CVS, rsync), news, printing, etc.
  - mobility on wide-area as well as local-area network

  Soution: secure connection proxy such as SSH

- branches of the same company, or research groups at different universities
  - long-term associations
  - VPNs
  - several applications and protocols (DB, old terminals, experimental distributed systems, etc.)



Advantages of network-level security.

3 Tiny introduction to IP networking

IPv4 header structure

<table>
<tr><td>vers.</td><td>IHL</td><td>TOS</td><td colspan="4">total length</td></tr>
<tr><td colspan="3">fragment identifier</td><td></td><td></td><td></td><td>fragment offset</td></tr>
<tr><td colspan="2">TTL</td><td>protocol</td><td colspan="4">header checksum</td></tr>
<tr><td colspan="7">souce address</td></tr>
<tr><td colspan="7">destination address</td></tr>
<tr><td colspan="7">options</td></tr>
<tr><td colspan="7">padding</td></tr>
</table>

IPv6 header structure

<table>
<tr><td></td><td>vers.</td><td>TOS</td><td colspan="3">flow label</td></tr>
<tr><td></td><td colspan="3">payload length</td><td>next header</td><td>hops rem.</td></tr>
<tr><td>4x</td><td colspan="5">souce address</td></tr>
<tr><td>4x</td><td colspan="5">destination address</td></tr>
<tr><td>opt.</td><td colspan="5">hop−hop header</td></tr>
<tr><td>opt.</td><td colspan="5">source routing header</td></tr>
<tr><td>opt.</td><td colspan="5">fragmentation header</td></tr>
<tr><td>opt.</td><td colspan="5">end−end options</td></tr>
</table>

4 IP-level security overview: goals

IPsec is designed to provide interoperable, high quality, cryptographically-based security for IPv4 and IPv6. In particular:

- access control
- connectionless integrity
- data origin authentication
- protection against replays (a form of partial sequence integrity)
- confidentiality (encryption)
- limited traffic flow confidentiality.

These services are provided at the IP layer, offering protection for IP and/or upper layer protocols.

5 General benefits of IPSec

- network-,transport-, and application-level
  - virtual private networks
  - transparent security for applications and users
- routing infrastructure
  - authentication for routing advertisements

6 How IPSec applies to network traffic

- applications are oblivious to IPSec
- so who decides where, when, and how to apply security?

For each packet, an IPSec implementation decides whether to

- discard that packet
- bypass IPSec security services, or
- afford IPSec security services

In other words, applying IPSec security services is largely a network management decision.

7 Security Policy Database (SPD)

The decision is based on a *security policy*, stored in a Security Policy Database (SPD).
The SPD contains an ordered list of policy entries. Each entry is defined by one or more *selectors*, and an indication whether the traffic matching this policy will be bypassed, discarded, or subject to IPsec processing.

- source address
- destination address
- protocol (IPv4 Protocol or IPv6 Next Header field)
- source port
- destination port
- user id
- …

Selectors may use single values, lists of values, or wild-card expressions.

8 How IPSec provides security

IPsec uses two protocols to provide traffic security:

- *Authentication Header (AH)* provides connectionless integrity, data origin authentication, and an optional anti-replay service.
- *Encapsulating Security Payload (ESP)* provides
  - confidentiality (encryption), and limited traffic flow confidentiality, and optionally
  - data origin authentication, and an anti-replay service

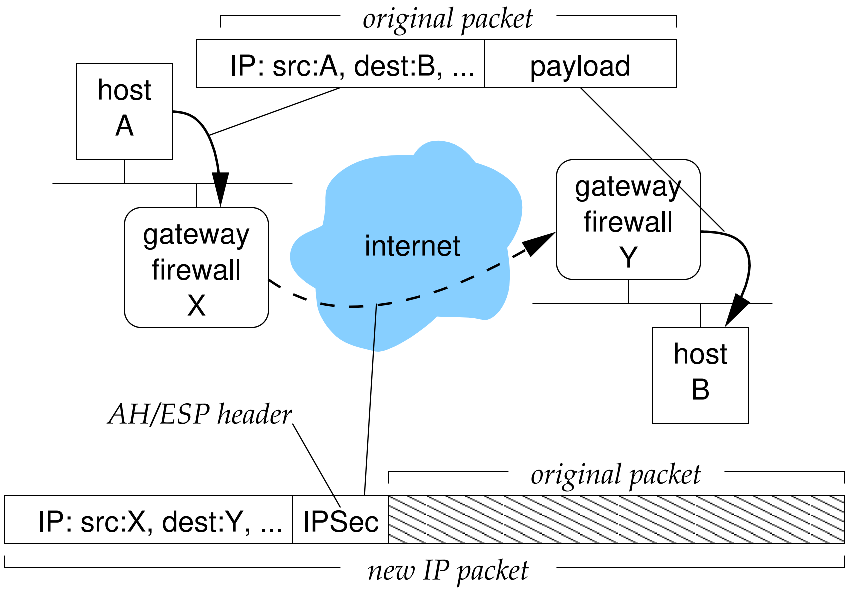
Both AH and ESP, indirectly, provide access control, based on the distribution of cryptographic keys and the management of traffic flows relative to these security protocols.

9 Transport and tunnel modes

Both AH and ESP can function in either

- *transport mode* provides security for upper-level protocols (such as TCP or UDP) by authenticating and/or encrypting the payload
- *tunnel mode* provides security for the whole IP packet by encapsulating (tunneling) that packet into another IP packet.

10 Virtual private networks with tunnel mode



11 Security associations (SA)

- key concept appearing in both authentication and confidentiality mechanisms of IPSec

- *one-way* relation
- between a *sender* and a *receiver*

a security association is identified by

- *security parameter index (SPI)*: a key that identifies a set of parameters for this association, stored in the Security Association Database
- *destination address* (currently only unicast addresses)
- *security protocol identifier*: AH or ESP

12 Authentication Header Format

0	8	16	31
next header	payload len	reserved	
security parameters index (SPI)			
sequence number			
authentication data (variable)			

payload length is the length of this header in 32-bit words, minus 2. (This is a common feature of all IPv6 extension headers.) So, for example, in the case of a 96-bit authentication value (using HMAC-MD5-96 or HMAC-SHA1-96), the value of *payload length* would be 4.

reserved a 16-bit field reserved for future use. It must be set to 0. Note that this value is included in the scope of the authentication function.

13 AH outbound processing

1. *security association lookup*
2. *sequence number generation*
   - sequence number is initialized to 0 when the SA is establised
   - if anti-replay service enabled: sequence number is incremented for each packet
   - if anti-replay service enabled: sender checks for sequence number overflow. Overflow is not permitted, so a new SA must be established.
3. *integrity code calculation*

   Authentication data contains the MAC for the packet. It is a *variable-length* field. The current specification mentions two MAC algorithms:

- HMAC-MD5-96
- HMAC-SHA-1-96

The value of *authentication data* (MAC) is computed over

- immutable IP header fields, for example:
  - IPv4 Internet Header Length
  - IPv6 Version
  - Source Address
- predictable IP header fields, for example:
  - destination address
- AH header (other than Authentication Data field)
- the entire payload (which is immutable in transit)

Mutable fields are zeroed when computing the MAC.

For example, this is the classification of header fields for the IPv6 (base) header:

- *Immutable*
  - Version
  - Payload Length
  - Next Header (This should be the value for AH.)
  - Source Address
  - Destination Address (without Routing Extension Header)
- *Mutable but predictable*
  - Destination Address (with Routing Extension Header)
- Mutable (zeroed prior to ICV calculation)
  - Class
  - Flow Label
  - Hop Limit

It might be necessary to add some *padding* to the data in order to compute the integrity code. Necessary padding must be added at the end of the packet, it must be *zero*, and is not transmitted.

4. *fragmentation*

   if necessary, fragmentation occurs only *after* AH processing.

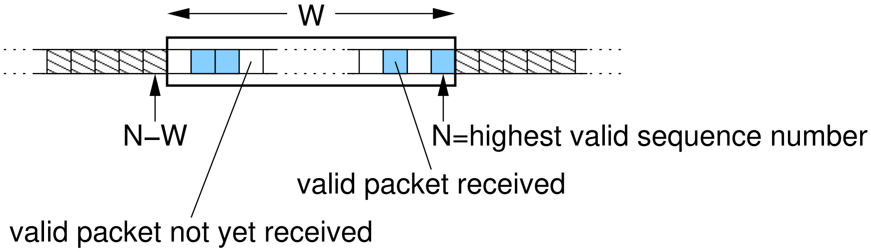
14 Inbound traffic processing

1. *reassembly*: fragments are reassembled as ususal

2. *security association lookup*: when the destination receives an IP packet with an AH in it, it looks up the SA database using (IP.dest, SPI, and AH) as a key. If no SA is found, the destination must drop the packet. If an SA is found, the destination checks whether anti-replay protection is enabled. If anti-replay protection is enabled, the destination performs a sequence number verification, otherwise it proceeds with the integrity check.

3. *sequence number verification*: the Sequence Number field allows some protection against replay attacks. If this feature is enabled, the sender must make sure that Sequence Number is initialized to 0 and incremented for each packet sent over the same SA.

   IP is unreliable, therefore the receiver must keep track of what sequence number it has seen. The specification mandates a window of fixed size W (at least $W = 32$, default $W = 64$)

   
   W
N−W
N=highest valid sequence number
valid packet received
valid packet not yet received

   Only packets that are either inside the window and new, or to the right of the window edge, will proceed to the integrity check. The right edge of the window is shifted to the highest integrity-verified sequence number received.

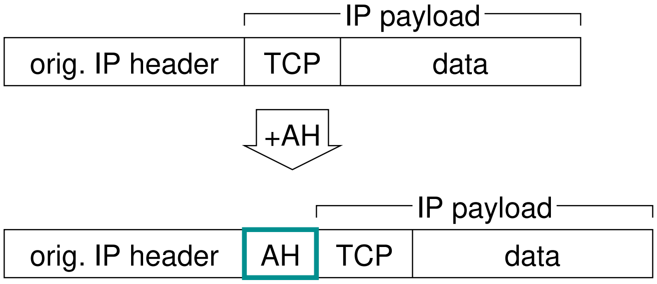
4. *integrity check*:

   (a) save the ICV
   (b) replace the bytes with zeroes
   (c) zero out all other mutable fields
   (d) add (zero) padding if necessary
   (e) compute the MAC and compare with saved ICV

   packets that fail the integrity verification must be discarded

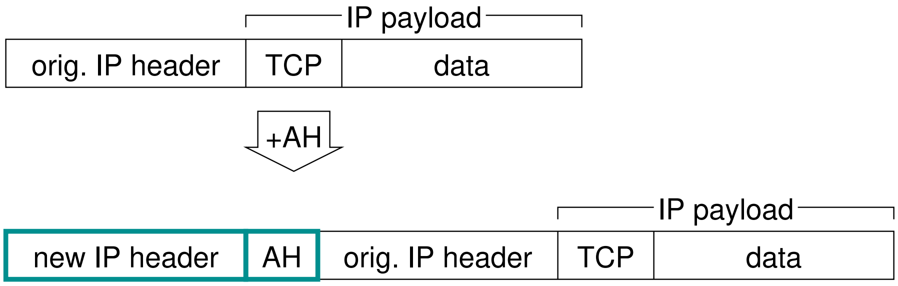
15 Transport mode for AH

For *transport mode AH*, the authentication header is inserted after the original IP header and before the IP payload (with minor differences between IPv4 and IPv6):

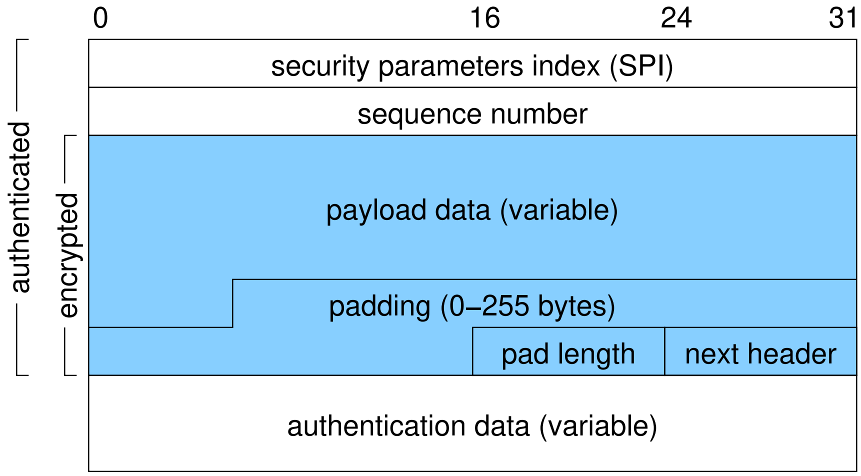


16 Tunnel mode for AH

For *tunnel mode AH*, the original IP packet is encapsulated within another IP packet. The authentication header authenticates the entire original IP packet, and is inserted between the outer IP header and the inner (original) IP header.



17 Encapsulating Security Payload (ESP)



- confidentiality of payload
- integrity (⇔ authentication)
- protection against replay attacks

18 Authentication and reply attack protection in ESP

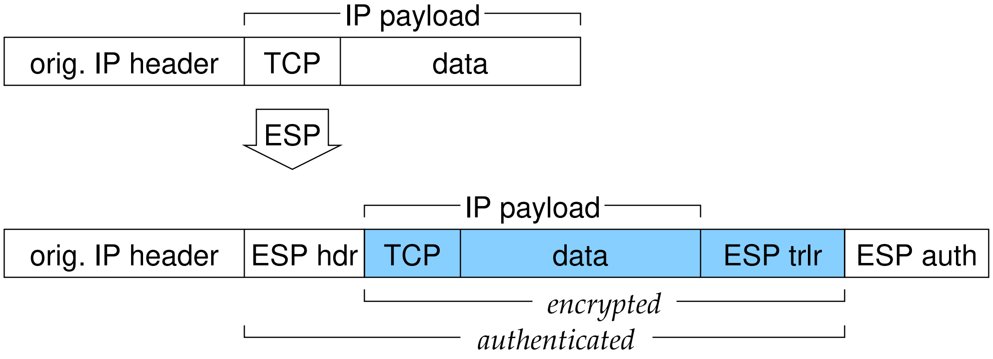
- *protection against replay attack*: same as in AH
  - sender manages Sequence Number
  - receiver implements sliding window
- *integrity and authentication*: computed over ESP header (SPI and Sequence Number)

19 Confidentiality in ESP

- covers payload
- padding
  - match plaintext length required by encryption algorithm
  - additional confidentiality, by hiding the actual payload length
- *encryption algorithm*: implementations must support DES in CBC
- other algorithms have been identified:
  - 3DES
  - RC5
  - IDEA
  - 3IDEA
  - CAST
  - Blowfish

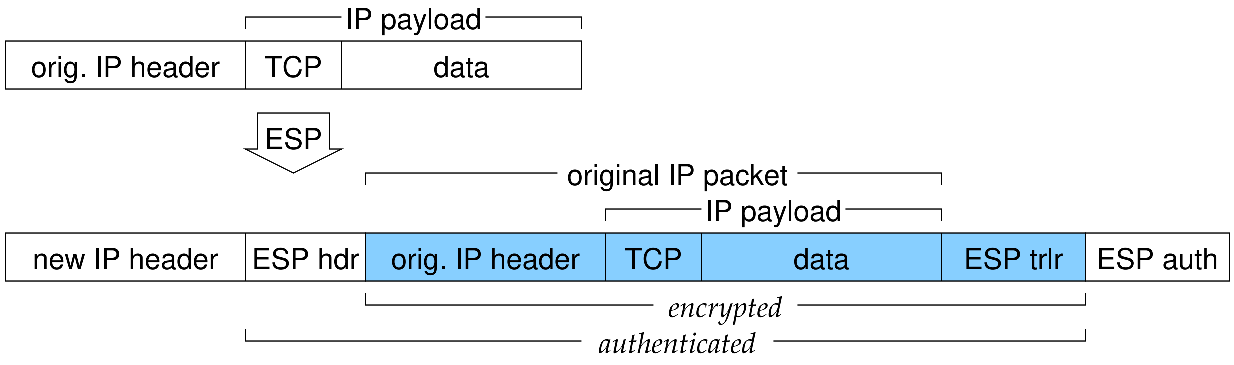
20 Transport mode for ESP

Transport mode ESP can be seen as a simple IP payload: the ESP packet is inserted after the original IP header (with some minor differences between IPv4 and IPv6):



21 Tunnel mode for ESP

For *tunnel mode ESP*, the original IP packet is encapsulated within another IP packet as an encrypted and (optionally) authenticated ESP:



22 Security associations parameters

Here's what a SA stores:

- *sequence number counter*: 32-bit value used to generate sequence numbers in AH and ESP headers
- *sequence counter overflow*: flag indicating whether an overflow of the sequence number counter should produce an auditable event
- *anti-replay window*
- *AH information*: authentication algorithm, keys, key lifetime, other AH-related parameters
- *ESP information*: encryption and authentication algorithms, keys (encryption, and auth.), key lifetimes, initialization values, and other ESP-related parameters
- *lifetime of the SA* time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated
- *IPSec protocol mode*
- *path MTU*

23 Some comments

- key exchange and cipher negotiation is outside the scope of the basic architecture
- policy definition is outside of the scope of IPSec
- policy implementation is essentially a network management issue
- tunneling modes, especially in ESP with authentication allows you to set up virtual private networks
- transport mode provides a security service comparable to transport-level security

24 Security Association

A security association is identified (in processing an inbound datagram) by:

- *security parameter index (SPI)*: a key that identifies a set of parameters for this association, stored in the Security Association Database
- *destination address* (currently only unicast addresses)
- *security protocol identifier*: AH or ESP

25 Security associations and security services

A security association (SA) is a set of policy and key(s) used to protect information. The ISAKMP SA is the shared policy and key(s) used by the negotiating peers in this protocol to protect their communication.
The general IPSec architecture specifies two classes of SA and key management:

- *manual*: ad-hoc, network administrators directly configure the security databases and the policy databases.
- *automated*: IKE: using Oakley, an authenticated key-exchange protocol
- IPSec protocols (AH and ESP) are mostly independent of the way SAs are set up and maintained
- however, the ultimate level of security of an IPSec association is almost completely dependent on the process by which the SA is setup and managed.
- also, some specific features (for example, the anti-replay counter feature) require an automated management procedure.

26 Manual vs Automated SA and key management

Obvious advantages and disadvantages of manual management

+ ad hoc: easier to implement, at least initially

− not scalable

− unable to support some IPSec features (e.g., anti-replay in both AH and ESP)

and of automated management:

+ scalable to large networks, and across administrative boundaries

− requires PKI for complete authentication

27 Automated SA and key management

The default automated key management protocol for IPSec is called ISAKMP/Oakley.

ISAKMP Internet Security Association and Key Management Protocol provides a framework for authentication and key exchange. ISAKMP is designed to be key exchange independent; that is, it is designed to support many different key exchange protocols.

Oakley is a specific key-exchange protocol defined as the default key-exchange protocol in ISAKMP.

28 Overview of ISAKMP

Phase 1 is where the two ISAKMP peers establish a secure, authenticated channel with which to communicate. This is called the ISAKMP Security Association (not to be confused with the SAs that the protocol is trying to establish).

Phase 2 is where Security Associations are negotiated on behalf of services such as IPsec or any other service which needs key material and/or parameter negotiation

29 Overview of Oakley

Oakley is a refinement of Diffie-Hellman. Diffie-Hellman has the following problems that Oakley solves:

- no authentication
- vulnerable to man-in-the-middle attack
- computationally intensive, therefore vulnerable to DOS attack

Oakley uses the following features to counter these weaknesses:

- *cookies* to counter clogging attacks
- *groups* global parameters of the Diffie-Hellman key exchange (p and α). Oakley allows parties to negotiate groups.
- *nonces* to protect against replay attacks
- *authenticated Diffie-Hellman* to protect against man-in-the-middle attack. Authentication is based on either:
  - digital signature (with public/private keys)
  - public-key encryption
  - symmetric-key encryption

30 ISAKMP

ISAKMP defines procedures and packet format (i.e., a protocol) to establish, negotiate, modify, and delete security associations.
In essence, ISAKMP defines a header.

<table>
<tr><td>2x</td><td colspan="5">initiator cookie</td></tr>
<tr><td>2x</td><td colspan="5">responder cookie</td></tr>
<tr><td></td><td>next payload</td><td>maj.v.</td><td>min.v.</td><td>exchng type</td><td>flags</td></tr>
<tr><td></td><td colspan="5">message id</td></tr>
<tr><td></td><td colspan="5">length</td></tr>
</table>

This header introduces the following types of payloads:

proposal payload used for SA negotiation

- SA protocol (AH or ESP)
- sender's SPI
- a list of *transform payloads*

transform payload identifies a cryptographic function for a specific protocol function (e.g., 3DES for ESP), and its parameters.

key exchange payload Oakley, or Diffie-Hellman, or RSA-based key exchange, etc.

identification payload identifies the two ISAKMP peers

certificate payload passes a public-key certificate

hash payload verification (authentication) hash computed over some state information of this running ISAKMP

nonce payload random data used during the exchange

notification payload either error or status information associated with the current SA or with the SA negotiation

delete payload identifies one or more SAs that the sender has deleted from its database