La suite di protocolli IPSec

IPSec

- IPsec è una collezione di protocolli formata da
  - Protocolli che forniscono la cifratura-autenticità del flusso di dati (ESP, AH)
  - Protocolli che implementano lo *scambio iniziale delle chiavi* per realizzare il flusso crittografato (ISAKMP+IKE).

Set up Ipsec (1)

http://www.unixwiz.net/techtips/iguide-ipsec.html#flavors

- **AH *vs* ESP**
  - **AH** (autentica)
    - garantisce l'autenticazione e l'integrità del messaggio ma non offre la confidenzialità
  - **ESP** (cifra+autentica)
    - fornisce autenticazione, confidenzialità e controllo di integrità del messaggio

Set up Ipsec (2)

- **Tunnel mode *vs* Transport mode**
  - IPsec supporta queste due modalità di funzionamento
  - **Transport Mode**
    - offre una connessione sicura tra endpoints (host-to-host)
    - viene cifrato solo il payload dei datagram IP e non l'header
    - computazionalmente leggero
  - **Tunnel Mode**
    - connessione gateway-to-gateway
    - viene cifrato tutto il pacchetto IP originale
    - computazionalmente oneroso
    - solo i gateway devono avere il software Ipsec

Set up IPsec (3)

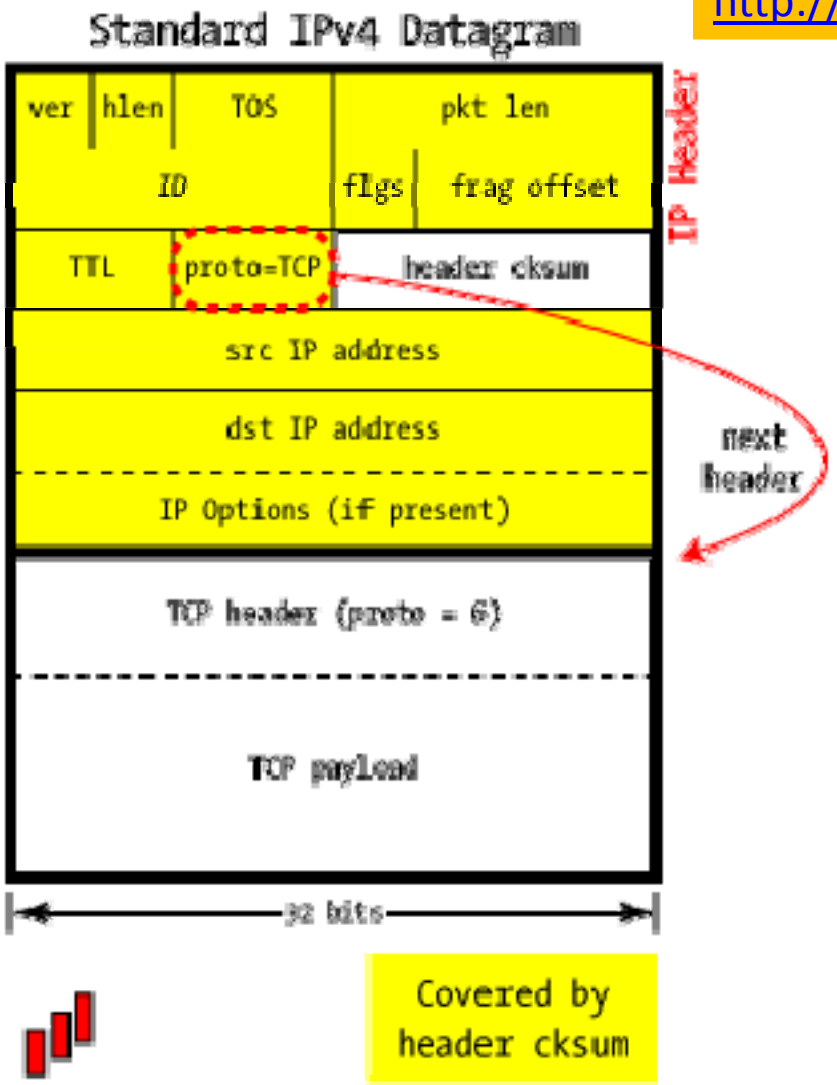
- MD5 *vs* SHA-1 *vs* DES *vs* 3DES *vs* AES *vs* blah blah blah
  - Metodi di cifratura, ogni connessione può adottarne 2-3 per volta
    - *In modalità Authentication* usato per calcolare un valore ICV (Integrity Check Value) sul contenuto del pacchetto, tipicamente costruito su un valore hash cifrato conMD5 o SHA-1. Include una chiave segreta nota ad entrambe le parti e ciò consente di calcolare l'ICV nello stesso modo.
    - *In modalità Encryption* usati con una chiave segreta per cifrare I dati prima della trasmissione (algoritmi come DES, 3DES, Blowfish, AES).

Set up Ipsec (4)

- **IKE *vs* manual keys**
  - **I**nternet **k**ey **e**xchange ed è il protocollo usato per stabilire una *security association (SA)*
    - usato per stabilire uno *shared session secret*, ossia una chiave condivisa corrispondente alla sessione da instaurare
    - dalla *shared secret* vengono successivamente derivate le chiavi crittografiche che verranno utilizzate per la successiva comunicazione.
  - *Manual keys* richiede una gestione manuale per lo scambio delle chiavi (che avviene fuori banda)
- **Main mode *vs* aggressive mode**
  - efficiency-*versus*-security tradeoff durante la fase iniziale di scambio delle chiavi (IKE, Initial key exchange).
    - Main mode richiede 6 pacchetti e offre sicurezza durante l'inizializzazione di una connessione IPSec
    - Aggressive Mode utilizza la metà dei messaggi. Il prezzo da pagare per la maggior velocità è una minore sicurezza, alcune informazioni sono trasmesse in chiaro

Il Datagramma IP tradizionale (1)

http://www.unixwiz.net/techtips/iguide-ipsec.html#ip



ver

Versione del protocollo

hlen

Lunghezza dell'header IP, a 4 bit, fornisce la lunghezza dell'intestazione del datagramma misurata in parole a 32 bit. Un header IPv4 è 20 bytes (5 parole).

TOS

Tipo di Servizio. Specifica come deve essere trattato il datagram (optimize for bandwidth? Latency? Low cost? Reliability?)

pkt len

Lunghezza totale del pacchetto IP (fino a 65535). Include I bytes dell'header.

ID

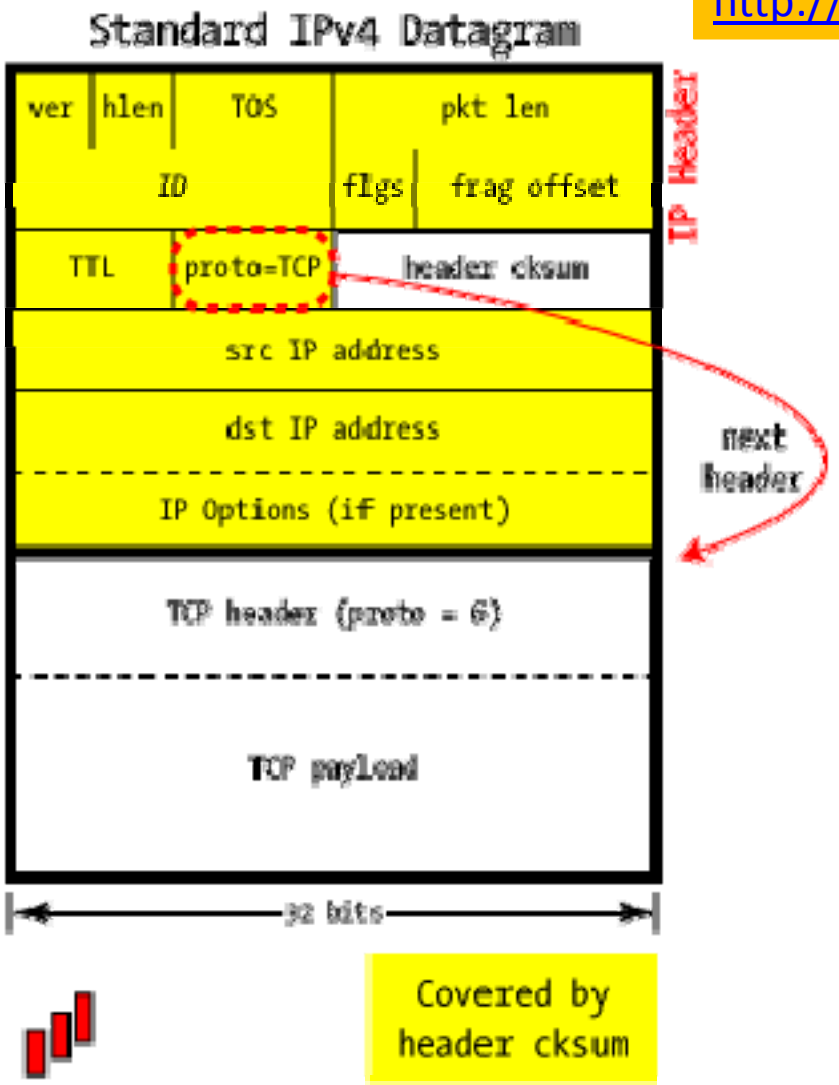
Usato per associare pacchetti correlati che sono stati frammentati

flgs

Bit utilizzati per il controllo del protocollo e della frammentazione dei datagrammi

Il Datagramma IP tradizionale (1)

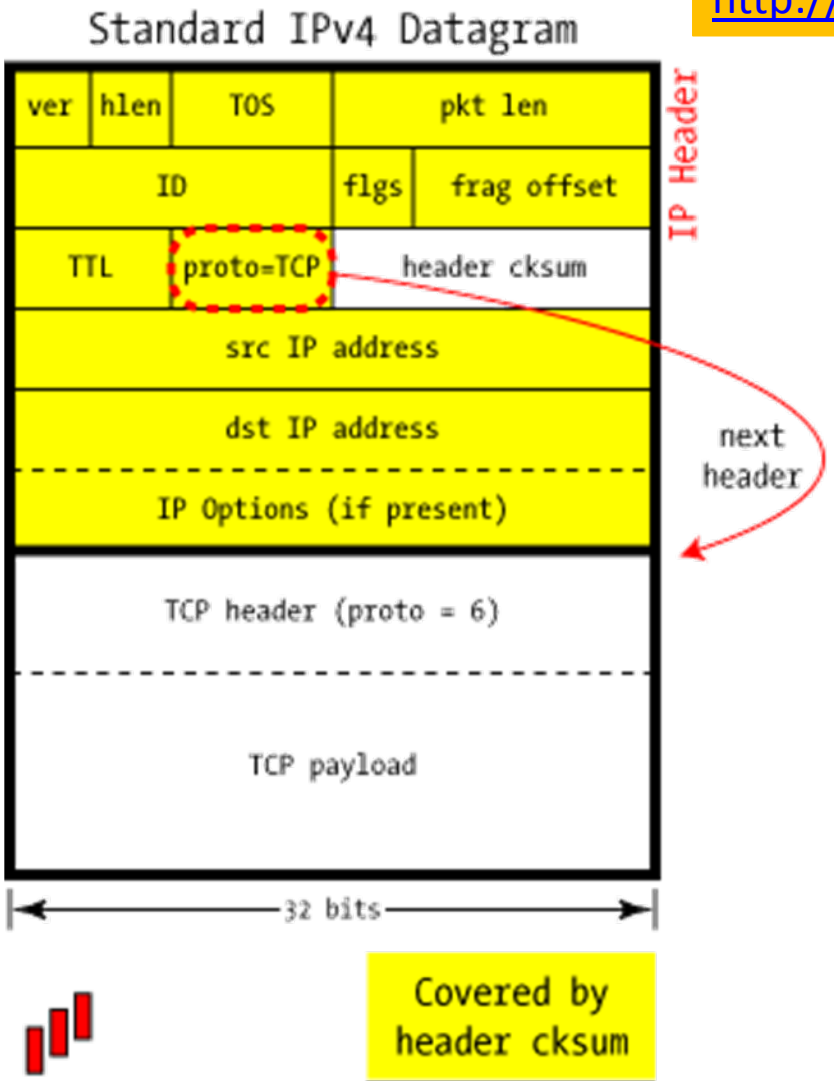
http://www.unixwiz.net/techtips/iguide-ipsec.html#ip



- **frag offset**
  - indica l'offset (misurato in blocchi di 8 byte) di un particolare frammento
- **TTL**
  - Indica il *tempo di vita* del datagramma.
- **proto**
  - Indica il codice associato al protocollo utilizzato nel campo dati del datagramma IP: per esempio al protocollo TCP è associato il codice *6*, ad UDP il codice *17. Altri protocolli (47, GRE. 50, ESP. 51, AH)*
- **header chksum**
  - È un campo usato per il controllo degli errori dell'header. Non è un checksum cifrato e non tiene conto della parte del datagramma che segue l'IP header.

Il Datagramma IP tradizionale (1)

http://www.unixwiz.net/techtips/iguide-ipsec.html#ip



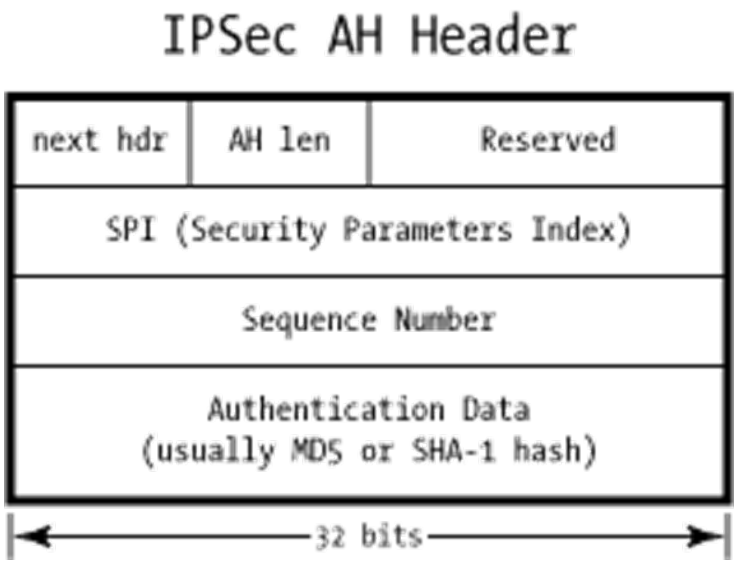
- **src IP address**
  - Indica l'*indirizzo IP associato all'host del mittente* del datagramma (32-bit)
- **dst IP address**
  - Indica l'*indirizzo IP associato all'host del destinatario* del datagramma
- **IP Options**
  - Opzioni (facoltative e non molto usate) per usi più specifici del protocollo.
- **Payload**
  - I dati in transito.

AH: Authentication Only (2)

- AH è usato per autenticare, non cifrare il traffico IP
  - garantisce che stiamo parlando con chi noi pensiamo che sia, individua alterazioni dei dati in transito, e opzionalmente può ostacolare attacchi da parte di chi cattura i dati dalla rete e cerca di ri-iniettarli in un secondo momento
- *Authentication*
  - si ottiene calcolando un codice di autenticazione hash su tutti i campi del pacchetto IP (tranne quelli che cambiano perchè modificati durante il percorso, come TTL, checksum) e memorizzando questo valore in un nuovo header AH

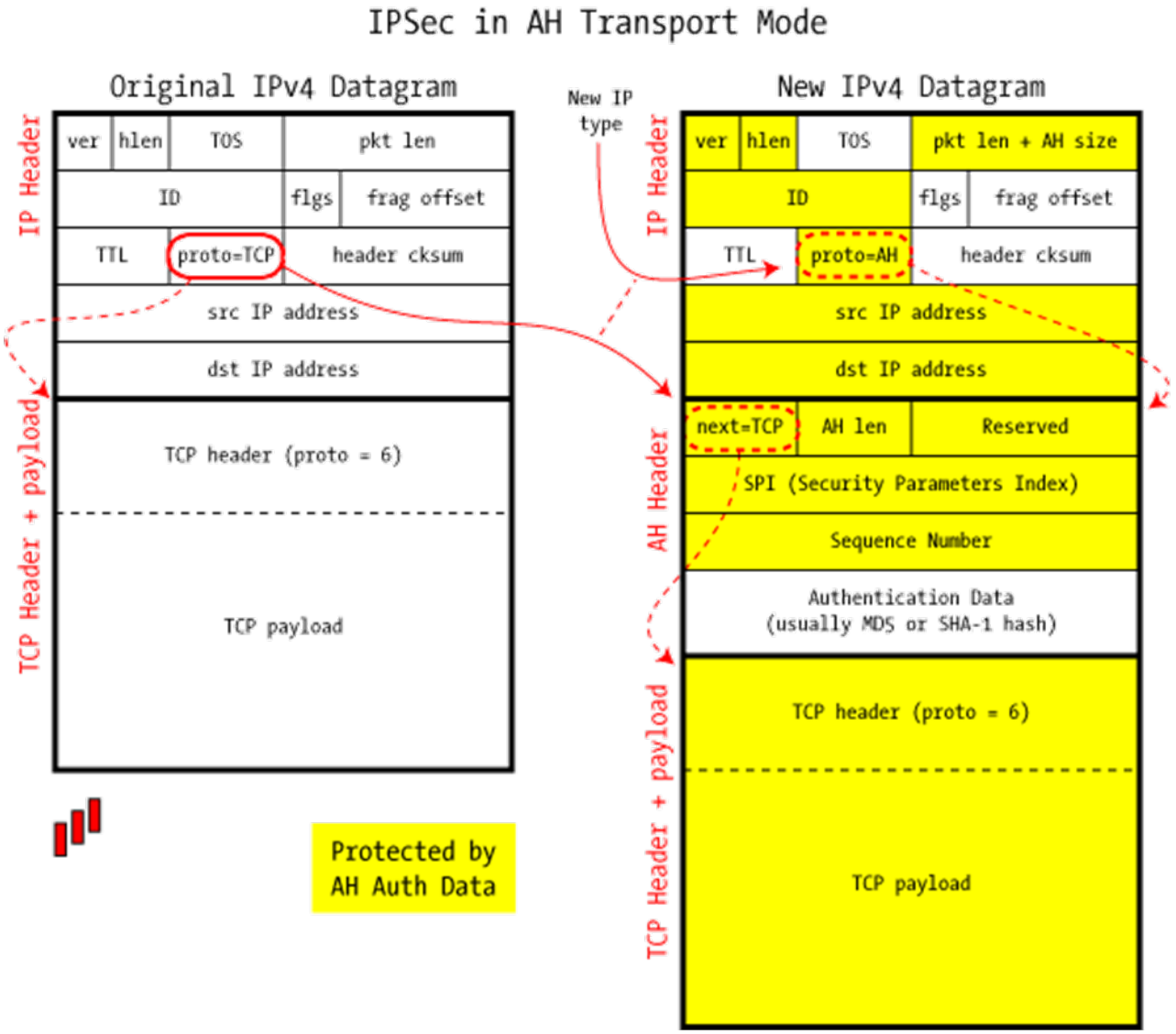
AH: Authentication Only (1)

http://www.unixwiz.net/techtips/iguide-ipsec.html#ah



- **next hdr**
  - Indica che tipo di protocollo verrà dopo.
- **AH len**
  - La lunghezza dell'AH in word
- **Reserved**
  - Spazio lasciato per sviluppi futuri. Tutti i bit di sono impostati a 0.
- **Security Parameters Index**
  - identifica i parametri di sicurezza correnti in combinazione con la coppia di indirizzi IP.
- **Sequence Number**
  - Una successione di numeri monotonicamente crescenti, è usato per impedire i replay attack.
- **Authentication Data**
  - Contiene l'Integrity Check Value (ICV)

AH Transport Mode (1)



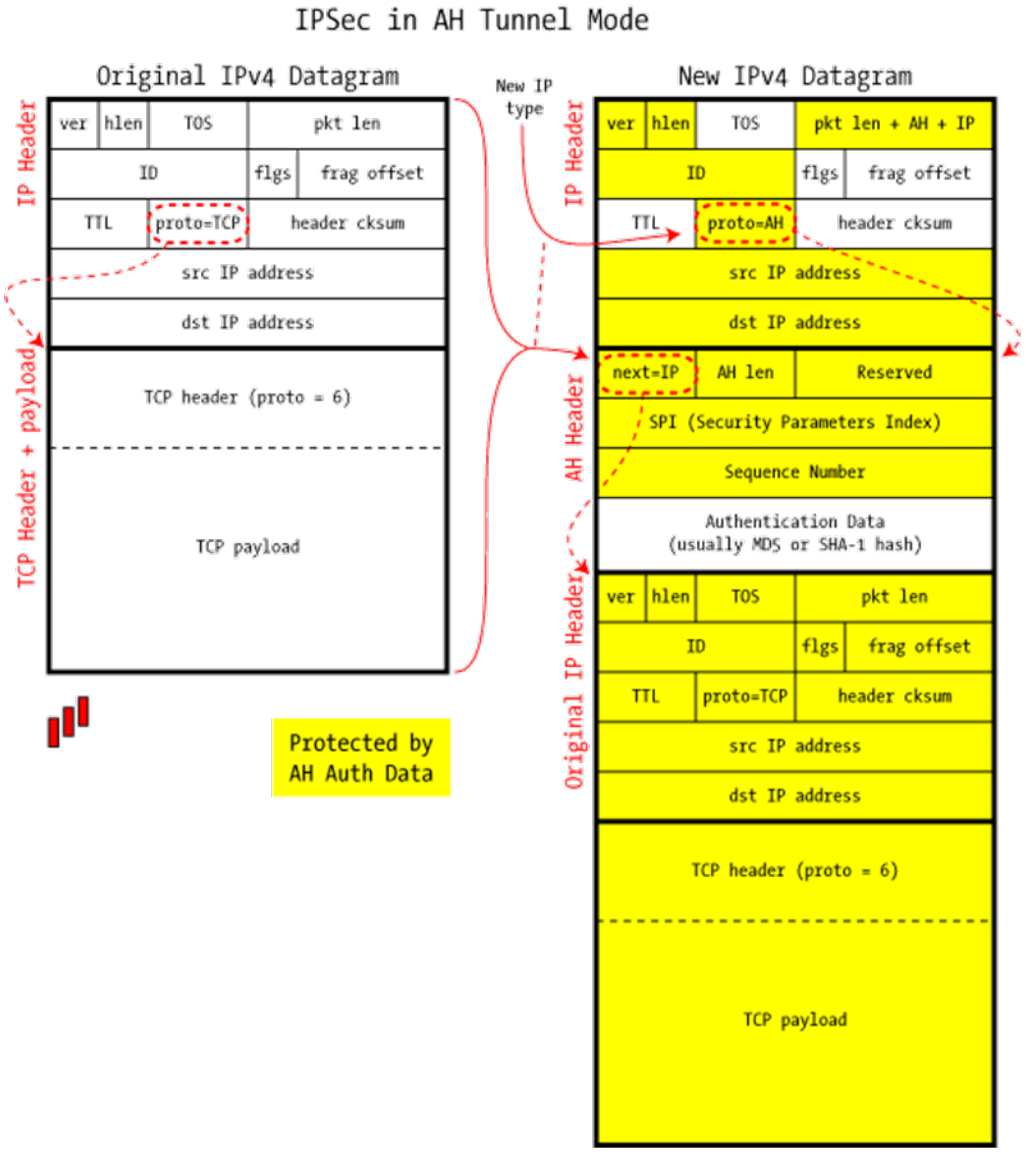
AH Transport Mode (2)

- È usato per proteggere conversazioni end-to-end tra due hosts.
  - La protezione può essere solo autenticazione.
  - Solo il payload del datagramma IP viene trattato da IPsec che inserisce il proprio header tra l'header IP ed i livelli superiori

AH Transport Mode (3)

- Quando per proteggere il traffico viene utilizzato il protocollo AH in transport mode, un nuovo header AH viene aggiunto tra l'header IP e il protocol payload (TCP, UDP, etc.)
- Nell'header IP viene modificato il campo protocol per indicare che il prossimo header da trattare è il protocollo AH (campo next header)
- Poi il pacchetto IP intero così ottenuto ad eccezione di alcuni campi mutabili dell'header IP viene autenticato dal processo di hashing e inviato a destinazione
- Quando il pacchetto arriva a destinazione e supera il controllo di autenticazione, l'header AH viene rimosso e il campo Proto=AH nell'header IP header è rimpiazzato con "Next Protocol"

AH Tunnel Mode (1)



AH Tunnel Mode (2)

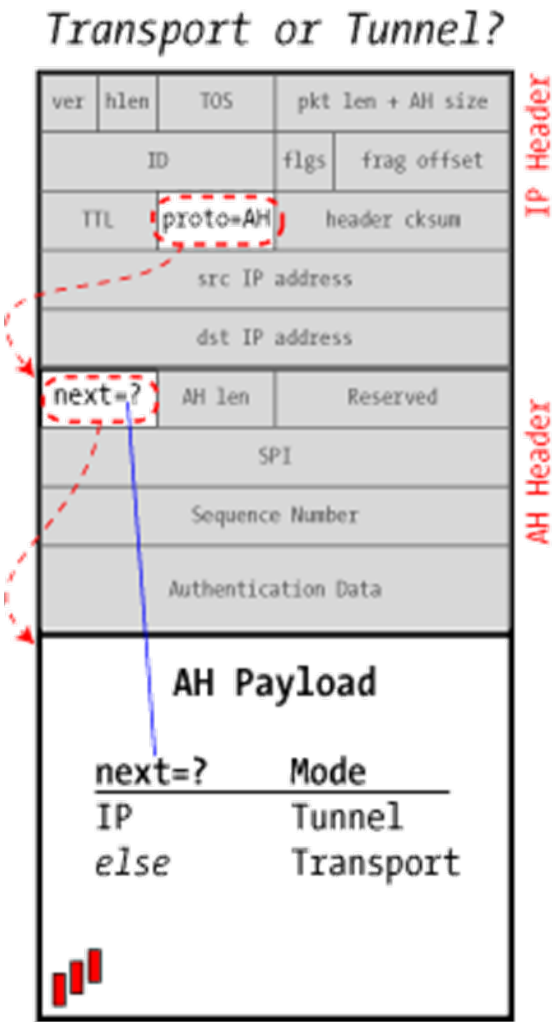
- Nel tunnel mode il datagramma IP viene completamente incapsulato in un nuovo datagramma IP utilizzando IPsec.
  - Il pacchetto viene fornito di un Integrity Check Value per autenticare il mittente e prevenire alterazioni
  - viene incapsulato l'intero header IP e il payload e ciò consente alla sorgente e destinazione di essere diversi da quelli del pacchetto che li contiene (ciò consente la creazione di un tunnel).

AH Tunnel Mode (3)

- Quando il pacchetto arriva a destinazione, dopo il controllo di autenticazione l'intero IP e header AH vengono estrapolati
  - il datagramma IP originale viene ricostruito e può essere recapitato localmente o altrove (in accordo alla destinazione IP incapsulata nel pacchetto)

- *Transport mode* è usato per la sicurezza di una connessione end-to-end tra due computers,
- *Tunnel mode* è usato invece tra due gateway (routers, firewalls, o standalone VPN devices) per fornire una Virtual Private Network

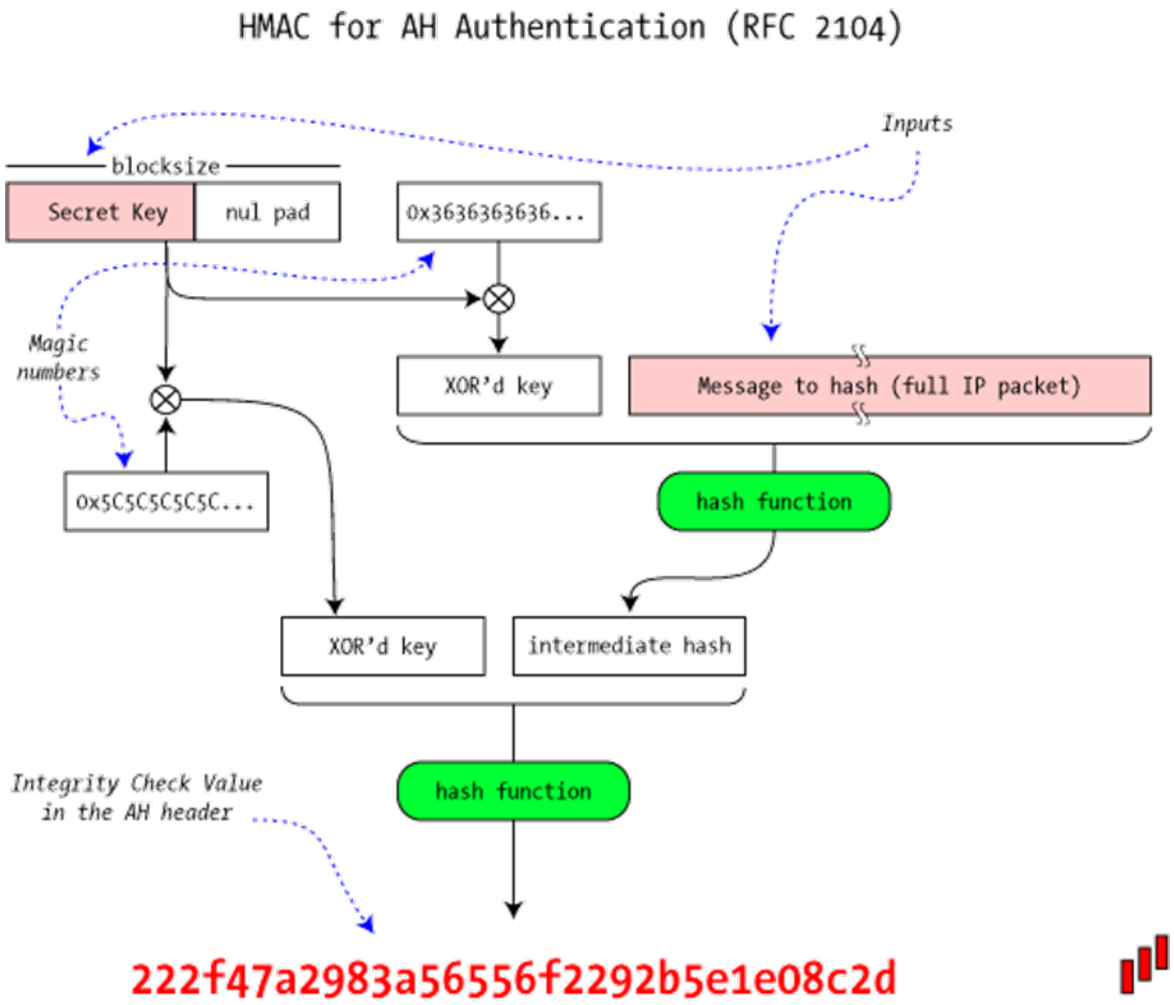
Transport or Tunnel? (1)



Transport or Tunnel? (2)

- Non c'è esplicitamente un campo "Mode" in Ipsec….come distinguere Transport mode da Tunnel mode?
  - In base al campo *next header* nell' header AH
    - se next-header è *IP*, significa che il pacchetto incapsula un intero datagramma -> Tunnel mode.
    - Ogni altro valore (TCP, UDP, ICMP,) ->Transport mode

Authentication Algorithms (1)

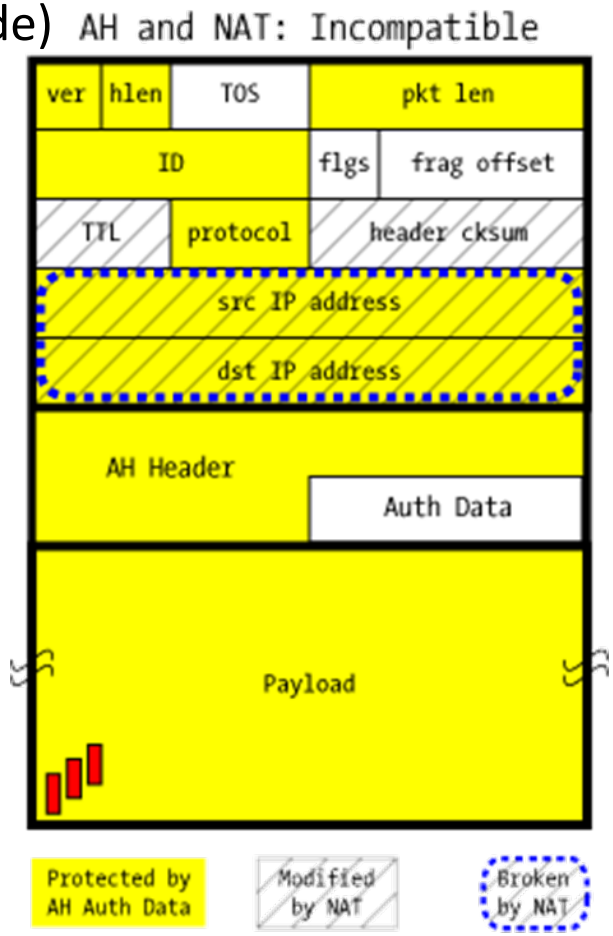


Authentication Algorithms (1)

- AH utilizza un Integrity Check Value nella porzione Authentication Data dell'header, costuita in base a algoritmi come MD5 o SHA-1.
  - Piuttosto che un semplice checksum usa un *Hashed Message Authentication Code* (HMAC) che include un valore segreto nel creare l' ICV.
  - In tal modo pur ricostruendo l'hash un attacker dovrebbe conoscere il valore segreto per ricreare l'esatto ICV

AH and NAT

- AH non è compatibile con NAT (Network Address Translation)!
  - (sia in tunnel mode che in transport mode)



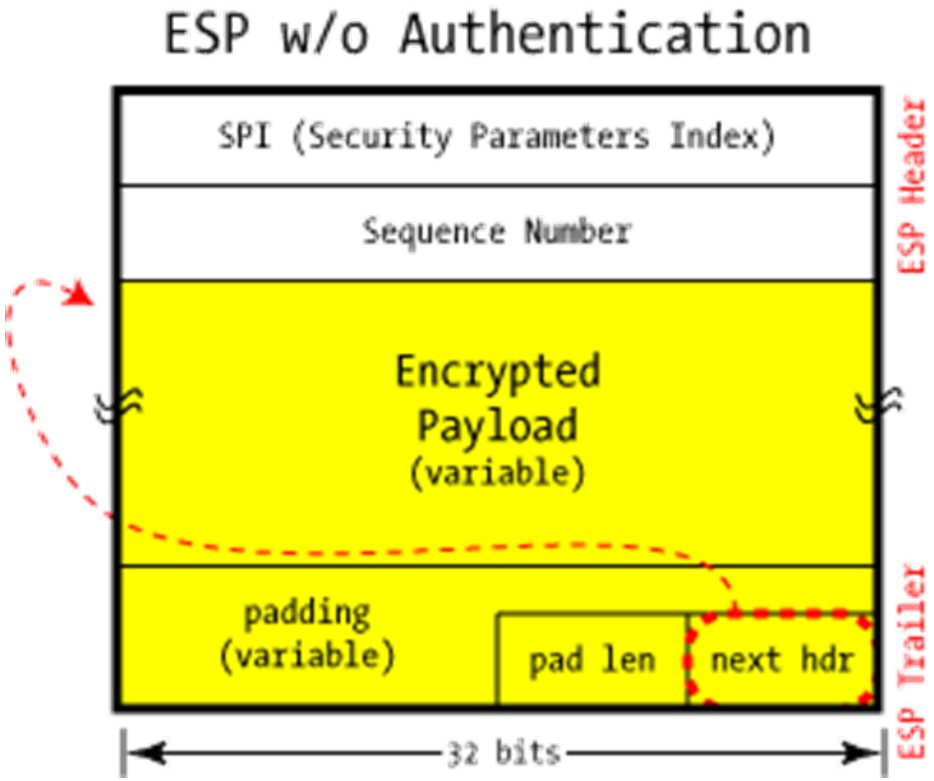
AH and NAT

- AH verifica l'integrità di tutto il pacchetto IP
- AH altera i campi indirizzo nell'header IP
  - in ricezione la checksum fallisce subito.

- ESP non copre l'header IP con controlli di sorta né in Tunnel mode né in Transport mode
  - per cui risulta adatto per NAT

ESP — Encapsulating Security Payload

http://www.unixwiz.net/techtips/iguide-ipsec.html#esp



ESP — Encapsulating Security Payload

- Il suo obiettivo è fornire confidenzialità e controllo di integrità e autenticità alla comunicazione.
  - Contrariamente a quanto fa AH, l'header IP non viene coperto dai controlli.
  - Al pari di AH, però, supporta sia il tunnel mode che il transport mode.
- È possibile utilizzare solo il servizio di riservatezza, oppure solo i servizi di autenticazione e integrità (ed eventualmente anti-replay), oppure tutti e due i servizi insieme.

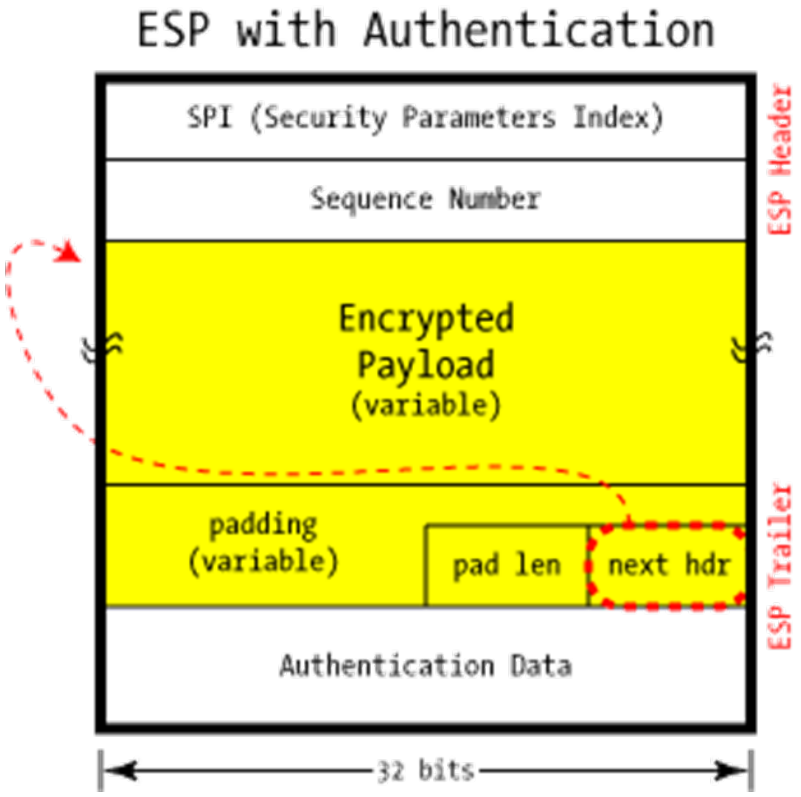
ESP -con cifratura-

- Aggiungere cifratura complica ESP perchè la cifratura *avvolge il payload* piuttosto che anteporre un header come nel caso di AH
  - ESP richiede che header e trailer supportino cifratura e opzionalmente autenticazione
  - DES, triple-DES, AES, Blowfish, sono algoritmi usati, quale scegliere viene dalla Security Association

ESP –senza cifratura-

- Usare per algoritmo di cifratura NULL
  - No confidenzialità
  - Ha senso se combinato con autenticazione ESP

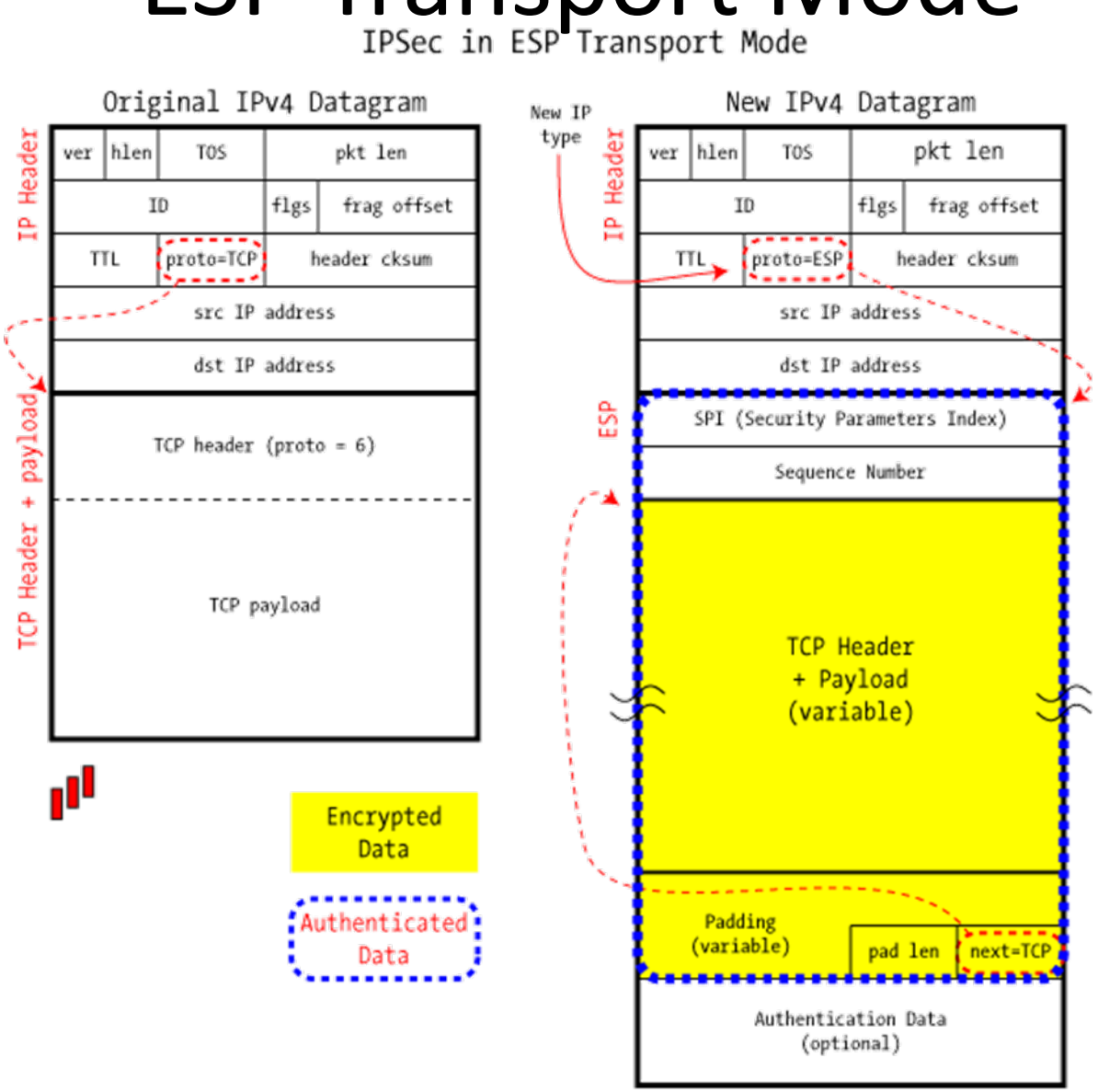
ESP –con autenticazione-



ESP –con autenticazione-

- HMAC come per AH.
  - Autenticazione solo per l'header *ESP e payload cifrato (non l'intero pacchetto)*
  - Quando dall'esterno si esamina il pacchetto IP contenente I dati ESP è impossibile indovinare il contenuto dei dati nell'header IP (sorgente e destinatario), sarà solo possibile capire che si tratta di dati ESP

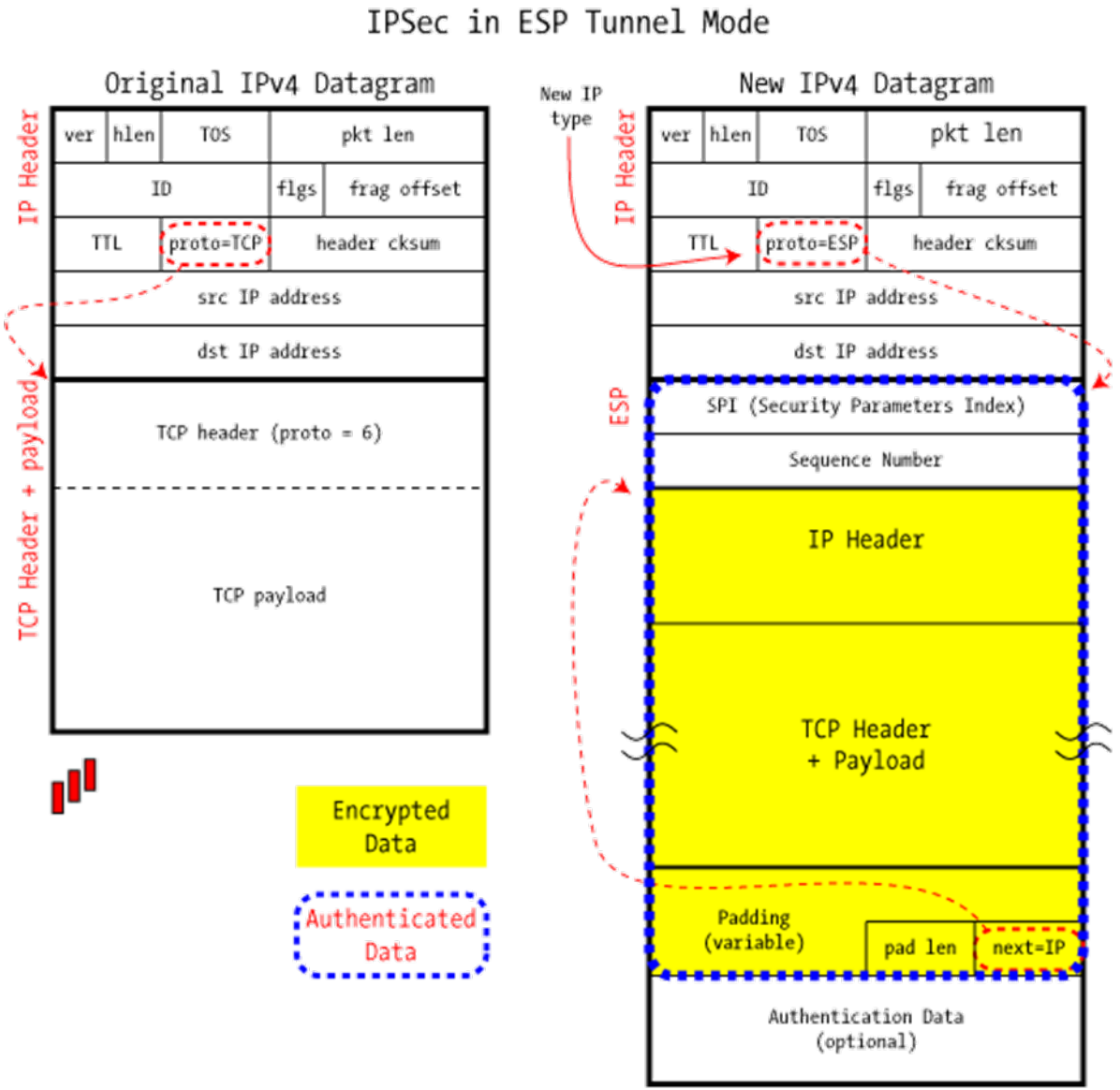
ESP Transport Mode



ESP Transport Mode

- Incapsula solo il payload del datagramma ed è pensata per comunicazioni host-to-host
- L'header IP originale resta al suo posto
  - source e destination IP addresses restano invariati

ESP Tunnel Mode



ESP Tunnel Mode

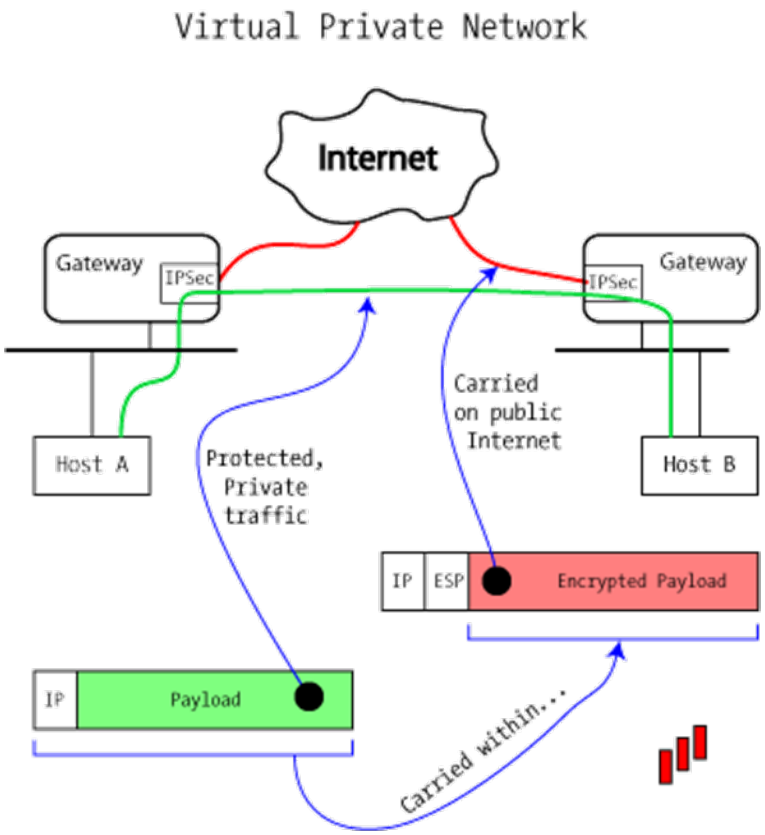
- Incapsula l’intero datagramma IP

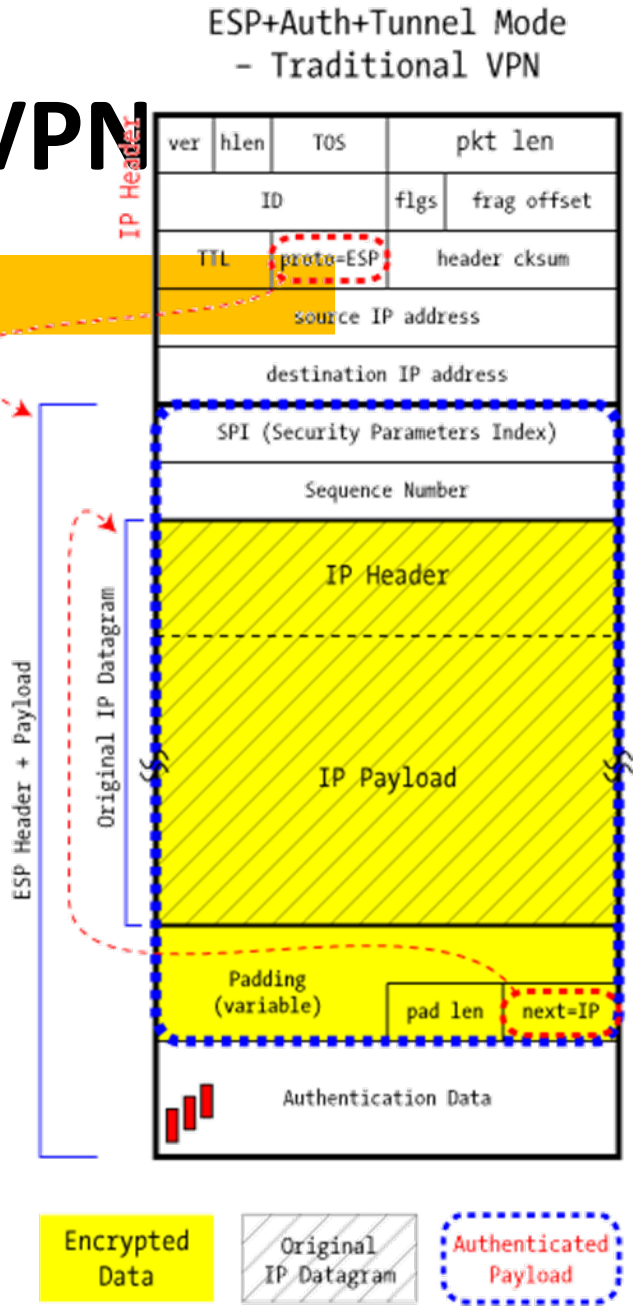
Riassunto

	Transport Mode SA	Tunnel Mode SA
AH	Autentica il payload IP, porzioni selezionate dell'intestazione IP e le intestazioni di estensione IPv6.	Autenticazione relativa all'intero pacchetto Ip incapsulato, ed alcuni campi e/o estensioni delle intestazioni IP esterne.
ESP	Cifra il payload IP e tutte le intestazioni di estensione IPv6 che seguono l'intestazione ESP.	Cifra il pacchetto IP interno.
ESP with authentication	Cifra il payload IP e tutte le intestazioni di estensione IPv6 che seguono l'intestazione ESP. Autentica il payload ma non l'intestazione IP.	Cifra il pacchetto IP interno. Autentica il pacchetto IP interno.

Costruire una VPN

http://www.unixwiz.net/techtips/iguide-ipsec.html#vpn





Security Associations and the SPI

http://www.unixwiz.net/techtips/iguide-ipsec.html#other

- **SA**: una connessione logica unidirezionale tra il mittente ed il ricevente
- Identificata da tre parametri:
  - Indice dei parametri di sicurezza (Security Parameter Index, **SPI**)
  - Indirizzo IP di destinazione
  - Identificatore del protocollo di sicurezza

Security Associations and the SPI

- Security Association Database (SADB)
  - Un database contenente SA, presente sugli host
- Security Parameter Index (SPI)
  - Indice univoco associato ad ogni entry del SADB
  - Identifica la SA associata ad un pacchetto
- Security Policy Database (SPD)
  - Memorizza le policy utilizzate per stabilire le SA (indica le preferenze su che tipo di SA sono accettabili)

IPSec

ISAKMP + IKE

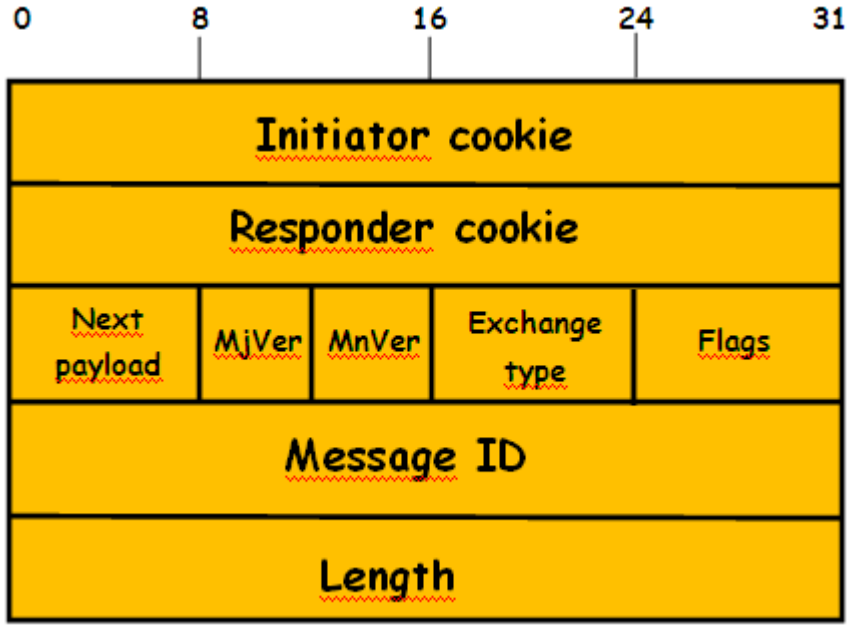
Internet Security Association and Key Management Protocol

- Il protocollo **ISAKMP**
  - definisce le procedure e i formati dei pacchetti per
    - Attivare, negoziare, modificare, cancellare le *security assoc iations*
  - Definisce il payload per lo scambio dei dati di generazione e autenticazione delle chiavi
    - indipendentemente dallo specifico protocollo di scambio delle chiavi, dall'algoritmo di crittografia e dal meccanismo dia utenticazione

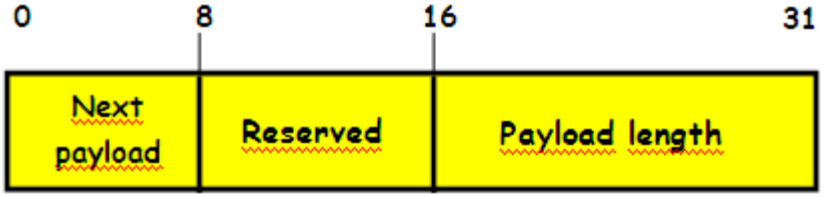
Messaggio ISAKMP

- Un messaggio ISAKMP è costituito da:
  - Intestazione + uno o più carichi utili
- Trasportato in un protocollo di trasporto
  - le specifiche richiedono il supporto per UDP

Header ISAKMP



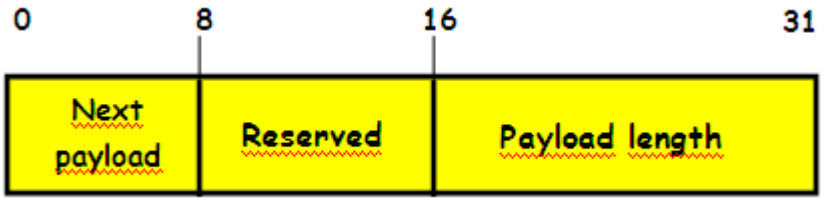
Intestazione ISAKMP



Intestazione generica del payload ISAKMP

- **Initiator Cookie (64 bit):** cookie dell'entità che ha iniziato l'attivazione, la notifica o la cancellazione della SA (serve ad evitare attacchi di tipo DOS)
- **Responder Cookie (64 bit):** cookie dell'entità che risponde; nullo nel primo messaggio dell'iniziatore
- **Next Payload (8 bit):** indica il tipo del primo payload del messaggio
- **MajorVersion (4 bit):** indica la versione (major) di ISAKMP usata
- **MinorVersion (4 bit):** indica la versione (minor) di ISAKMP usata
- **Exchange Type (8 bit):** indica il tipo di scambio
- **Flag (8 bit):** indica le opzioni impostate per lo scambio ISAKMP
- **Message ID (32 bit):** codice ID univoco del messaggio
- **Length (32 bit):** lunghezza totale del messaggio misurata in ottetti

Payload ISAKMP



Intestazione generica del payload ISAKMP

- **Next Payload (8 bit):** vale 0 se questo è l'ultimo payload del messaggio, altrimenti il suo valore è il tipo del payload successivo
- **Payload length (8 bit):** indica la lunghezza in ottetti del payload

Tipi di payload ISAKMP (1)

Tipo	Parametri	Descrizione
SA (Security Association)	Domain of interpretation, situation	Usato per negoziare gli attributi di sicurezza e indicare il dominio di interpretazione e la situazione nei quali si svolge la negoziazione
P (Proposal)	Proposal #, Protocol-ID, SPI Size, # of Transforms, SPI	Usato durante la negoziazione di una associazione di sicurezza: indica il protocollo da usare e il numero di trasformazioni
T(Transform)	Transform #, Transform-ID, SA Attributes	Usato durante la negoziazione di una associazione di sicurezza: indica gli attributi della trasformazione e della relativa associazione di sicurezza

Tipi di payload ISAKMP (2)

Tipo	Parametri	Descrizione
KE (Key Exchange)	Key Exchange data	Supporta varie tecniche di scambio delle chiavi
ID(Identification)	ID Type, ID Data	Usato per scambiare le informazioni di identificazione
CERT(Certificate)	Cert Encoding, Certificater Data)	Usato per trasportare i certificati e le altre informazioni correlate
CR (Certificate Request)	# Cert Types, Certificate Types, # Certificate Auths, certificate Authorities	Usato per richiedere certificati: indica i tipi di certificati richiesti e le autorità di certificazione accettate.
HASH(Hash)	Hash data	Contiene i dati generati da una funzione hash
SIG(Signature)	Signature Data	Contiene i dati generati da una funzione di firma digitale

Tipi di payload ISAKMP (3)

Tipo	Parametri	Descrizione
NONCE(nonce)	Nonce Data	Contiene un codice *nonce*
N(Notification)	DOI,Protocol-ID,SPI Size, Notify Message Type, SPI,Notification Data	Usato per trasmettere i dati di notifica, come per esempio una condizione d'errore
D (Delete)	DOI,Protocol-ID, SPI Size, # of SPIs, SPI (uno o più)	Indica che una associazione di sicurezza non è più valida

Tipi di payload (1)

- Il payload **SA** inizia l'attivazione di una associazione di sicurezza
  - Il parametro ***D**omain **O**f **I**nterpretation* identifica il dominio nel quale si svolge la negoziazione
  - Il parametro *Situation* definisce la politica di sicurezza della negoziazione (si specificano i livelli di sicurezza)

- Il payload **Proposal** contiene info usate durante la negoziazione dell'associazione di sicurezza
  - Indica il protocollo di questa associazione di sicurezza (AH o ESP), include l'identificatore SPI dell'iniziatore e il numero di trasformazioni

- Il payload **Transform** definisce la trasformazione di sicurezza da usare per rendere sicuro il canale di comunicazione per il protocollo indicato
  - Il parametro *Transform #* identifica questo specifico payload in modo che chi risponde possa usarlo per indicare l'accettazione di questa trasformazione
  - I campi *Transform ID* e *Attributes* identificano una trasformazione (3DES per ESP, HMAC-SHA-1-96 per AH) con i relativi attributi

Tipi di payload (2)

- Il payload **Key Exchange** può essere usato per varie tecniche di scambio delle chiavi (Oakley, Diffie-Hellman,..)
  - Il campo dati contiene i dati necessari a generare una chiave di sessione e dipende dall'algoritmo di scambio delle chiavi usato

- Il payload **Identification** è usato per determinare l'identità dei nodi in comunicazione e si può usare per valutare l'autenticità delle informazioni
  - In genere il campo ID Data contiene un idnirizzo IPv4/v6

- Il payload **Certificate** trasferisce un certificato a chiave pubblica
  - Il campo *Certificate Encoding* indica il tipo di certificato

- Il payload **Certificate Request** si può usare per richiedere il certificato dall'altra entità in comunicazione
  - Può elencare più tipi di certificati e autorità accettabili

Tipi di payload (3)

- Il payload Hash contiene dati generati da una funzione hash su una **parte** del messaggio e/o lo stato ISAKMP
  - Si può usare per verificare l'integrità dei dati in un messaggio o per autenticare le entità in negoziazione
- Il payload **Signature** contiene dati generati da una firma digitale su una parte del messaggio e/o lo stato ISAKMP
  - Usato per verificare l'integrità del messaggio e per servizi di non ripudiabilità
- Il payload **Nonce** contiene dati casuali
  - usati per garantire l'attualità dello scambio e proteggersi da attacchi a replay
- Il payload **Notification** contiene info di errore o di stato relative a questa associazione di sicurezza o a questa negoziazione della associazione di sicurezza
- Il payload **Delete** indica associazioni di sicurezza che il mittente ha cancellato dal proprio database e che non sono più valide

ISAKMP: scambio di messaggi

- **Base**
  - Consente lo scambio contemporaneo delle chiavi e delle info di autenticazione
  - Riduce il numero di scambi ma non protegge l'identità
- **Identity Protection**
  - Espande lo scambio base per proteggere le identità degli
- **Authentication Only**
  - Usato per svolgere la reciproca autenticazione senza scambio di chiavi
- **Aggressive**
  - Riduce il numero di scambi ma non garantisce la protezione dell'identità
- **Informational**
  - Usato per la trasmissione monodirezionale di informazioni per la gestuione dell'associazione di sicurezza

Scambio Base

- (1) **I->R**: SA;NONCE
  - Inizia la negoziazione dell'associazione di sicurezza ISAKMP
- (2) **R->I**: SA;NONCE
  - Associazione di sicurezza base concordata
- (3) **I->R:** KE;ID_I;AUTH
  - Chiave generata; identità dell'iniziatore verificata da chi risponde
- (4) **R->I:** KE;ID_R;AUTH
  - Identità di chi risponde verificata dall'iniziatore; chiave generata; associazione di sicurezza attivata

Notazione:
I= Iniziatore
R=Risponditore
*=crittografia del payload dopo l'intestazione ISAKMP
AUTH= meccanismo di autenticazione impiegato

Scambio Base

- (1) **I->R**: SA;NONCE
- (2) **R->I**: SA;NONCE
- (3) **I->R:** KE;ID_I;AUTH
- (4) **R->I:** KE;ID_R;AUTH

- Inizia la negoziazione dell'associazione di sicurezza ISAKMP
- Associazione di sicurezza base concordata
- Chiave generata; identità dell'iniziatore verificata da chi risponde
- Identità di chi risponde verificata dall'iniziatore; chiave generata; associazione di sicurezza attivata

•I primi 2 messaggi forniscono i cookie e attivano una associazione di sicurezza, le trasformazioni su e il protocollo concordati

•Entrambe le parti usano un codice nonce per proteggersi dagli attacchi a replay

•Gli ultimi 2 messaggi scambiano le informazioni delle chiavi e i codici ID utente con un meccanismo di autenticazione usato per autenticare le chiavi, le identità e i codici nonce dei primi due messaggi

Scambio Identity Protection

(1) **I->R**: SA	Inizia la negoziazione dell'associazione di sicurezza ISAKMP
(2) **R->I**: SA	Associazione di sicurezza base accordata
(3) **I->R:** KE;NONCE	Chiave generata
(4) **R->I:** KE;NONCE	Chiave generata
(5)* **I->R**: ID_I;AUTH	Identità dell'iniziatore verificata dal risponditore
(6)* **R->I**: ID_R;AUTH	Identità del risponditore verificata dall'iniziatore; associazione di sicurezza attivata

Notazione:
I= Iniziatore
R=Risponditore
*=crittografia del payload dopo l'intestazione ISAKMP
AUTH= meccanismo di autenticazione impiegato

Scambio Identity Protection

- (1) **I->R**: SA
- (2) **R->I**: SA
- (3) **I->R:** KE;NONCE
- (4) **R->I:** KE;NONCE
- (5)*** I->R**: ID_I;AUTH
- (6)*** R->I**: ID_R;AUTH

- Inizia la negoziazione dell'associazione di sicurezza ISAKMP
- Associazione di sicurezza base accordata
- Chiave generata
- Chiave generata
- Identità dell'iniziatore verificata dal risponditore
- Identità del risponditore verificata dall'iniziatore; associazione di sicurezza attivata

•I primi 2 messaggi attivano l'associazione di sicurezza.

•I due messaggi successivi eseguono lo scambio delle chiavi utilizzando codici nonce per evitare attacchi a replay

•Calcolata la chiave di sessione le due parti si scambiano messaggi crittografati che contengono le info di autenticazione come le firme digitali e opzionalmente i certificati di convalida delle chiavi pubbliche

Scambio Authentication Only

- (1) **I->R**: SA; NONCE
  - Inizia la negoziazione dell'associazione di sicurezza ISAKMP
- (2) **R->I**: SA; NONCE; IDR;AUTH
  - Associazione di sicurezza base accordata; identità del risponditore verificata dall'iniziatore
- (3)**I->R**: ID_I;AUTH
  - Identità del risponditore verificata dall'iniziatore; associazione di sicurezza attivata

Notazione:
I= Iniziatore
R=Risponditore
*=crittografia del payload dopo l'intestazione ISAKMP
AUTH= meccanismo di autenticazione impiegato

Scambio Authentication Only

- (1) **I->R**: SA; NONCE
- (2) **R->I**: SA; NONCE; IDR;AUTH
- (3)**I->R**: ID_I;AUTH

- Inizia la negoziazione dell'associazione di sicurezza ISAKMP
- Associazione di sicurezza base accordata; identità del risponditore verificata dall'iniziatore
- Identità del risponditore verificata dall'iniziatore; associazione di sicurezza attivata

•I primi 2 messaggi attivano l'associazione di sicurezza.
•Inoltre il risponditore usa il secondo messaggio per trasferire il proprio codice utente e usa l'autenticazione per proteggere il messaggio
•L'iniziatore invia il terzo messaggio per trasmettere il proprio codice utente autenticato

Scambio Aggressive

- (1) **I->R**: SA; KE; NONCE; IDI
  - Inizia la negoziazione dell'associazione di sicurezza ISAKMP e lo scambio delle chiavi
- (2) **R->I**: SA; KE; NONCE; IDR;AUTH
  - Identità del risponditore verificata dall'iniziatore; chiave generata
- (3)**I->R**: AUTH
  - Identità del risponditore verificata dall'iniziatore; associazione di sicurezza attivata

•Nel primo messaggio l'iniziatore propone un'associazione di sicurezza offrendo dei protocolli e opzioni di trasformazione. L'iniziatore attiva anche lo scambio della chiave e fornisce il proprio codice utente

•Nel secondo messaggio il risponditore indica se ha accettato l'associazione di sicurezza con un certo protocollo e una certa trasformazione, completa lo scambio della chiave e autentica le info trasmesse

•Nel terzo messaggio l'iniziatore autentica le info precedenti crittografandole con la chiave segreta di sessione segreta condivisa

Scambio Informational

- (1) **I->R**: N/D

- Cancellazione o notifica di errore o stato

Viene usato per la trasmissione monodirezionale di informazioni per la gestione dell'associazione di sicurezza

Notazione:
I= Iniziatore
R=Risponditore
***=crittografia del payload dopo l'intestazione ISAKMP
AUTH= meccanismo di autenticazione impiegato

Security Association

- Nell'architettura di IPsec è centrale il concetto di *security association*, ma né AH né ESP si preoccupano della gestione delle *SA*
- Le security associations possono essere costruite manualmente o automaticamente
  - una loro gestione manuale non è sempre praticabile
  - il protocollo **IKE** (**Internet Key Exchange**) risolve questo problema

IKE (Internet Key Exchange)

- Protocollo per la gestione automatica delle chiavi necessarie per tutte le operazioni di security fornite da IPsec
  - protocollo ibrido
  - agisce nelle fasi iniziali di una comunicazione, permettendo la creazione di SA e la gestione dell'archivio a queste dedicato
  - Si basa su **ISAKMP**

IKE (Internet Key Exchange)

- Una **Security Association** è un contratto stabilito tra 2 endpoints IPsec (hosts o security gateways)
  - Negoziazione Automatica dei parametri da usare per la connessione IPsec
  - SA distinte sono richieste per ogni sottorete o singolo hos
  - SA distinte sono richieste per connessioni inbound e outbound
  - Alle SAs sono assegnate un unico **Security Parameters Index (SPI)** e sono mantenute in un database

IKE Elementi costitutivi

- **Internet Security e Key Management Protocol (ISAKMP)**
  - L'implementazione attuale prevede l'uso combinato delle caratteristiche di due protocolli
    - **OAKLEY** (un protocollo con il quale due parti autenticate possono giungere ad un accordo circa il materiale chiave da utilizzare e di cui IKE sfrutterà le caratteristiche per lo scambio chiave;
    - **SKEME**: un protocollo di scambio chiave simile a OAKLEY di cui però IKE utilizzerà caratteristiche diverse come il metodo crittografico a chiave pubblica e quello di rinnovo veloce della chiave

IKE: Lo scopo

- Viene raggiunto attraverso una negoziazione in due fasi:
  - la prima realizza una *Internet Security Association Key Management Security Association* (ISAKMP SA)
  - Nella seconda l'ISAKMP SA viene utilizzata per la negoziazione e l'instaurazione delle IPsec SAs

IKE: Fase 1

- Stabilisce una SA per ISAKMP da utilizzare come canale sicuro per effettuare la successiva negoziazione IPSec, in particolare:
  - Negozia i parametri si sicurezza
  - Genera un segreto condiviso
  - Autentica le parti

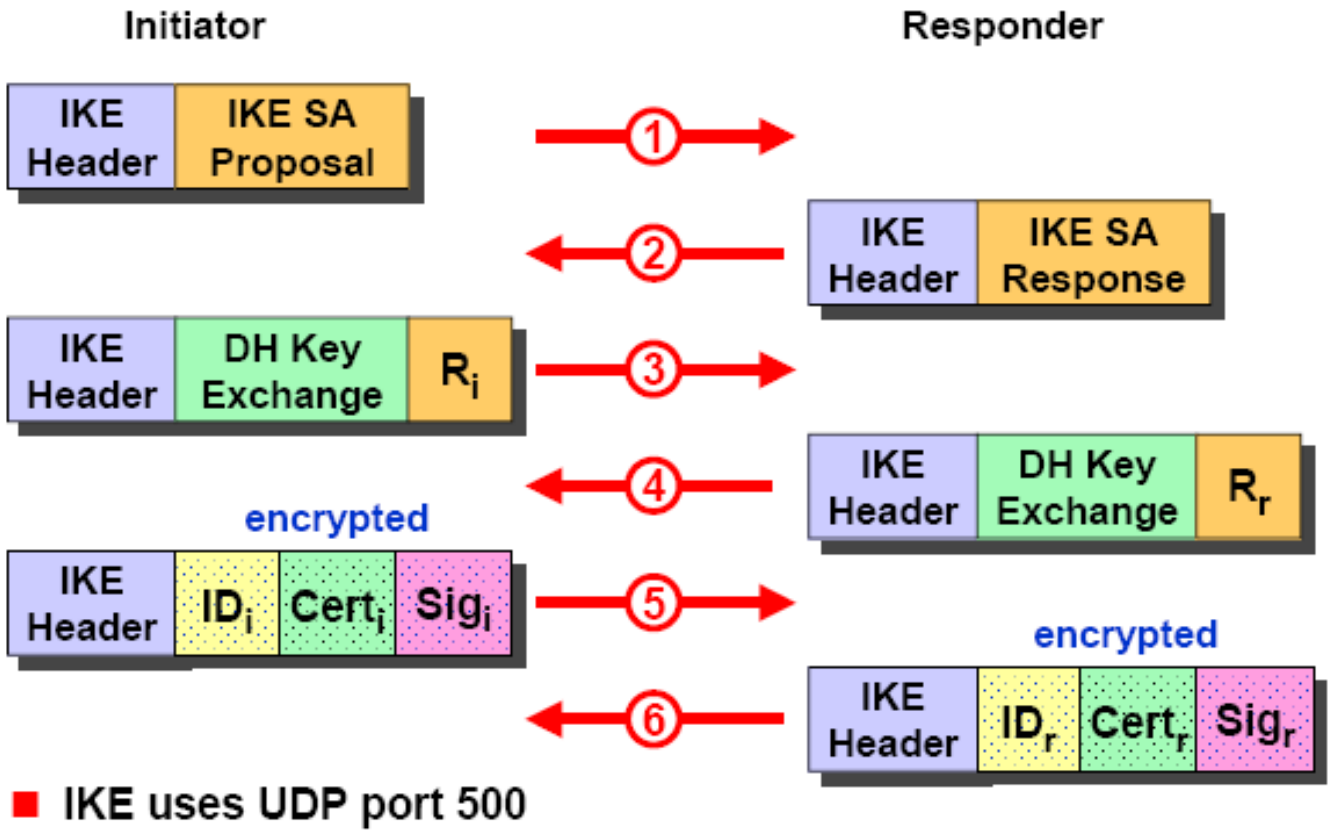
IKE: Fase 1

- Due possibili tipi di Fase 1:
  - **Main mode**: consiste nello scambio di sei messaggi di cui tre inviati dal mittente al destinatario e tre di risposta nel senso opposto
  - **Aggressive mode**: utilizza solo tre messaggi. Due messaggi inviati dal mittente ed uno di risposta.
- La differenza principale, oltre al numero di messaggi utilizzati risiede nel fatto che la prima modalità, anche se più lenta, garantisce una protezione dell'identità
  - Entrambe le modalità autenticano le parti e stabiliscono una ISAKMP SA
  - L'aggressive mode è in grado di farlo utilizzando la metà dei messaggi
    - Il prezzo da pagare per la maggior velocità è *l'assenza del supporto per l'identificazione dei partecipanti* e quindi la possibilità di attacchi di tipo man-in-the-middle nel caso di utilizzo di pre-shared keys

IKE: Fase 2

- Detta anche **Quick mode**
  - Serve principalmente a negoziare dei servizi IPSec di carattere generale ed a rigenerare il materiale chiave
  - è simile ad una negoziazione "Aggressive mode" ma meno complessa visto che sfrutta la comunicazione già in atto (vedi avanti..)

IKE Phase 1 - Main Mode
Establish a Secure Negotiation Channel



IKE Phase 1 - Main Mode
Establish a Secure Negotiation Channel

- **6 messaggi** scambiati tra initiator e responder per stabilire una ***IKE Security Association*** (**IKE SA**)
  - IKE usa la porta UDP 500
- **Msg #1**
  - L' initiator invia una *IKE SA Proposal* che elenca tutti I metodi di autenticazione supportati, Diffie-Hellman groups, una scelta di algoritmi di cifratura e hash e il tempo di vita della SA
- **Msg #2**
  - Il responder risponde con una *IKE SA Response* che indica il metodo di autenticazione preferito, Diffie-Hellman group, gli algoritimi di cifratura e hash e un tempo di vita accettabile per la SA
- Se le 2 aprti riescono a negoziare un insieme condiviso di metodi il protocollo viene completato instaurando unncanale cifrato di comunicazione usando l'algoritmo Diffie-Hellman Key-Exchange

IKE Phase 1 - Main Mode
Establish a Secure Negotiation Channel

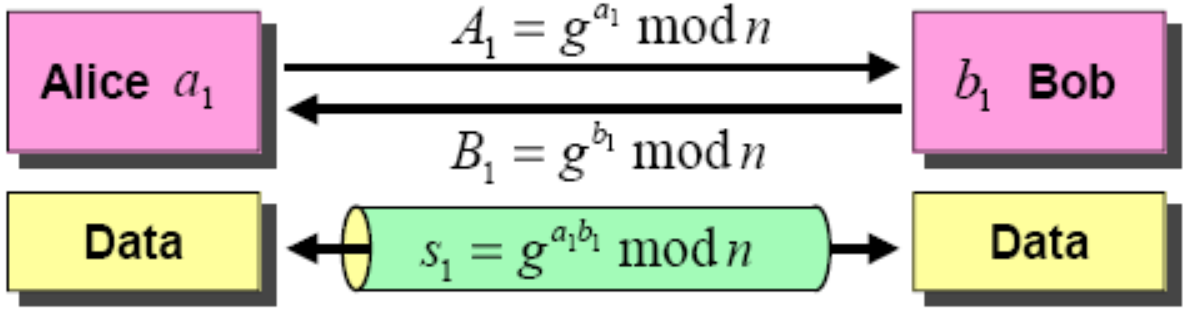
- **Msg #3**
  - L'initiator invia la sua porzione del *segreto Diffie-Hellman* più un valore random
- **Msg #4**
  - Il responder fa lo stesso inviando la sua porzione del *segreto Diffie-Hellman* più un valore random
- **Diffie-Hellman Key-Exchange** può essere competato da entrambe le parti costituendo il segreto comune condiviso
  - Questo segreto condiviso è usato epr generare una chiave di sessione simmetrica con cui saranno cifrati I restanti I messaggi del protocollo IKE

IKE Phase 1 - Main Mode
Establish a Secure Negotiation Channel

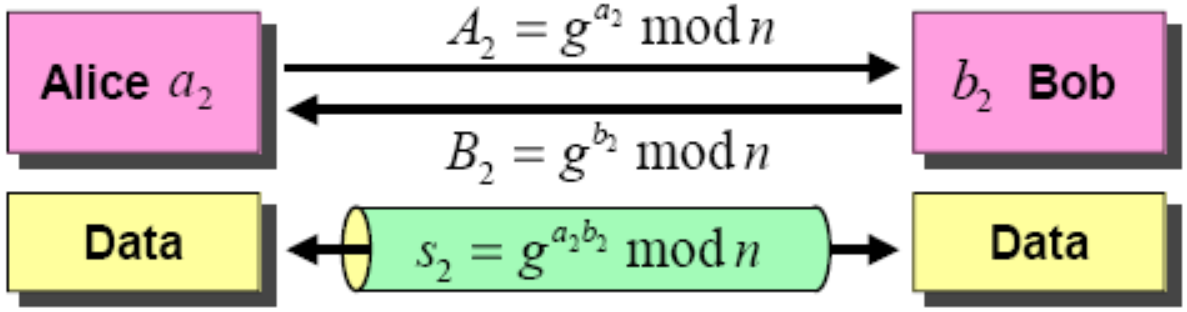
- **Msg #5**
  - L'initiator invia opzionalmente la sua identità seguita da un certificato che collega l'identità alla sua chiave pubblica.
  - Questo è seguito da un hash su tutti I campi del messaggio firmato tramite un segreto preshared o tramite una chiave privata RSA This is followed by a hash over all message
- **Msg #6**
  - Come Msg #5 ma formato e inviato dal responder
- Se l'identità di entrambi I peers è autenticata con successo si può considerare stabilita una IKE SA

The Diffie-Hellman Key-Exchange Algorithm
Perfect Forward Secrecy

- Session 1: January 26 2001



- Session 2: February 2 2001



Se la chiave s1 viene compromessa, la chiave s2 resta ancora completamente sicura!

IKE Aggressive Mode

- L'aggressive mode ottiene lo stesso risultato del main mode ma con un numero inferiore di messaggi (tre anziché sei), al prezzo però di non proteggere le identità degli interlocutori
  - dato che i payload sono scambiati prima che sia terminato lo scambio Diffie-Hellman, questi viaggiano in chiaro e non cifrati come nel caso del main mode.

IKE: Fase 2

- Dopo aver terminato la fase 1, con il main mode o con l'aggressive mode, i due interlocutori hanno creato una SA, e quindi possono procedere alla fase 2
  - Questa negoziazione avviene mediante il **Quick Mode**
  - Al contrario di quanto avviene nella fase 1, qui tutti i messaggi sono cifrati perché sono protetti dalla SA

IKE Phase 2 - Quick Mode Establish or Renew an IPsec SA

- **Encrypted Quick Mode Message Exchange**
  - Tutte le negoziazioniQuick Mode sono cifrate con un segreto condiviso
  - Chiave derivata da Diffie-Hellmann key-exchange più parametri aggiuntivi
- **Negotiation of IPsec Parameters**
  - La fase 2 Quick Mode stabilisce una IPsec SA usando il canale sicuro creato nella fase 1 IKE SA
  - I parametri di configurazione specifici per la connessione IPsec sono negoziati (AH, ESP, metodi e parametri di autenticazione/cifratura)
  - Quick Mode può essere usato ripetutamente per rinnovare IPSec SAs che stanno per scadere
- **Optional Perfect Forward Secrecy**
  - Se è richiesto perfect forward secrecy ogni consecutive Modes effettuerà un nuovo Diffie-Hellmann key-exchange