IP Security

Ibéria Medeiros

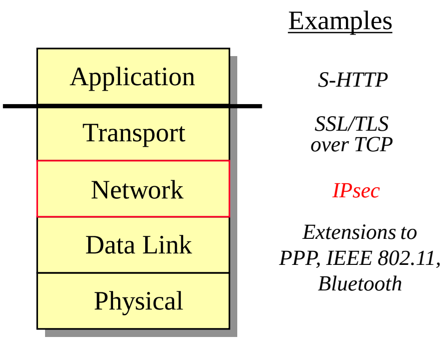
Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa

Secure Inter-Domain Routing Standards Aim to Help Secure Border Gateway Protocol (Oct. 2017)

The US National Institute of Standards and Technology's (NIST) National Cybersecurity Center of Excellence (NCCoE) and the Department of Homeland Security's (DHS) Science and Technology Directorate have worked together to develop a set of standards to help **secure the Border Gateway Protocol (BGP)**, which is used to route Internet traffic between various providers and networks. Known as **Secure Inter-Domain Routing (SIDR)**, the people working on the project have begun posting some of the developed standards to the Internet Engineering Task Force (IETF) portal.

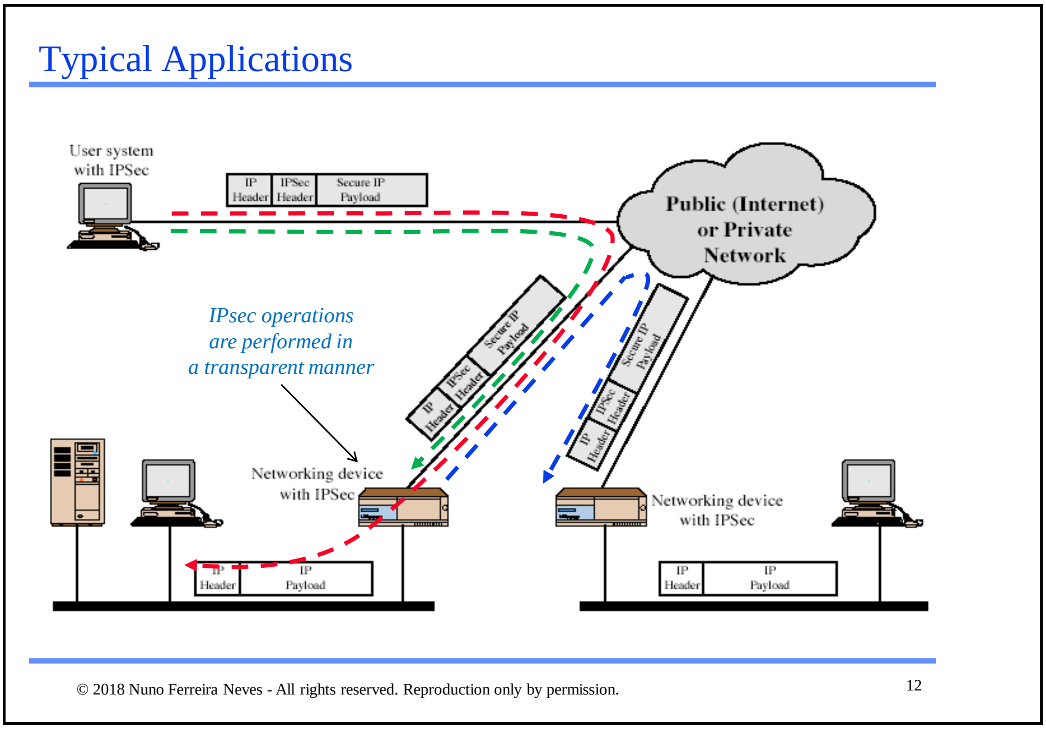
Secure Communication

- A message goes through several layers from the moment it leaves the application and until it reaches the network
- Each layer typically implements a set of tasks to ensure certain properties (e.g., FIFO)
- Communication can be protected at the various layers, affording many options to secure the final application

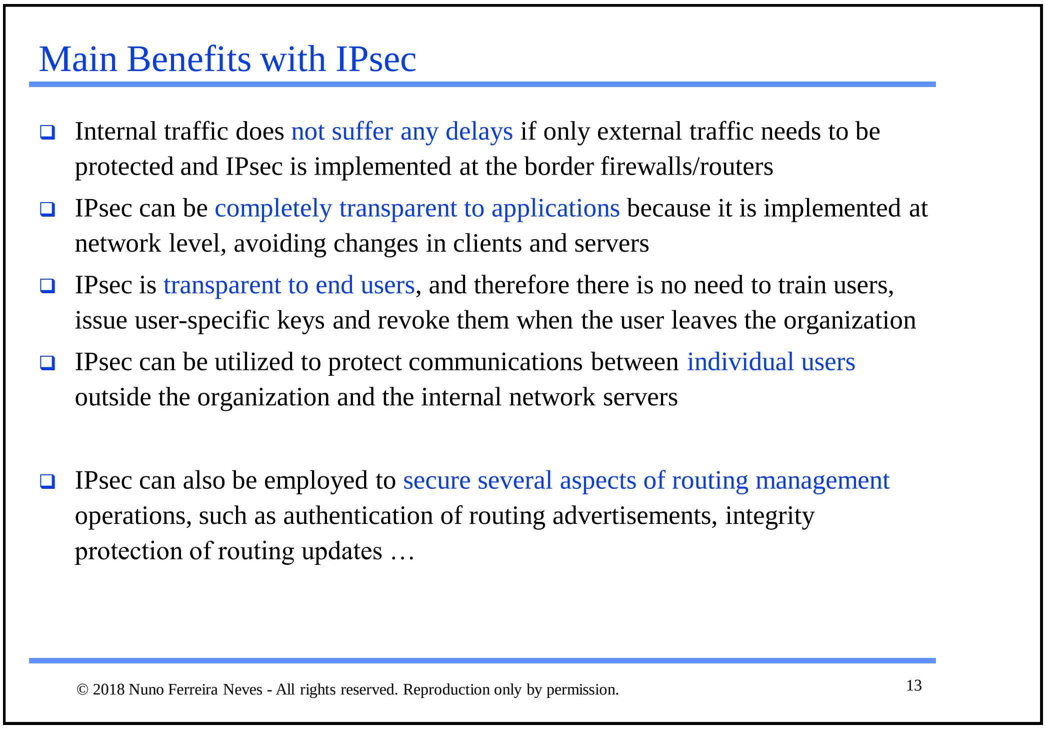


IPsec Overview

- Throughout the years there have been several attacks to the IP protocol, some of them extremely simple to perform like IP spoofing and sniffing of packets
- Many times there is the need for the creation of secure channels to protect all traffic between two machines, instead of relying on application specific solutions
- IPsec provides secure communication (integrity, confidentiality, and authenticity) in a LAN, in a private or public WAN, or in an Internet environment
- Security mechanisms included in IPsec were chosen to support both IP v4 and v6



Typical Applications



Main Benefits with IPsec

- Internal traffic does not suffer any delays if only external traffic needs to be protected and IPsec is implemented at the border firewalls/routers
- IPsec can be completely transparent to applications because it is implemented at network level, avoiding changes in clients and servers
- IPsec is transparent to end users, and therefore there is no need to train users, issue user-specific keys and revoke them when the user leaves the organization
- IPsec can be utilized to protect communications between individual users outside the organization and the internal network servers

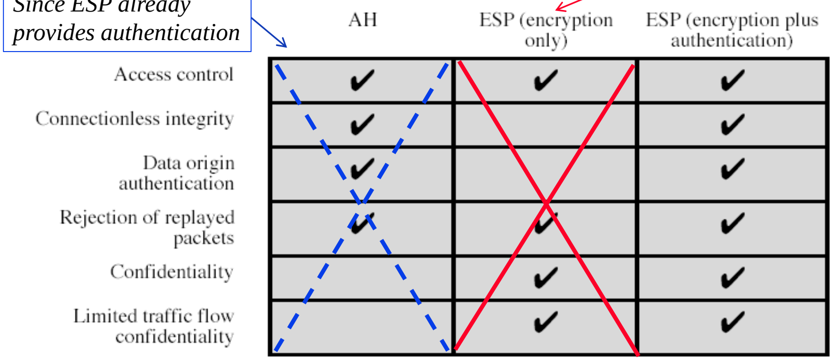
- IPsec can also be employed to secure several aspects of routing management operations, such as authentication of routing advertisements, integrity protection of routing updates …

IPsec Services

- ❑ Security mechanisms are implemented through extension headers
  - *ESP: Encapsulating Security Payload Header*
  - *AH: Authentication Header*

Should not be used because it can create security problems

Since ESP already provides authentication



	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✓	✓	✓
Connectionless integrity	✓		✓
Data origin authentication	✓		✓
Rejection of replayed packets	✓	✓	✓
Confidentiality		✓	✓
Limited traffic flow confidentiality		✓	✓

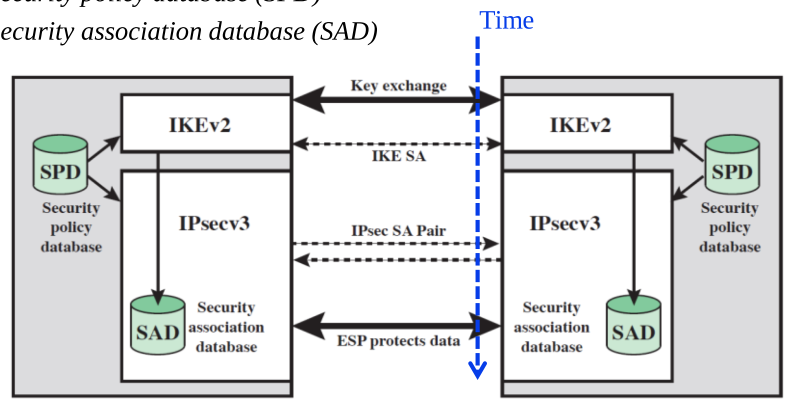
Transport and Tunnel Modes

- ❑ ***Transport***: protects data from protocols above IP
  - typically used to protect communications between final users
  - AH : authenticates IP payload + selected parts of IP header (and extensions)
  - ESP : encrypts IP payload + selected extension headers
  - ESP with auth : *similar to ESP* + authenticates IP payload (but **not** the header)

- ❑ ***Tunnel***: protects the whole IP packet (an IP packet is placed inside another)
  - typically used in communications between firewalls
  - outer header will have **different IP addresses** than the inner header
  - AH : authenticates inner IP packet + selected parts of outer IP header (and extensions)
  - ESP : encrypts inner IP packet
  - ESP with auth : *similar to ESP* + authenticates inner IP packet

Security Policy

- A *security policy* is applied to each packet transmitted from a given source to a destination
- The policy is determined by two databases
  - *security policy database (SPD)*
  - *security association database (SAD)*



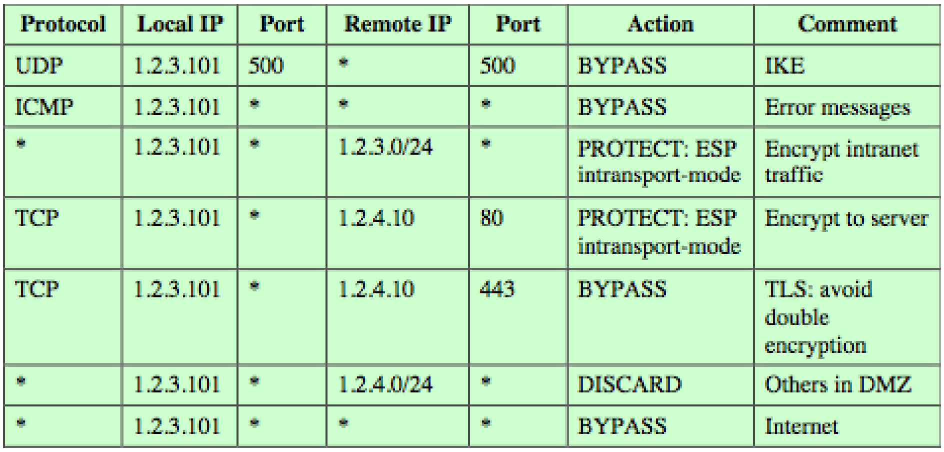
Security Associations

- A ***secure association (SA)*** is a one-way relationship between a sender and a receiver for secure data transmission between the two peers
  - bidirectional traffic requires the creation of **two** SA
  - a SA allows the use of AH or ESP, but **not both**
- A SA is identified using three components
  - security parameter index (SPI) : 32-bits with local meaning to receiver
  - destination IP address: just for unicast addresses
  - security protocol identifier: AH or ESP identifier
- The **SA database (SAD)** stores for every SA certain parameters, such as
  - sequence number counter and a flag for what to do when max value is reached
  - encryption/authentication algorithms, keys, key lifetimes, ...
  - SA lifetime or maximum number of bytes that can be transmitted
  - protocol mode, which is either transport, tunnel, or wildcard

Security Policy Database (SPD)

- Whenever a packet arrives from one application, IPsec needs
  - to determine if a security service needs to be applied to the packet, **and**
  - in the affirmative case, what service(s) has to be employed
- The SPD is configured by the user to indicate for each subset of IP traffic what SA or SAs are used
- ***Selectors*** are filters for outgoing packets in order to map them into a particular SA, and they are based on a set of upper-layer protocol field values
  - destination IP address (single or list of addresses, range of addresses, …)
  - source IP address (e.g., a firewall receives packets from various sources but only for some of them it needs to utilize IPsec)
  - transport protocol
  - user name from the operating system
  - source and destination ports

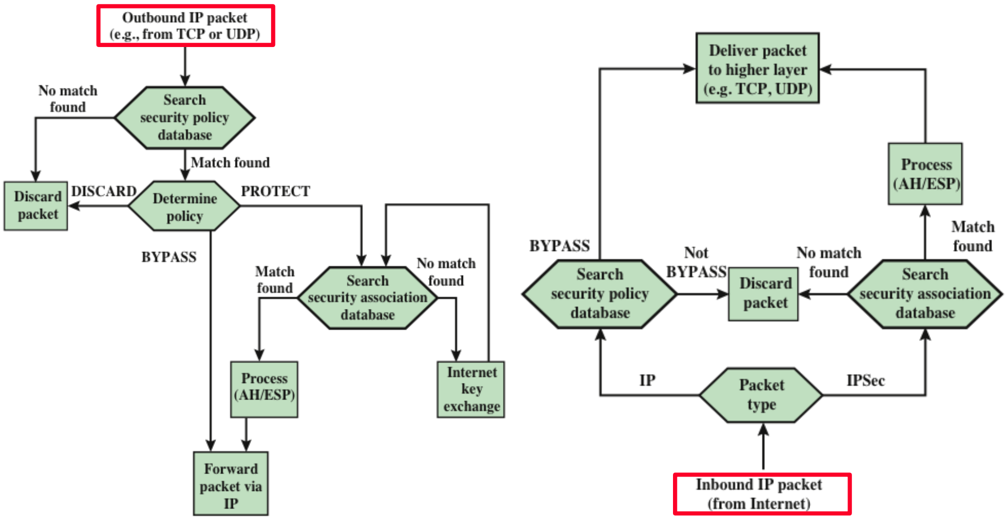
Example SPD for a end host system



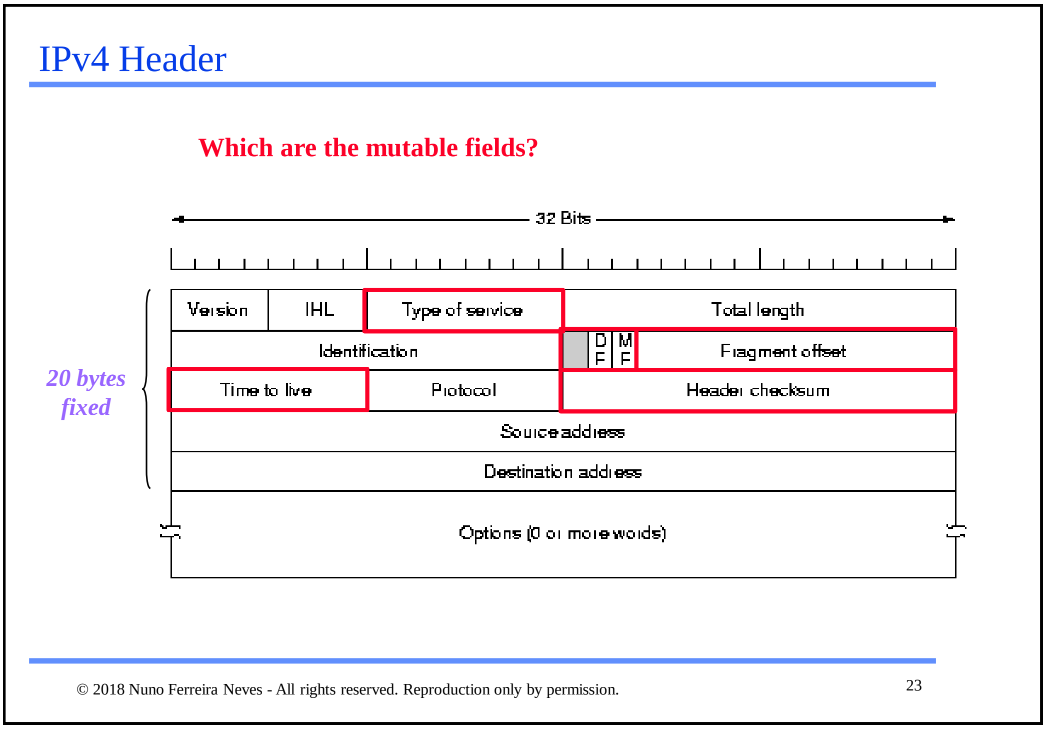
Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

NOTE: port 500 corresponds to IKE;
1.2.3.0/24 corporate network and 1.2.4.0/24 DMZ network

Overview of Traffic Processing


Outbound IP packet (e.g., from TCP or UDP)
No match found
Search security policy database
Match found
Discard packet
DISCARD
Determine policy
PROTECT
BYPASS
Match found
Search security association database
No match found
Process (AH/ESP)
Internet key exchange
Forward packet via IP
Deliver packet to higher layer (e.g. TCP, UDP)
Process (AH/ESP)
Match found
BYPASS
Not BYPASS
No match found
Search security policy database
Discard packet
Search security association database
IP
Packet type
IPSec
Inbound IP packet (from Internet)

IPSEC
IN MORE DETAIL



IPv4 Header

Which are the mutable fields?

32 Bits

Version | IHL | Type of service | Total length

Identification | D F | M F | Fragment offset

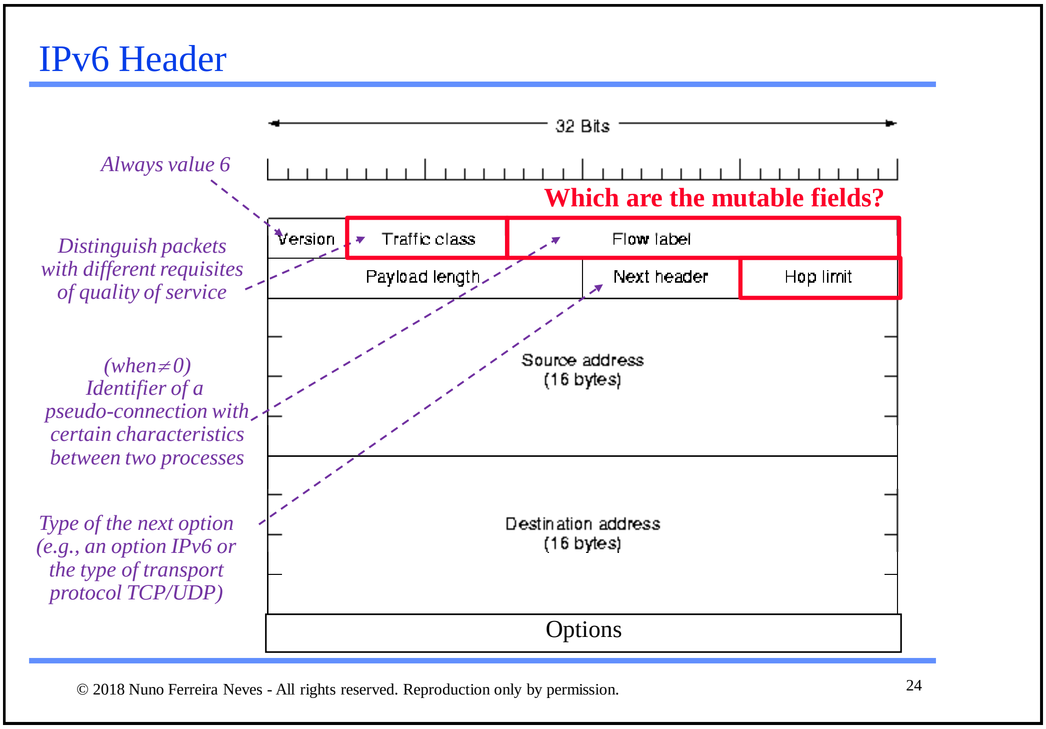
Time to live | Protocol | Header checksum

Source address

Destination address

Options (0 or more words)

20 bytes fixed



IPv6 Header

32 Bits

Which are the mutable fields?

Version | Traffic class | Flow label

Payload length | Next header | Hop limit

Source address (16 bytes)

Destination address (16 bytes)

Options

Always value 6

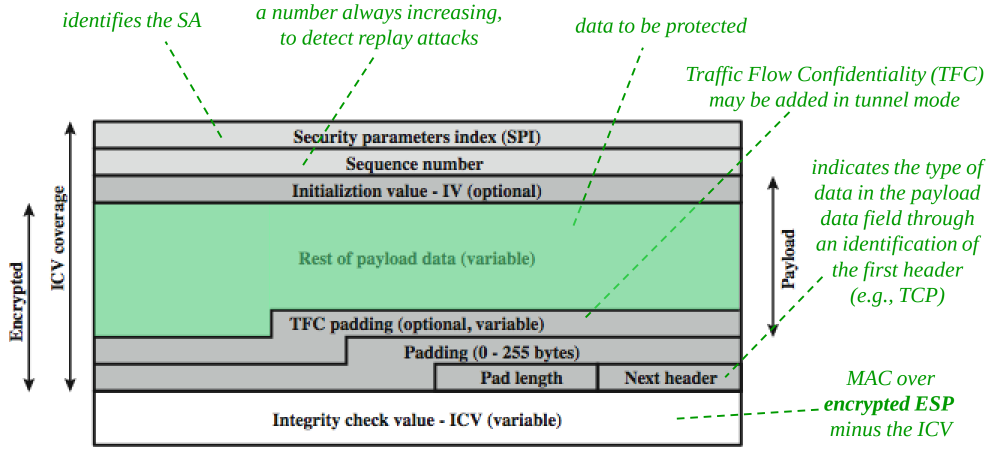
Distinguish packets with different requisites of quality of service

(when≠0) Identifier of a pseudo-connection with certain characteristics between two processes

Type of the next option (e.g., an option IPv6 or the type of transport protocol TCP/UDP)

ESP Header

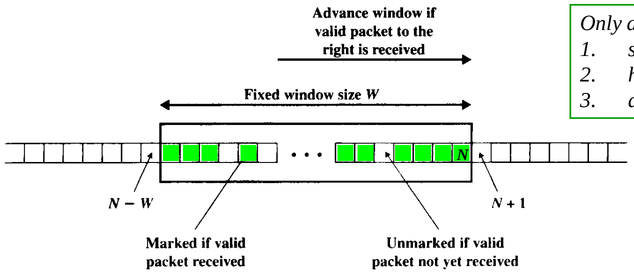
Based on Symmetric Cryptography



identifies the SA
a number always increasing, to detect replay attacks
data to be protected
Traffic Flow Confidentiality (TFC) may be added in tunnel mode
indicates the type of data in the payload data field through an identification of the first header (e.g., TCP)
MAC over ***encrypted ESP*** *minus the ICV*
Security parameters index (SPI)
Sequence number
Initializtion value - IV (optional)
Rest of payload data (variable)
TFC padding (optional, variable)
Padding (0 - 255 bytes)
Pad length
Next header
Integrity check value - ICV (variable)
Encrypted
ICV coverage
Payload

Detection of Replay Attacks

- **REMINDER:** IP does not ensure reliable and ordered packet delivery !!!
- Sender
  - when the SA is created, the sequence counter is initiated with 0
  - for each new packet, the counter is incremented by 1, and this value is stored in the header
  - the SA should be re-initialized when the counter reaches 2^{32}-1
- Receiver
  - can receive out of order packets or miss a few packets



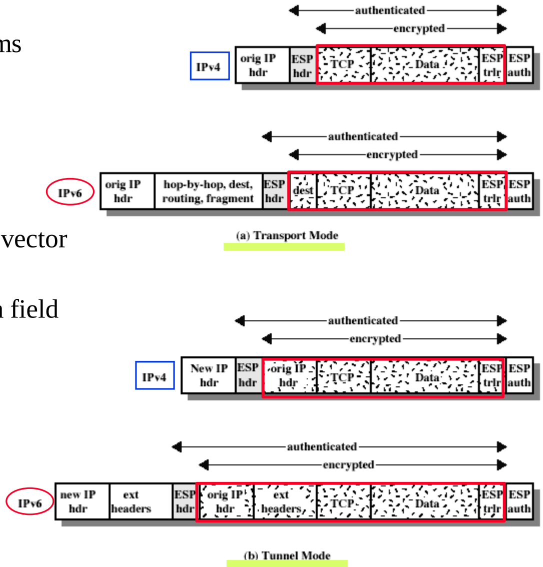
Advance window if valid packet to the right is received
Fixed window size *W*
N
N − W
N + 1
Marked if valid packet received
Unmarked if valid packet not yet received

Only accept packets if:

1. *sequence number > N - W*
2. *has not been received yet*
3. *authentication is valid*

ESP Protection

- ❑ Example encryption algorithms
  - AES
  - three-key triple DES
  - DES-CBC
- ❑ If necessary, an initialization vector is placed unencrypted at the beginning of the payload data field
- ❑ Authentication is done after encryption

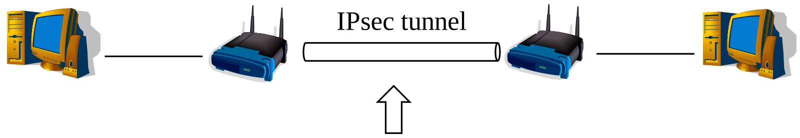


(a) Transport Mode

(b) Tunnel Mode

Attacking ESP in Tunnel Mode (**no** authentication)

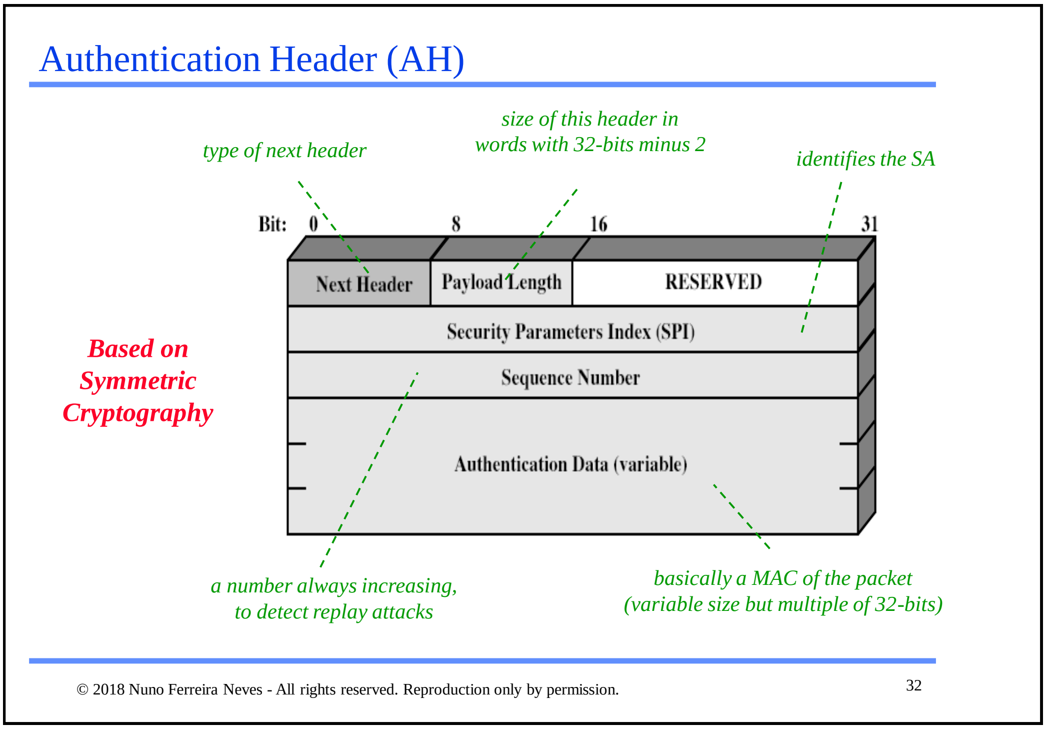
- ❑ Objective: decrypt a message by decrypting one byte at a time



Inject traffic with the necessary bit flips, and look at the generated ICMP error traffic to determine if the message was accepted, and therefore, be eventually able to guess Ci

The attack depends in a detailed way on the structure of the headers of IP datagrams, on the order in which the fields of inner IP datagram headers are processed by an implementation of IP after IPsec processing has completed, and on the way in which ICMP errors are generated in the event of certain errors in processing.

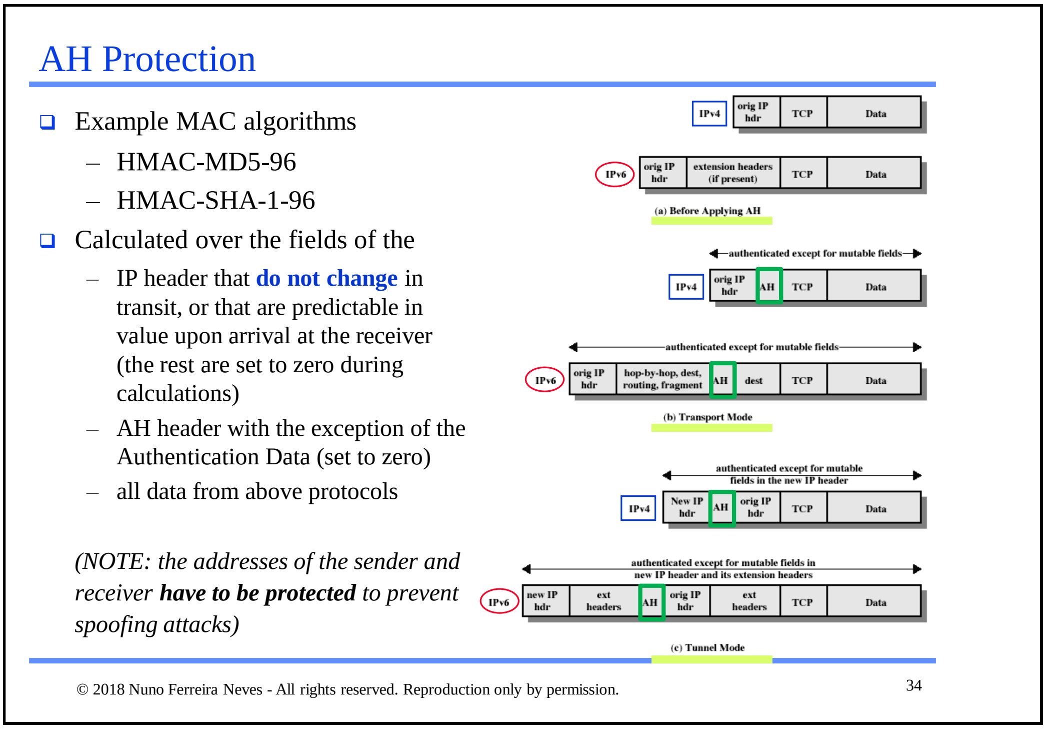
Details in: J. Degabriele, K. Paterson, *Attacking the IPsec Standards in **Encryption-only** Configurations*, SP 2007



Authentication Header (AH)

Based on Symmetric Cryptography

type of next header
size of this header in words with 32-bits minus 2
identifies the SA
Bit: 0
8
16
31
Next Header
Payload Length
RESERVED
Security Parameters Index (SPI)
Sequence Number
Authentication Data (variable)
a number always increasing, to detect replay attacks
basically a MAC of the packet (variable size but multiple of 32-bits)



AH Protection

- Example MAC algorithms
  - HMAC-MD5-96
  - HMAC-SHA-1-96
- Calculated over the fields of the
  - IP header that **do not change** in transit, or that are predictable in value upon arrival at the receiver (the rest are set to zero during calculations)
  - AH header with the exception of the Authentication Data (set to zero)
  - all data from above protocols

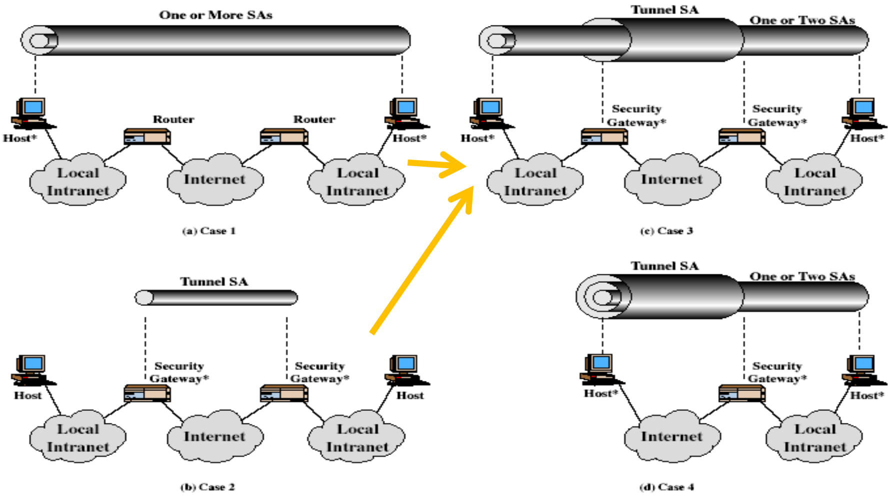
*(NOTE: the addresses of the sender and receiver **have to be protected** to prevent spoofing attacks)*

IPv4
orig IP hdr
TCP
Data
IPv6
orig IP hdr
extension headers (if present)
TCP
Data
(a) Before Applying AH
authenticated except for mutable fields
IPv4
orig IP hdr
AH
TCP
Data
authenticated except for mutable fields
IPv6
orig IP hdr
hop-by-hop, dest, routing, fragment
AH
dest
TCP
Data
(b) Transport Mode
authenticated except for mutable fields in the new IP header
IPv4
New IP hdr
AH
orig IP hdr
TCP
Data
authenticated except for mutable fields in new IP header and its extension headers
IPv6
new IP hdr
ext headers
AH
orig IP hdr
ext headers
TCP
Data
(c) Tunnel Mode

Combining SA

- A *security association bundle* refers to a sequence of SA through which traffic must be processed, and they can end in the same endpoints or different ones
- Basic ways to combine SA
  - ***transport adjacency*** : apply more than one SA without using tunnels (typically a AH and a ESP)
  - ***iterated tunneling*** : apply more than one tunnel possibly with different characteristics
- Example: How can we obtain *authentication plus confidentiality*?
  1. ESP with authentication: (MAC applied over the encrypted data)
     - transport ESP : does not protect the IP header
     - tunnel ESP : protects until the end of the tunnel (which might be a firewall and not the end destination)
  2. transport adjacency : ESP followed by AH, to protect also some fields of IP header
  3. transport & tunnel: transport AH followed by ESP tunnel, allows authentication information to be together with the clear data

Example Combinations of SA that MUST be Supported



(a) Case 1

(c) Case 3

(b) Case 2

(d) Case 4

Key Management

- ❑ Involves the authentication and distribution of secret keys
- ❑ It is typically necessary 4 keys between each pair of nodes

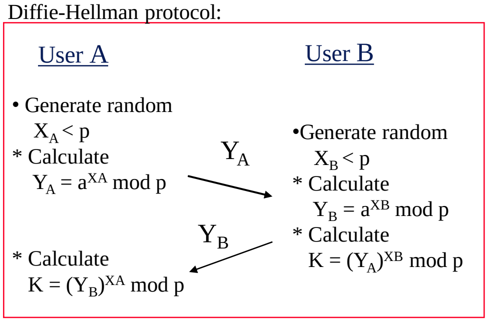
- ❑ Types of key management
  - – manual: the administrator sets up the machines with the necessary keys
  - – automated: the systems support dynamic key creation and distribution

- ❑ IKEv2 is the default automated key management protocol, which includes
  - – *Key exchange & authentication method* based on the Diffie-Hellman
  - – Framework for *formatting the messages* necessary for key management and negotiation of security attributes

Key Determination Protocol

- ❑ Problems with Diffie-Hellman (DH)
  - – does not provide the peers identities
  - – vulnerable to the man-in-the-middle (MIM) attack
  - – computationally intensive, which makes it vulnerable to DoS

Diffie-Hellman protocol:



User A

- Generate random $X_A < p$
- * Calculate $Y_A = a^{XA} \bmod p$
- * Calculate $K = (Y_B)^{XA} \bmod p$

User B

- •Generate random $X_B < p$
- * Calculate $Y_B = a^{XB} \bmod p$
- * Calculate $K = (Y_A)^{XB} \bmod p$

- ❑ Advantages of the IKE protocol
  1. uses *cookies* to thwart DoS attacks
  2. allows the negotiation of DH global parameters (called a ***group (p,a)***)
  3. supports the exchange of DH public parameters
  4. uses *nonces* to be protected from replay attacks
  5. authenticates messages to prevent MIM attacks

How do we get these “advantages”?

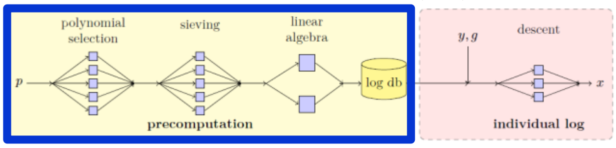
- ❑ The adversary spoofs an address of a valid user, and sends a message to a server containing some random DH parameters => server will spend some amount of CPU to calculate the secret key => ***DoS on the server***
  - – the *cookie* mechanism limits this form of attack by requiring
    - » each peer sends a pseudo-random number, the *cookie*, in hers first message (e.g., hash(IP_s, Port_s, IP_r, Port_r, local_secret)) which later needs to be confirmed
    - » the secret key is only calculated after the reception of her cookie in the first DH message
- ❑ MIM attacks are prevented from occurring by using one of the authentications
  - – digital signatures over an hash of the relevant fields of the messages
  - – encryption with the private key of the relevant fields of the messages
  - – symmetric encryption with a pre-set key of the relevant fields of the message

Attacking IPsec Cripto

- ❑ IKE defines a few sets of groups that should be used by IPsec implementations
  - – example for DH-1024: a = 2; p = $2^{1024} - 2^{960} - 1 + 2^{64}(\lfloor 2^{894} \times \pi \rfloor + 129093)$
- ❑ This makes it interesting to attack the specifically supported cripto groups by **performing pre-computations**!

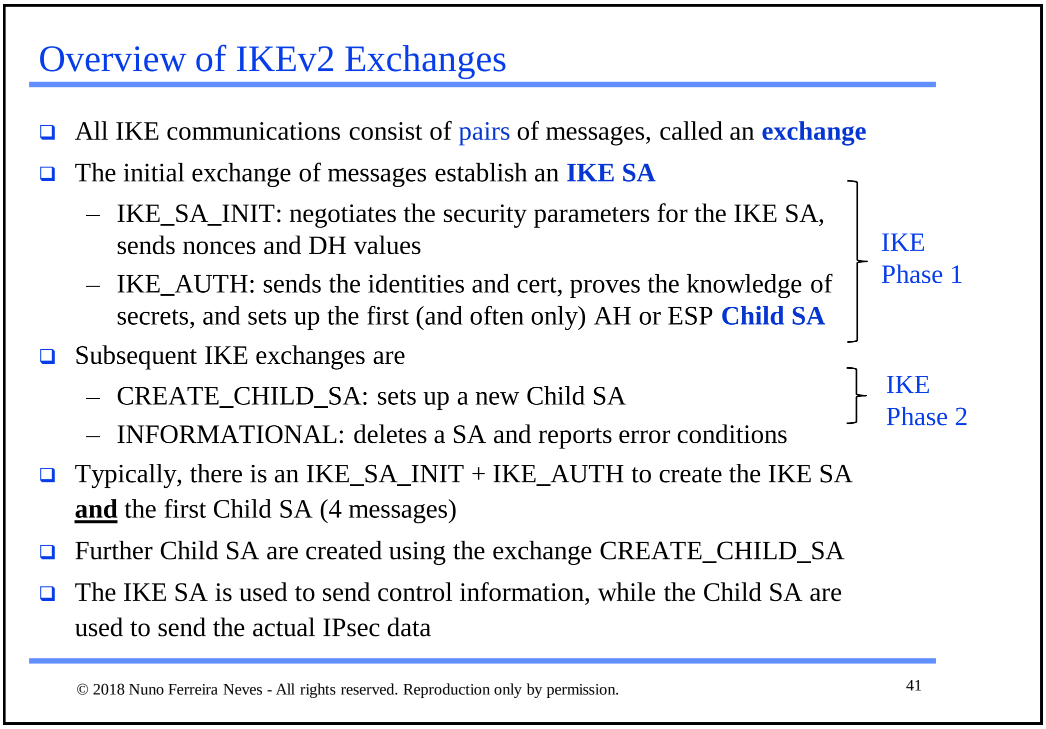
Pre-computation can be performed a priori before the actual calculation of the private DH parameter is needed

Number Field Sieve (NFS)



- ❑ It is within the capacity of the NSA resources to perform the pre-computation of the NFS for a **few** of the **standardized DH groups**
- ❑ The effort for the attack is in the order of **45M core-years**, but with specialized hardware (ASICS), it should be possible to do the pre-computation for DH-1024 in one year with a machine costing a **few hundred of millions** of dollars

Details: D. Adrian et al., *Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice*, CCS 2015



Overview of IKEv2 Exchanges

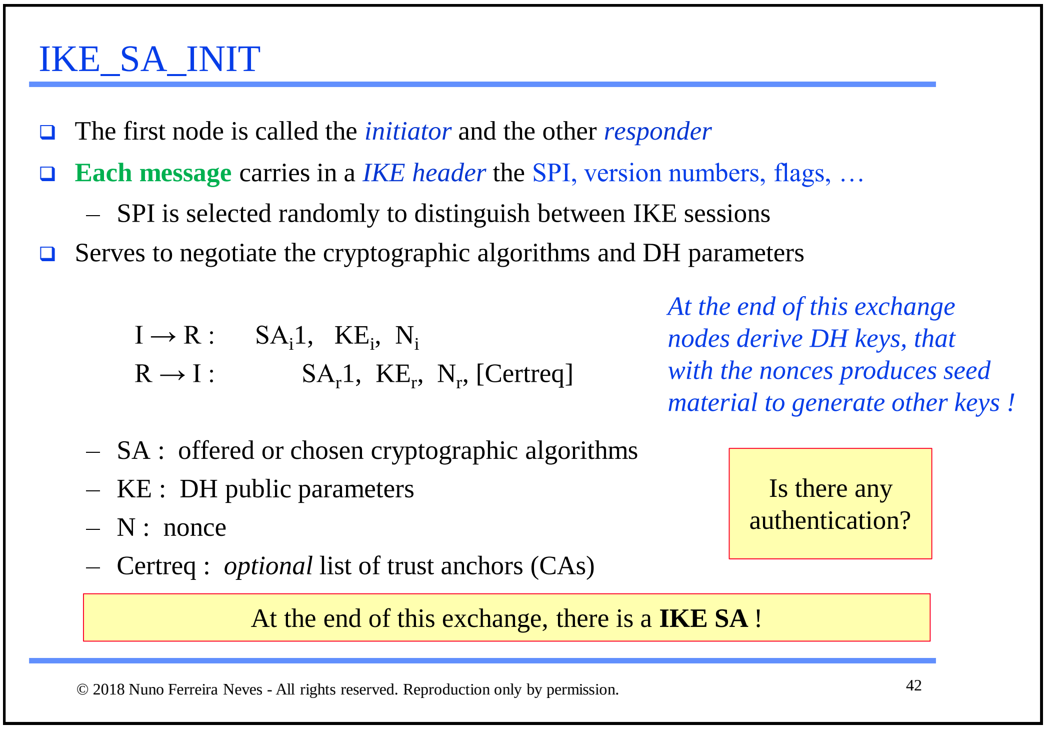
- All IKE communications consist of pairs of messages, called an **exchange**
- The initial exchange of messages establish an **IKE SA**
  - IKE_SA_INIT: negotiates the security parameters for the IKE SA, sends nonces and DH values
  - IKE_AUTH: sends the identities and cert, proves the knowledge of secrets, and sets up the first (and often only) AH or ESP **Child SA**

IKE Phase 1

- Subsequent IKE exchanges are
  - CREATE_CHILD_SA: sets up a new Child SA
  - INFORMATIONAL: deletes a SA and reports error conditions

IKE Phase 2

- Typically, there is an IKE_SA_INIT + IKE_AUTH to create the IKE SA **and** the first Child SA (4 messages)
- Further Child SA are created using the exchange CREATE_CHILD_SA
- The IKE SA is used to send control information, while the Child SA are used to send the actual IPsec data



IKE_SA_INIT

- The first node is called the *initiator* and the other *responder*
- **Each message** carries in a *IKE header* the SPI, version numbers, flags, …
  - SPI is selected randomly to distinguish between IKE sessions
- Serves to negotiate the cryptographic algorithms and DH parameters

$I \rightarrow R$: SA_i1, KE_i, N_i

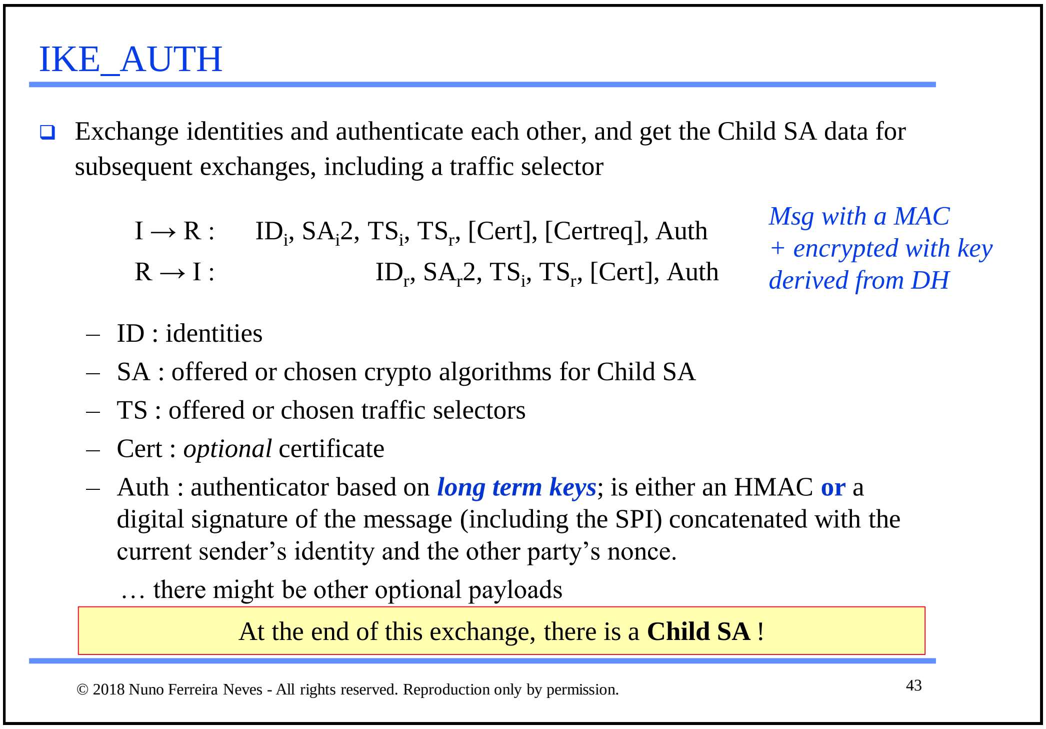
$R \rightarrow I$: SA_r1, KE_r, N_r, [Certreq]

At the end of this exchange nodes derive DH keys, that with the nonces produces seed material to generate other keys !

- SA : offered or chosen cryptographic algorithms
- KE : DH public parameters
- N : nonce
- Certreq : *optional* list of trust anchors (CAs)

Is there any authentication?

At the end of this exchange, there is a **IKE SA** !



IKE_AUTH

- Exchange identities and authenticate each other, and get the Child SA data for subsequent exchanges, including a traffic selector

$I \rightarrow R$: ID_i, SA_i2, TS_i, TS_r, [Cert], [Certreq], Auth

$R \rightarrow I$: ID_r, SA_r2, TS_i, TS_r, [Cert], Auth

Msg with a MAC
+ encrypted with key
derived from DH

– ID : identities

– SA : offered or chosen crypto algorithms for Child SA

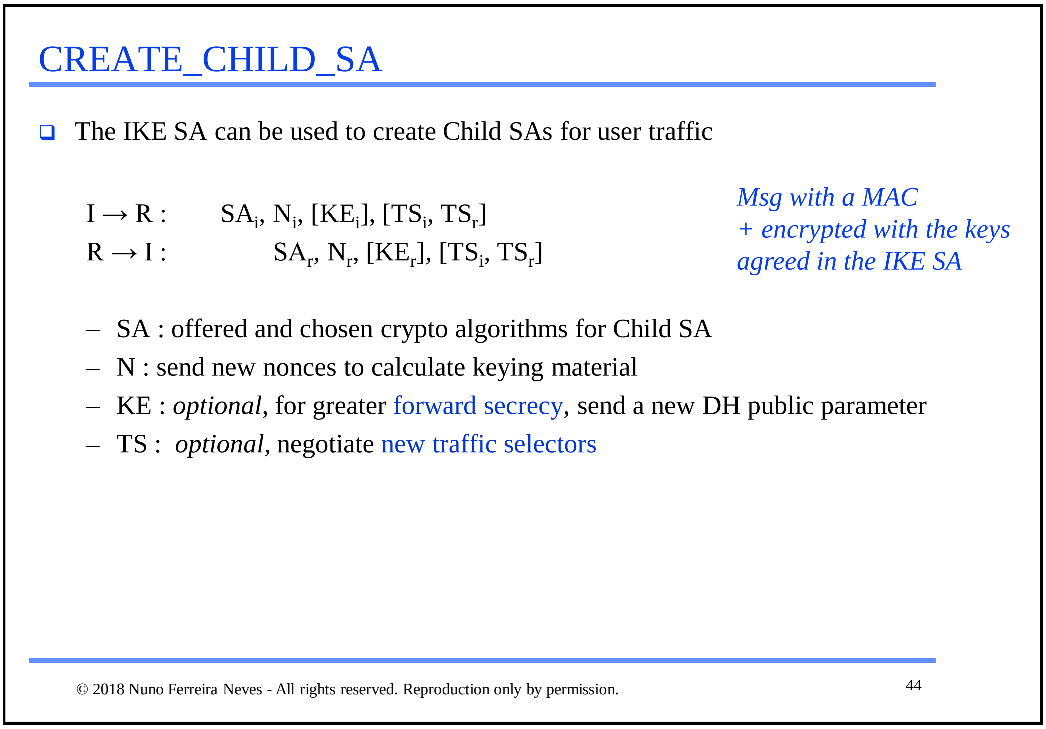
– TS : offered or chosen traffic selectors

– Cert : *optional* certificate

– Auth : authenticator based on ***long term keys***; is either an HMAC **or** a digital signature of the message (including the SPI) concatenated with the current sender's identity and the other party's nonce.

… there might be other optional payloads

At the end of this exchange, there is a **Child SA** !



CREATE_CHILD_SA

- The IKE SA can be used to create Child SAs for user traffic

$I \rightarrow R$: SA_i, N_i, [KE_i], [TS_i, TS_r]

$R \rightarrow I$: SA_r, N_r, [KE_r], [TS_i, TS_r]

Msg with a MAC
+ encrypted with the keys
agreed in the IKE SA

– SA : offered and chosen crypto algorithms for Child SA

– N : send new nonces to calculate keying material

– KE : *optional*, for greater forward secrecy, send a new DH public parameter

– TS : *optional*, negotiate new traffic selectors

Renegotiate the SA and Rekeying

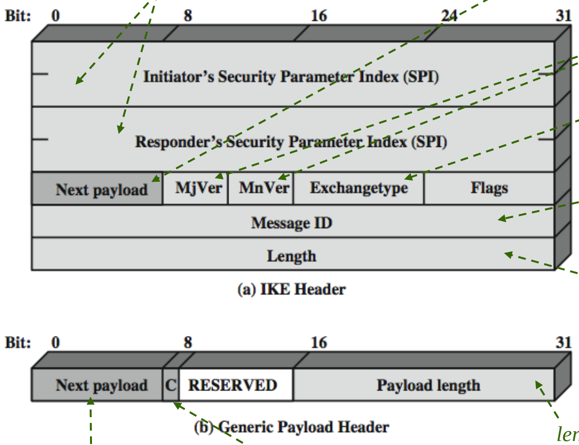
- SAs do not have pre-determined lifetimes (with the exception that sequence numbers can not be repeated)
- A new SA is renegotiated when either side thinks that it has been around for long enough
- The effect is that the SA lifetime is the shorter of the two sides' preferences

- A SA can also be rekeyed to setup fresh keys
- A rekey message is sent with the SA identifier, new nonces, and perhaps new DH public parameters (traffic selectors are not needed)
- The result is that the SA continues to exist but with new keys

Header

SPI chosen be the initiator and responder to identify the IKE security association

type of the first payload



(a) IKE Header

(b) Generic Payload Header

IKE version

type of exchange

unique id for the exchange (match request and response)

length of header + all payloads in bytes

zero if this is the last payload

critical / non-critical payload

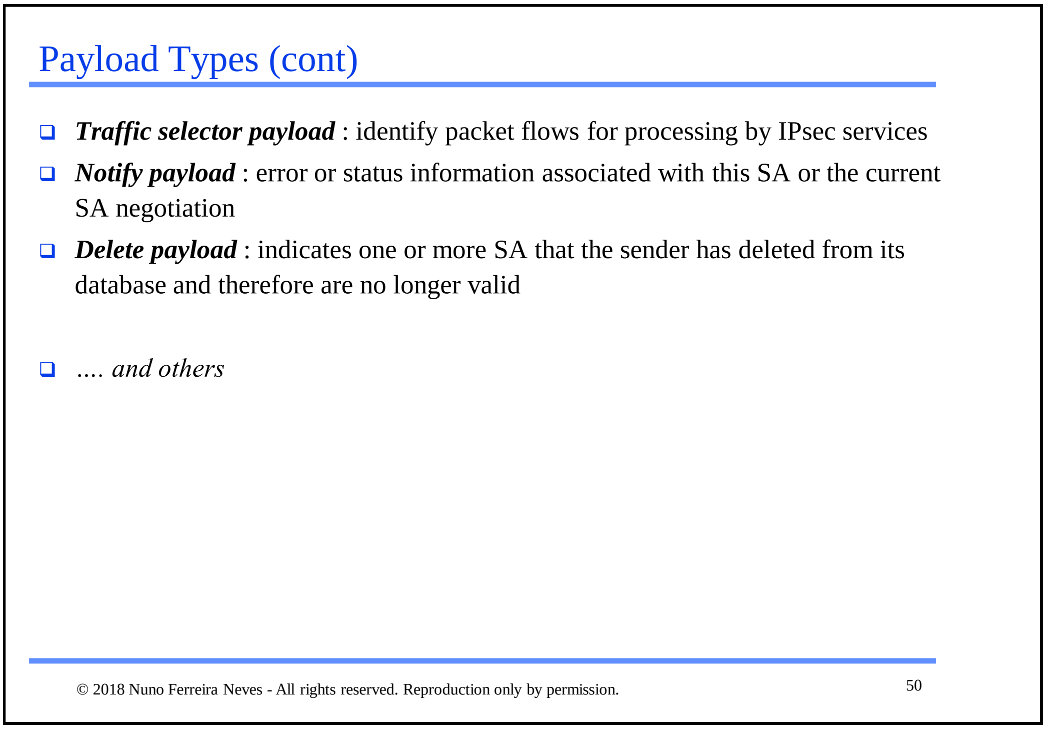
length of this payload data in bytes (including this header)

Payload Types

- ***SA payload*** : begin the establishment of a SA; it may contain multiple proposals, and each proposal can have various protocols, and each protocol can include multiple transforms, and each transform can contain several attributes
  - *proposal*: proposal number, protocol ID (IKE, AH, ESP), the number of transforms, and the transform structure
  - *transform*: defines a security transform to protect the channel; defines cryptographic algorithms to be used with a particular protocol
  - *attribute*: serves to complete the specification of the transform (e.g., key length)
- ***Key exchange payload*** : to support a variety of key exchange protocols (including Oakley, Diffie-Hellman, RSA-based key exchange)
  - *key exchange data* : depends on the protocol, and contains all information required to generate the session key

Payload Types (cont)

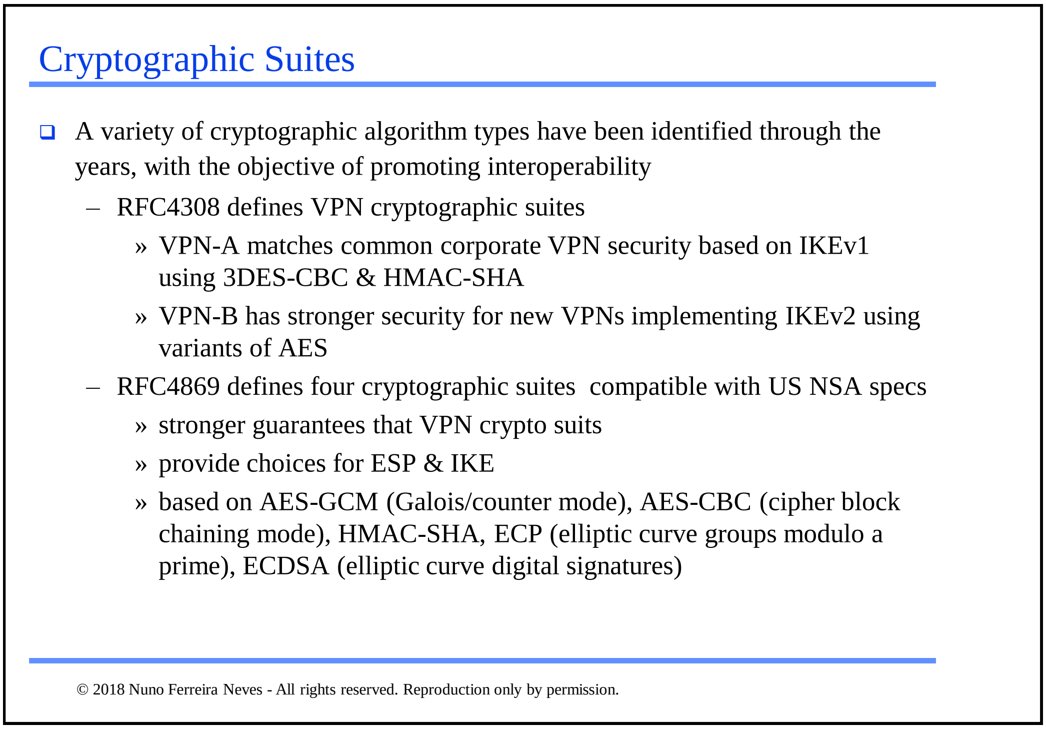
- ***Identification payload*** : to determine the identity of the communicating peers, and may be used to determine the authenticity of information
  - *ID type and data* : typically a IPv4 or IPv6 address
- ***Certificate payload*** : carries a public-key certificate
  - *certificate encoding* : the type of certificate (PGP certificate, X.509-signature)
  - *certificate data* : the certificate
- ***Certificate request payload*** : request a certificate from the other party
  - *certificate types*: acceptable types of certificates
  - *certificate authorities* : acceptable CA
- ***Authentication payload:*** contains data used for message authentication purposes (RSA digital signature, shared-key message integrity code, DSS digital signature)
- ***Nonce payload*** : contains random data



Payload Types (cont)

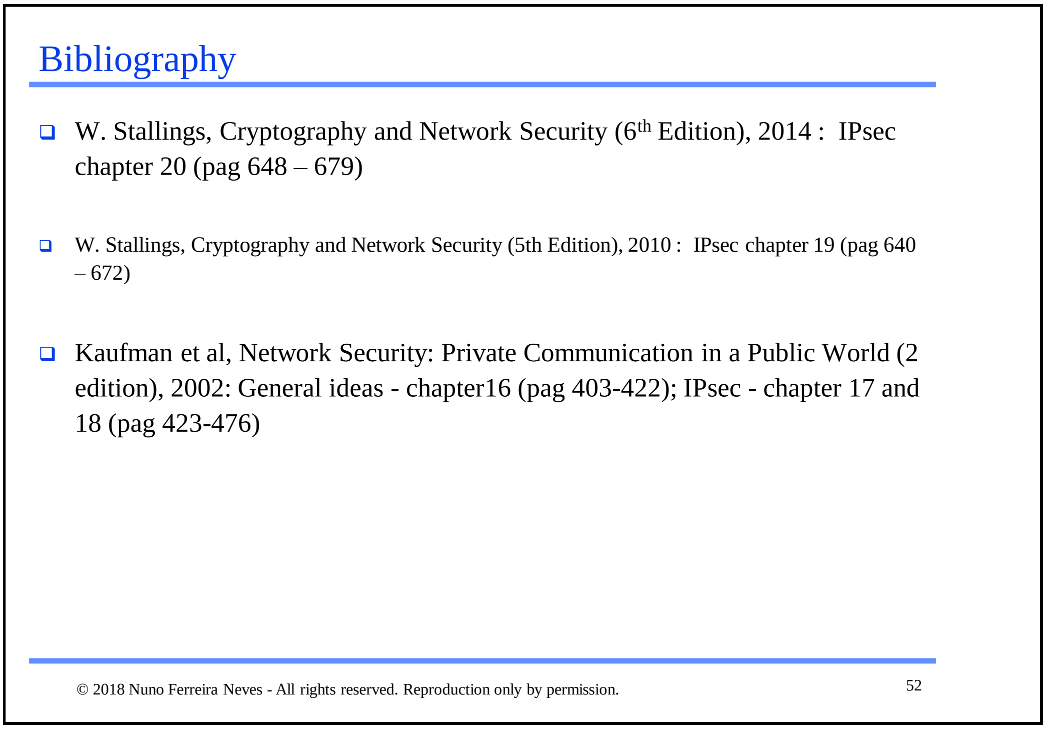
- ❑ ***Traffic selector payload*** : identify packet flows for processing by IPsec services
- ❑ ***Notify payload*** : error or status information associated with this SA or the current SA negotiation
- ❑ ***Delete payload*** : indicates one or more SA that the sender has deleted from its database and therefore are no longer valid

- ❑ *…. and others*



Cryptographic Suites

- ❑ A variety of cryptographic algorithm types have been identified through the years, with the objective of promoting interoperability
  - – RFC4308 defines VPN cryptographic suites
    - » VPN-A matches common corporate VPN security based on IKEv1 using 3DES-CBC & HMAC-SHA
    - » VPN-B has stronger security for new VPNs implementing IKEv2 using variants of AES
  - – RFC4869 defines four cryptographic suites compatible with US NSA specs
    - » stronger guarantees that VPN crypto suits
    - » provide choices for ESP & IKE
    - » based on AES-GCM (Galois/counter mode), AES-CBC (cipher block chaining mode), HMAC-SHA, ECP (elliptic curve groups modulo a prime), ECDSA (elliptic curve digital signatures)



Bibliography

- W. Stallings, Cryptography and Network Security (6th Edition), 2014 : IPsec chapter 20 (pag 648 – 679)

- W. Stallings, Cryptography and Network Security (5th Edition), 2010 : IPsec chapter 19 (pag 640 – 672)

- Kaufman et al, Network Security: Private Communication in a Public World (2 edition), 2002: General ideas - chapter16 (pag 403-422); IPsec - chapter 17 and 18 (pag 423-476)