

# *Segurança em Redes IP*

*FEUP*

*MPR*

## *Introdução*

---

- ◆ Conceitos básicos
  - Criptografia
  - Segurança em redes IP
  
- ◆ IP Seguro
  - Associação de Segurança, Modos de funcionamento
  - AH, ESP, Processamento de tráfego, IKE
  - Aplicações tipo
  
- ◆ IP Seguro nas Comunicações Móveis

---

## Conceitos Básicos

### (De)Cifrar

---

- ◆ Cifrar: mensagem aberta → mensagem cifrada
  - Função matemática + chave
- ◆ Decifrar: mensagem cifrada → mensagem aberta
  - Função matemática + chave
- ◆ Exemplo c/ DES
  - » Mensagem plana
    - *Encryption can make UNIX more secure*
  - » Chave
    - *nosmis*
  - » Mensagem cifrada
    - *M-itM-@g^B^?^B?^NM-XM-vZIM-U\_h^X^\$kM-^sI^M-fIM-^ZM-jM-gBM-6M-  
>^@M-”^M-^JM-^JM-7M--M-^T*

( ASCII; caracter de controlo precedido por ^; it mais significativo activo → M- )

## *Métodos de Cifra*

---

### ◆ Chave simétrica

- » chave única para cifrar e decifrar → chave simétrica
  - DES\_CBC (Data Encryption Standard, Cipher Block Chaining). Chave de 56 bits
  - IDEA (International Data Encryption Algorithm). Chave de 128 bits
  - 3DES – 3 chaves de 56 bits (1ª pode ser igual a 3ª)

### ◆ Chave assimétrica

- » 2 chaves: pública e privada → chave assimétrica
  - RSA (Rivest, Shamir, Adleman) – chaves longas
- » Em redes,
  - chaves assimétricas normalmente usadas para gerar chaves simétricas

## *Resumo de Mensagem / Assinatura Digital*

---

### ◆ Resumo de mensagem

- » Pequeno valor (128 a 512 bit) obtido a partir de uma mensagem
- » Função de Hash
- » Algoritmos comuns
  - MD5 (Message Digest 5). 128 bit
  - SHA (Secure Hash Algorithm). 160 bit

### ◆ Assinatura digital

- » Resumo de mensagem cifrado com chave assimétrica (a privada)
  - Ex. MD5+RSA, SHA+RSA
- » Resumo de mensagem cifrado com chave simétrica
  - Ex. Keyed MD5: [chave,mensagem,chave] → MD5 → assinatura ; mais usado em redes

### ◆ Com assinatura digital consegue-se verificar

- » Integridade → saber se mensagem foi modificada
- » Autenticidade → saber quem assinou a mensagem

## *Problemas de Segurança Frequentes*

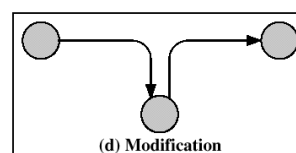
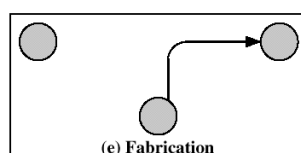
---

- ◆ Spoofing
- ◆ Session Hijacking
- ◆ Eaves dropping
- ◆ Man-in-the-Middle

## *Spoofing*

---

- » Datagrama IP
  - ◆ Cabeçalho → endereço origem, endereço destino
  - ◆ Dados → informação de camadas superiores
- » Router
  - ◆ Encaminha datagramas. Desconhece detalhes da arquitectura da rede
- » Rota de um datagrama
  - ◆ Não é controlada por origem nem destino
  - ◆ Datagramas do mesmo fluxo podem seguir rotas diferentes. Qualquer rota é legítima
- » Problema
  - ◆ Cabeçalhos de datagrama IP → facilmente gerados/alterados em qualquer máquina
  - ◆ Ex. Acesso a serviços configurados por endereços de rede. NFS



## Roubo de Sessão. Monitoração de Tráfego

### ◆ Roubo de sessão (Session Hijacking)

- Ex. Roubo de mail. Sobre ligação TCP/IP
- Cliente estabelece ligação TCP/IP com servidor de mail. Autentica-se no servidor
- Usurpador entra, depois da autenticação e
  - ◆ Termina ligação com o servidor
  - ◆ Continua ligação com o cliente, recebendo o mail
- Identificação inicial -/→ segurança durante toda sessão

### ◆ Monitoração de tráfego (Eavesdropping)

- LANs Ethernet → pacotes disponíveis em todos os nós da rede ( hubs, cabo )
- Carta de rede
  - ◆ Modo normal → copia tramas que lhe forem endereçadas
  - ◆ Modo promíscuo → copia todas as tramas. Outros nós não sabem da sua existencia.
- Acesso a toda a informação

## Man-in-the-Middle

### ◆ Solução para problemas anteriores

→ utilização técnicas de encriptação ou de autenticação

### ◆ Cifrar, autenticar

- Algoritmos conhecidos + chaves
- Chaves = bits de informação

### ◆ Problemas!

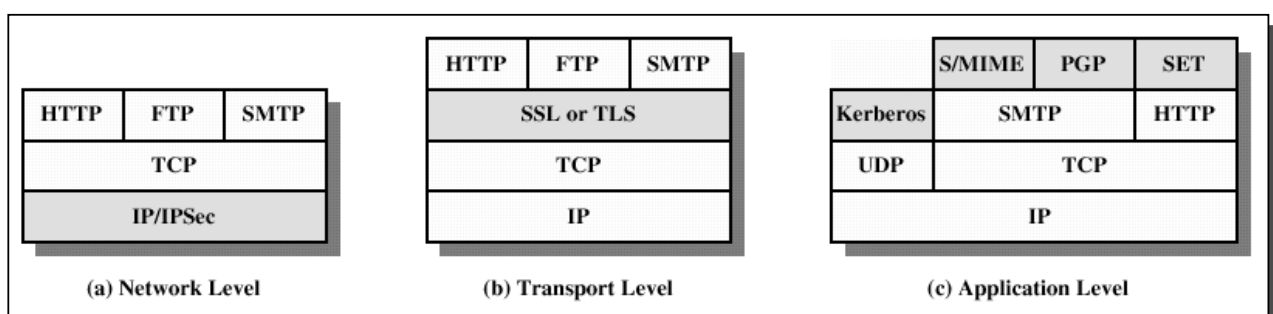
- Troca não protegida de chaves → interceptação → ataque *Man-in-the Middle*
- Os comunicadores assumem que a comunicação é segura, mas
- a comunicação pode ser espiada/ adulterada

## Requisitos de Segurança em Redes

- » Autenticação: O parceiro da comunicação deve ser o verdadeiro
- » Confidencialidade: Os dados transmitidos não devem ser espiados
- » Integridade: Os dados transmitidos não devem ser alterados

## Segurança na Pilha TCP/IP

- ◆ Aplicação
  - » Kerberos → sistema de autenticação global. Baseado em bilhetes. Chave privada (DES)
  - » PGP (Pretty Good Privacy). Usado com mail para (de)cifrar mensagens. Assinaturas digitais
  - » S/MIME → Cifra de mensagens + assinaturas electrónicas
  - » SSH → Secure Shell. Substituto seguro do rsh / rlogin
- ◆ Transporte
  - » TLS (Transport Layer Security). Nome antigo → SSL. Segurança de sessões HTTP
- ◆ Rede
  - » IPSec



## *A Importância da Camada de Rede*

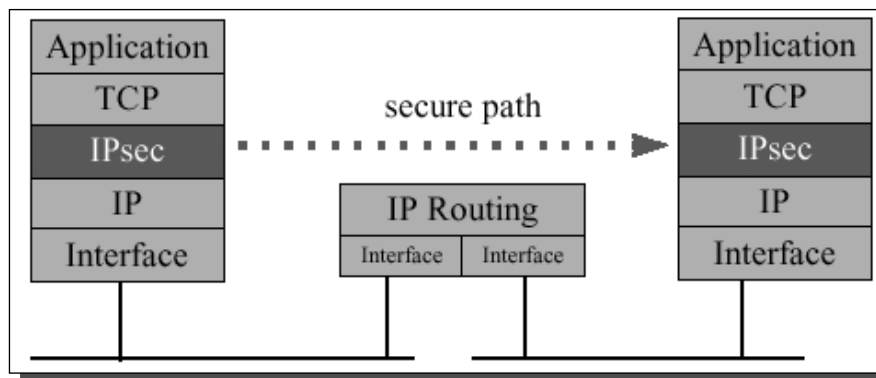
---

- ◆ Comunicação TCP/IP modelizada em camadas. Cada camada
  - Endereça um problema
  - Oferece serviços ao nível superior
- ◆ Camada de rede, em redes IP
  - Homogénea, universal. Tendência →
    - ◆ todas as aplicações sobre IP
    - ◆ IP sobre todas as redes
- ◆ Em teoria, IP seguro → rede segura → comunicações seguras
  - Mesmo as móveis ...
  
- ◆ Mas a utilização da segurança de aplicação tem vindo a crescer!
  - https, ssh

---

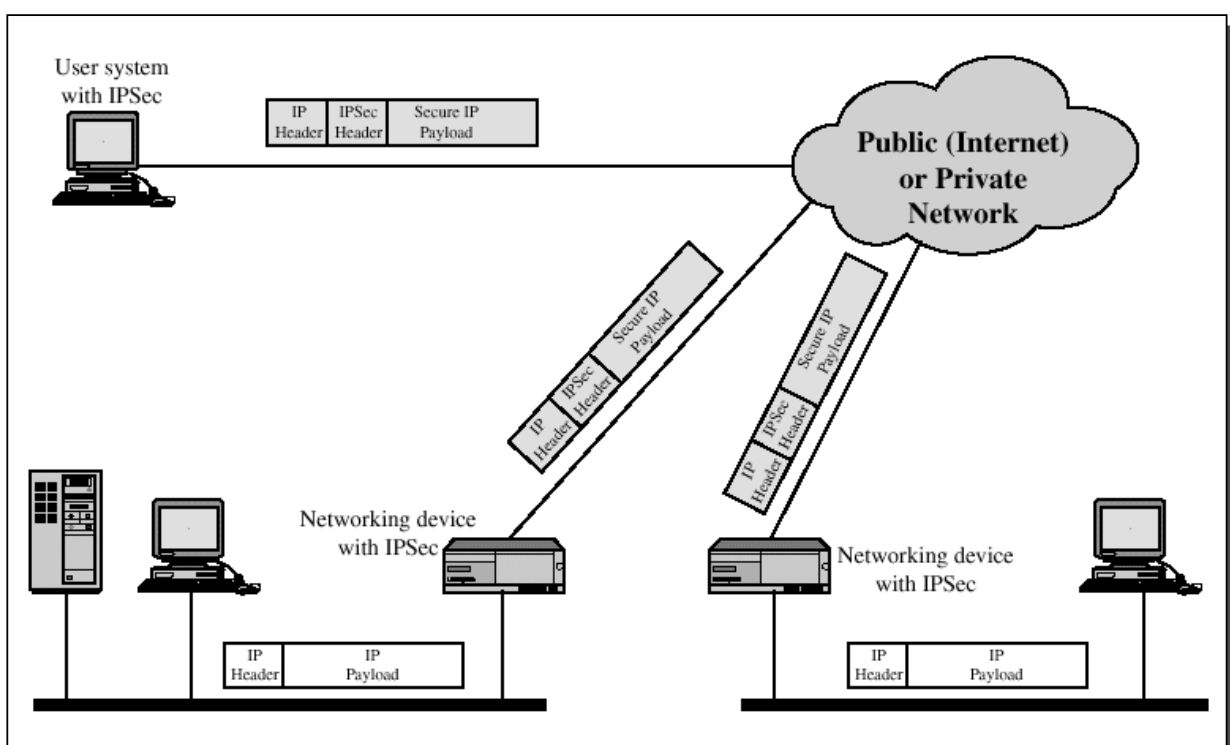
## *IP Seguro*

## IPSec



- ◆ Arquitectura segura para IP
  - » Aberta, normalizada
  - » Autenticação e integridade dos dados
  - » Protecção contra repetição de datagramas
  - » Algoritmos de cifra actuais
  - » Criação segura de chaves de segurança, com duração limitada

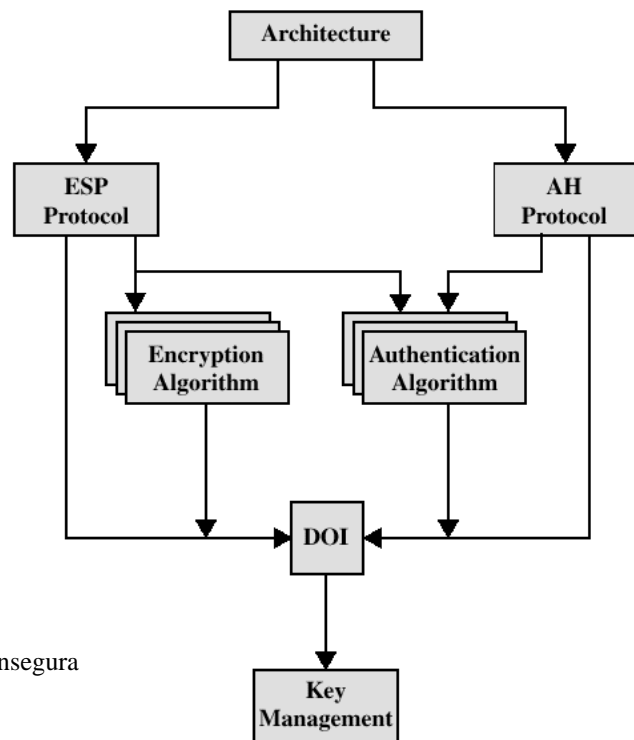
## Cenário de Utilização de IPSec



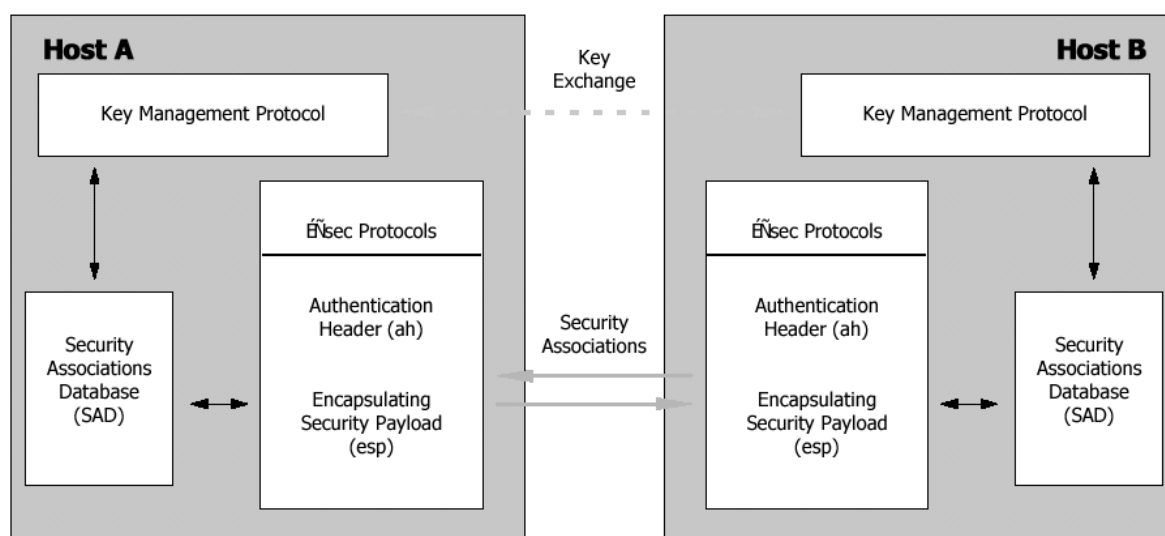


## IPSec no IETF

- ◆ Grupo de trabalho do IETF →
  - » IP security (IPSec) protocol suite
  - » RFCs 2401, 2412, and 2451
- ◆ IPSec
  - » Compatível com IPv4
  - » Obrigatório no IPv6
  - » Transparente para utilizadores
  - » Escalável
- ◆ Quando usado,
  - » Protege comunicações, de todas as aplicações e todos os utilizadores
  - » Podem ser construídas VPN (Virtual Private Network) →
    - Rede privada segura sobre rede pública insegura
    - Estabelecida e terminada dinamicamente



## Arquitectura



## *Associação de Segurança*

---

### ◆ SA – Security Association

- Ligação lógica unidireccional
- Funcionamento (exclusivo) em modo túnel ou modo transporte
- Suporta (apenas) 1 protocolo de segurança (ESP ou AH)

### ◆ Identificado por 3 valores

- SPI, Security Parameter Index → 32 bit
- Endereço IP de destino (só endereços unicast)
- Protocolo de segurança → AH ou ESP

- » 1 ligação bidireccional → estabelecimento de 2 SAs
- » Bidireccional c/ utilização de AH e ESP → estabelecimento de 4 SAs

## *Modos de Utilização de uma SA - Transporte, Túnel*

---

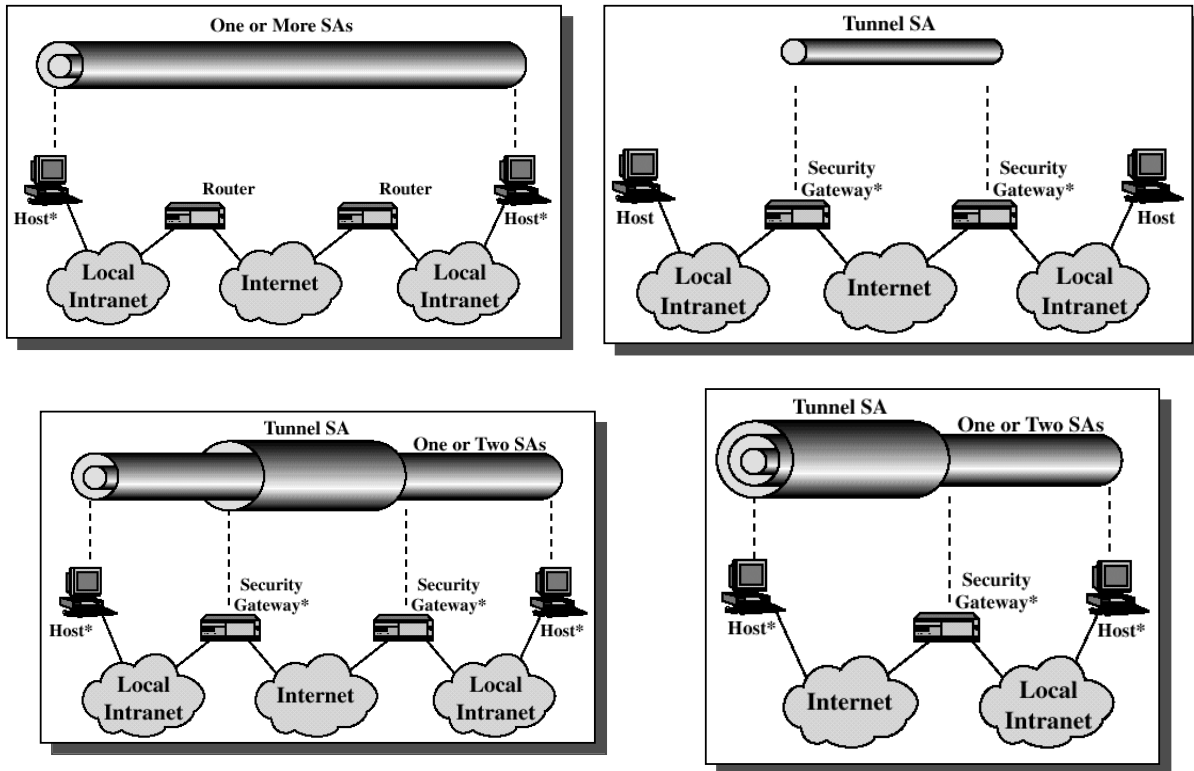
### ◆ Modo Transporte

- » Cabeçalho do datagrama IP é mantido
- » Usados endereços originais (globais)
- » Alguns campos do cabeçalho não são protegidos

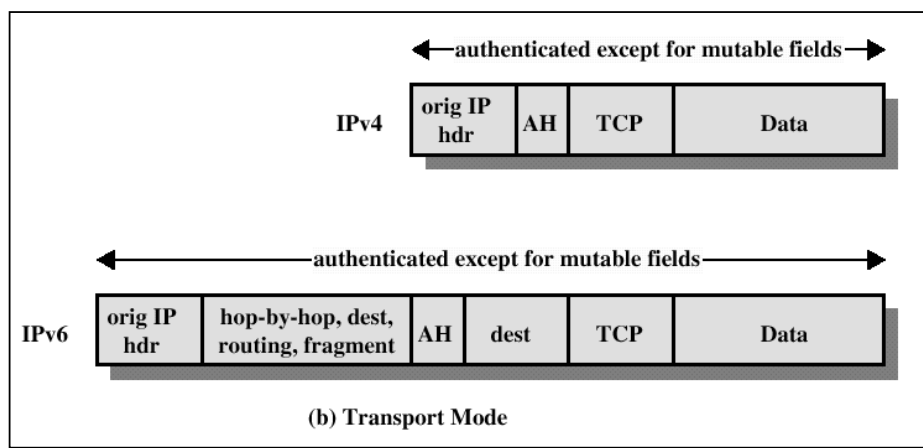
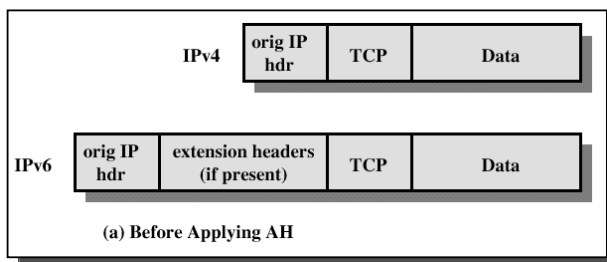
### ◆ Modo Túnel

- » Datagrama original encapsulado dentro do novo pacote
- » Protege completamente o datagrama original
- » Datagrama original pode ter endereços internos (ilegais)

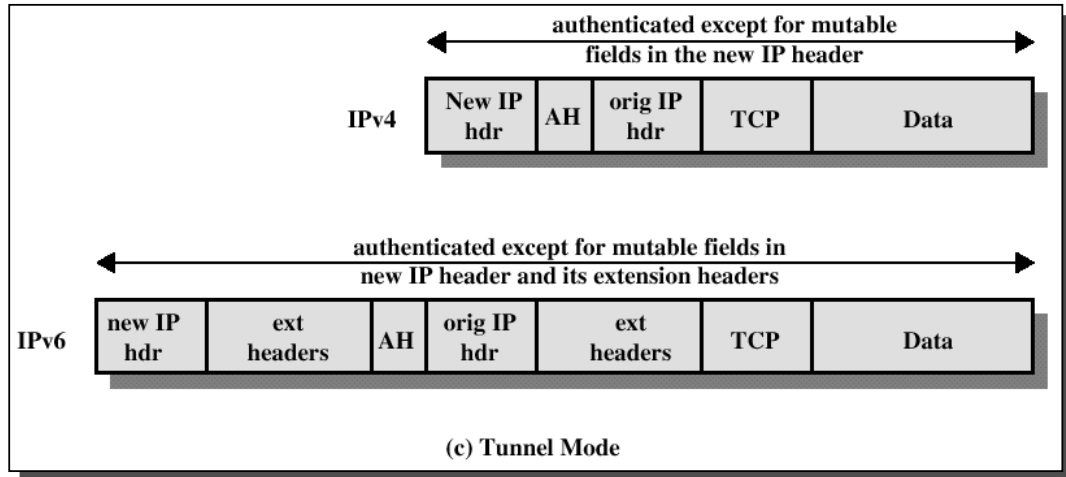
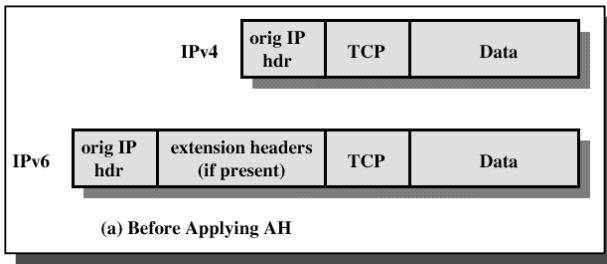
# Associações de Segurança



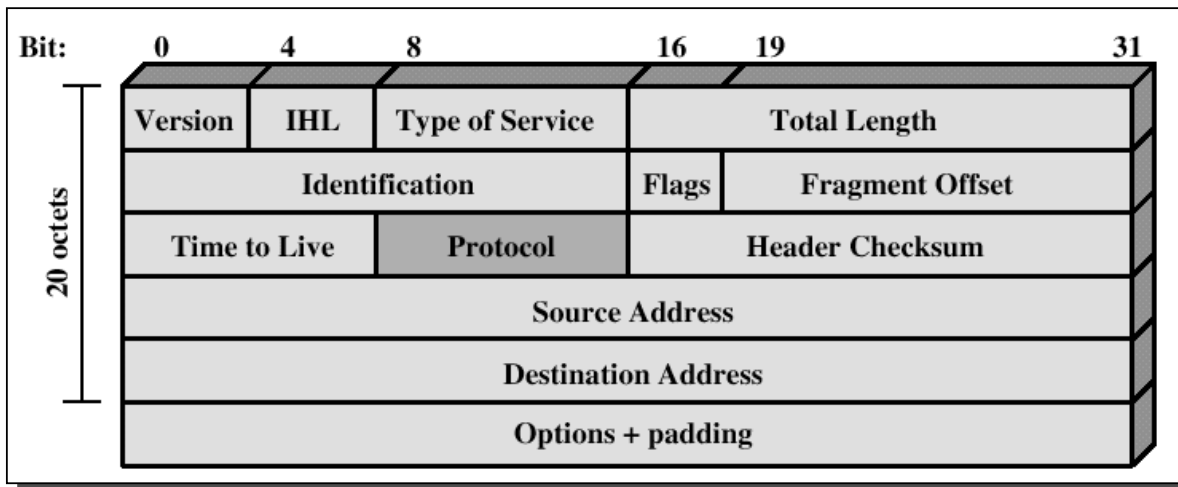
# AH, Authentication Header – Modo Transporte



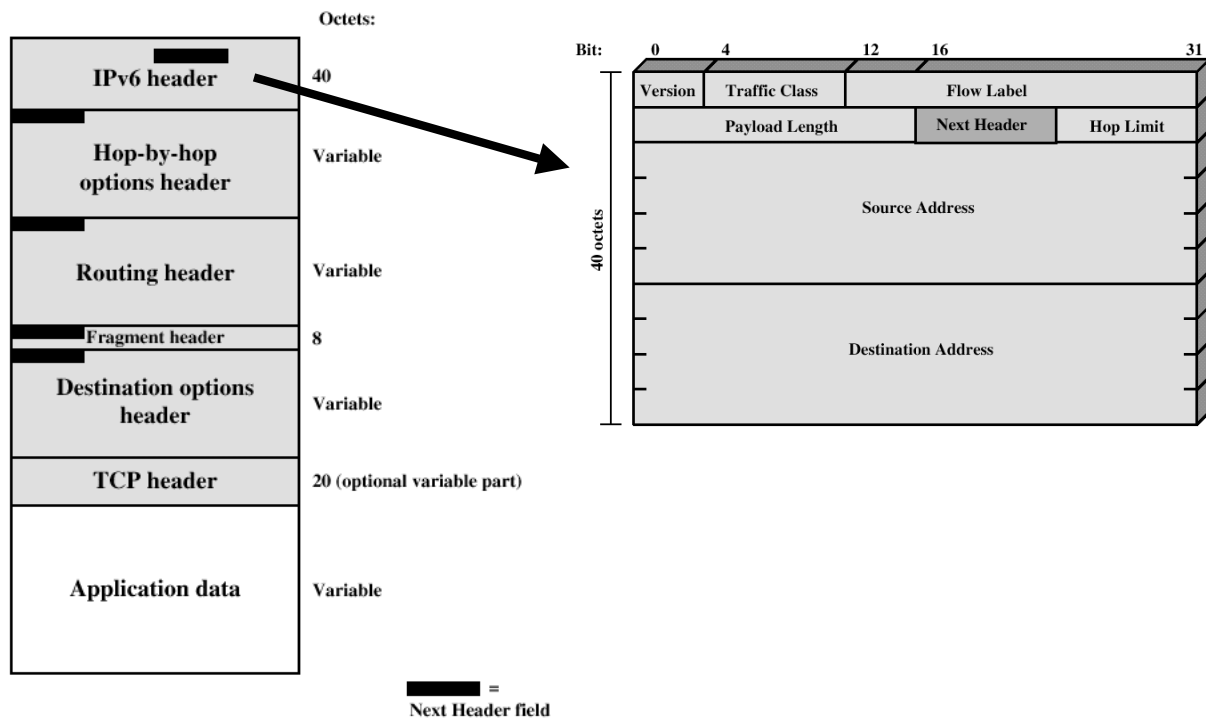
# AH, Authentication Header – Modo de Túnel



# Cabeçalho IPv4

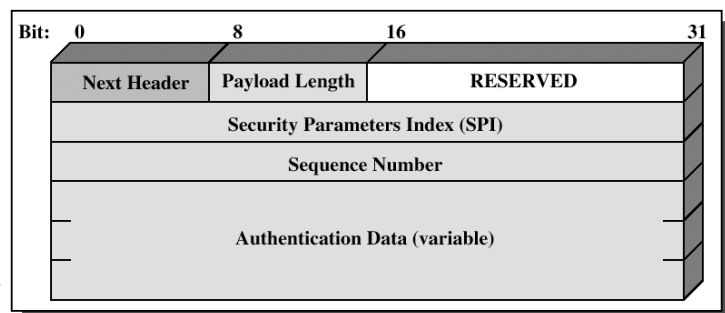


## Pacote e Cabeçalho IPv6



## Cabeçalho AH

- ◆ Protocolo 51
- ◆ Campos
  - » Tipo do protocolo seguinte
    - Ex. TCP (6), ESP (50)
  - » Comprimento cabeçalho
    - Palavras 32 bits
  - » SPI
    - Identificador do grupo de segurança
  - » Número de sequência
  - » Assinatura digital
    - Cálculo do resumo do datagrama
      - ◆ Campos variáveis excluídos (ex. TTL)
      - ◆ Algoritmos de hash MD5, SHA
    - Utilização de uma chave secreta *comum*
    - RFC2403, RFC2404



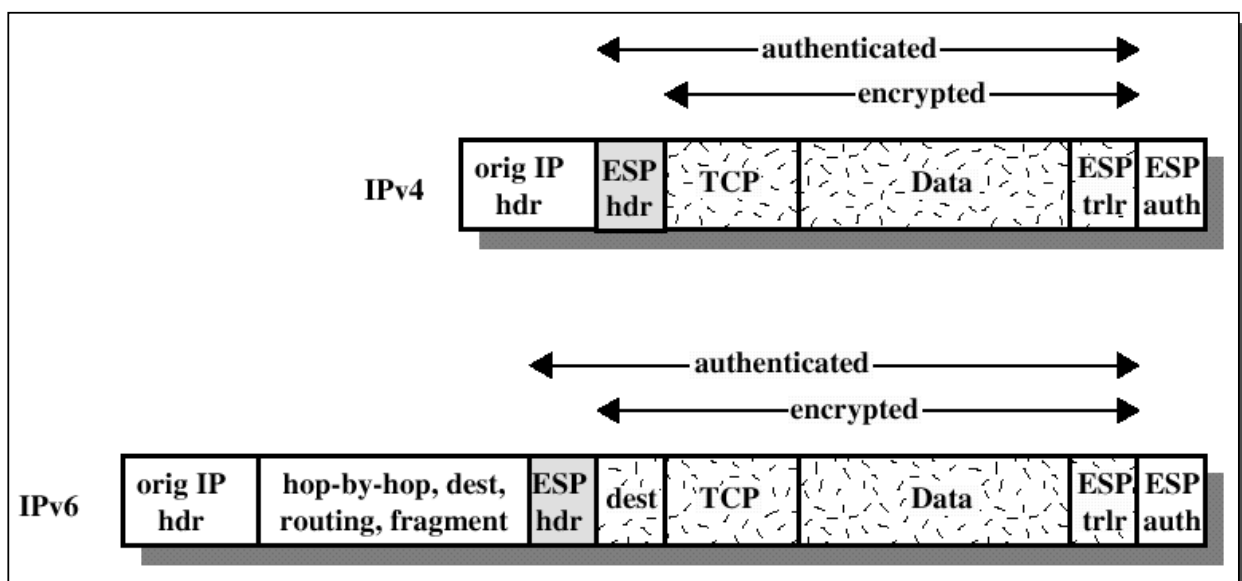
## *AH, Authentication Header*

---

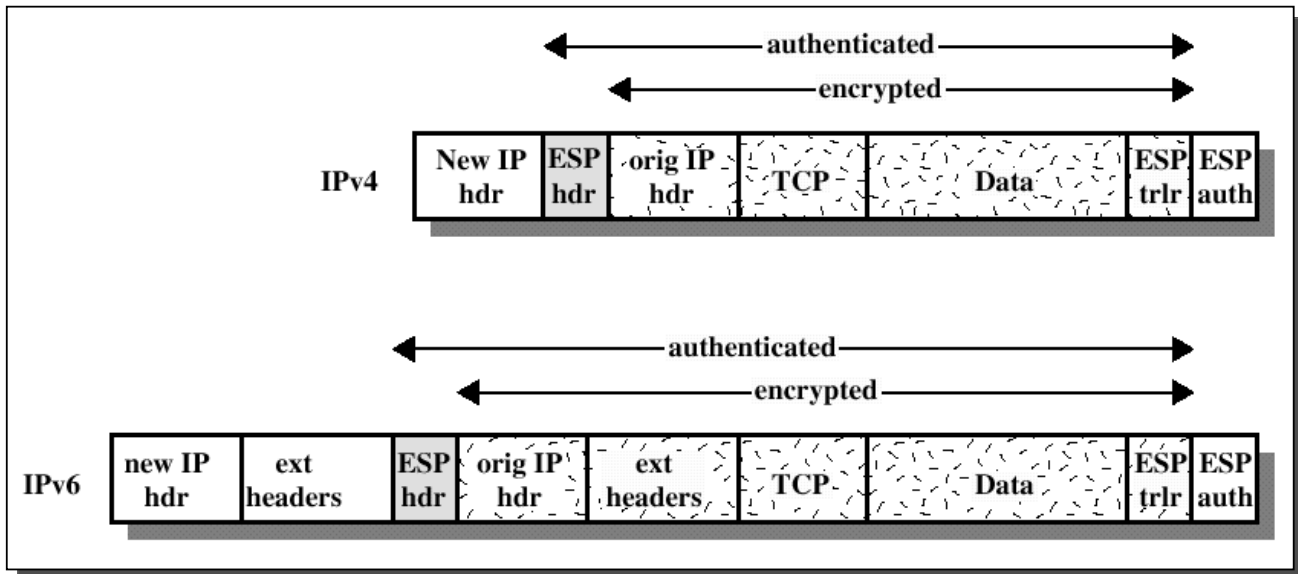
- ◆ Permite
  - » Autenticar o cabeçalho do datagrama
  - » Verificar a integridade dos dados
  
- ◆ Conteúdo do pacote não é cifrado
  
- ◆ Campos variáveis são excluídos do cálculo do resumo
  - » TOS, Flags, TTL, checksum, ...
  
- ◆ 24 octetos adicionados por datagrama

## *ESP, Encapsulating Security Payload – Modo Transporte*

---

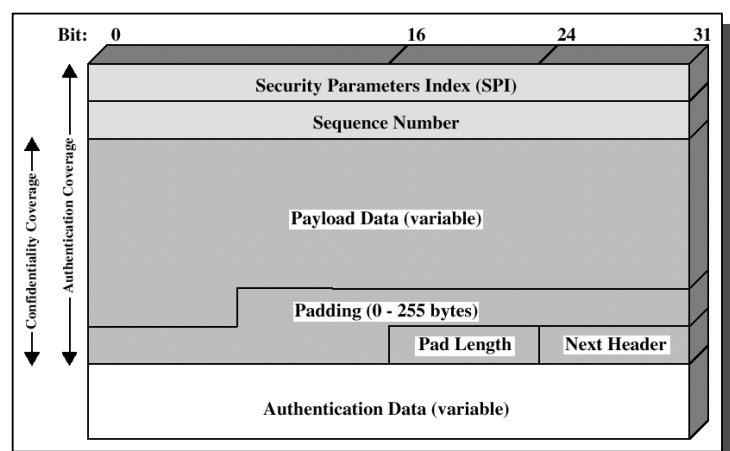


# ESP, Encapsulating Security Payload – Modo Túnel



## Cabeçalho ESP

- ◆ Protocolo 50
- ◆ Não cifrado
  - » SPI – Security Parameter Index
    - ◆ Grupo de segurança
  - » Número sequência
  - » Assinatura digital (opcional)
    - Calculada sobre os outros campos do cabeçalho ESP
- ◆ Cifrado
  - » Dados
    - (ex. Cabeçalho TCP + dados)
  - » *Padding*
    - Para algoritmos de cifra de comprimentos pre determinados
  - » Comprimento do *padding*
  - » Protocolo seguinte



## Encapsulating Security Payload (ESP)

---

- ◆ Cifra o conteúdo do pacote. Segredo (chave) partilhado
  - Algoritmos de cifra: DES, IDEA, 3DES, etc
  
- ◆ Opcionalmente, permite
  - » Autenticar parte do cabeçalho do datagrama
  - » Verificar a integridade dos dados
  
  - » Com técnicas de autenticação iguais às do AH

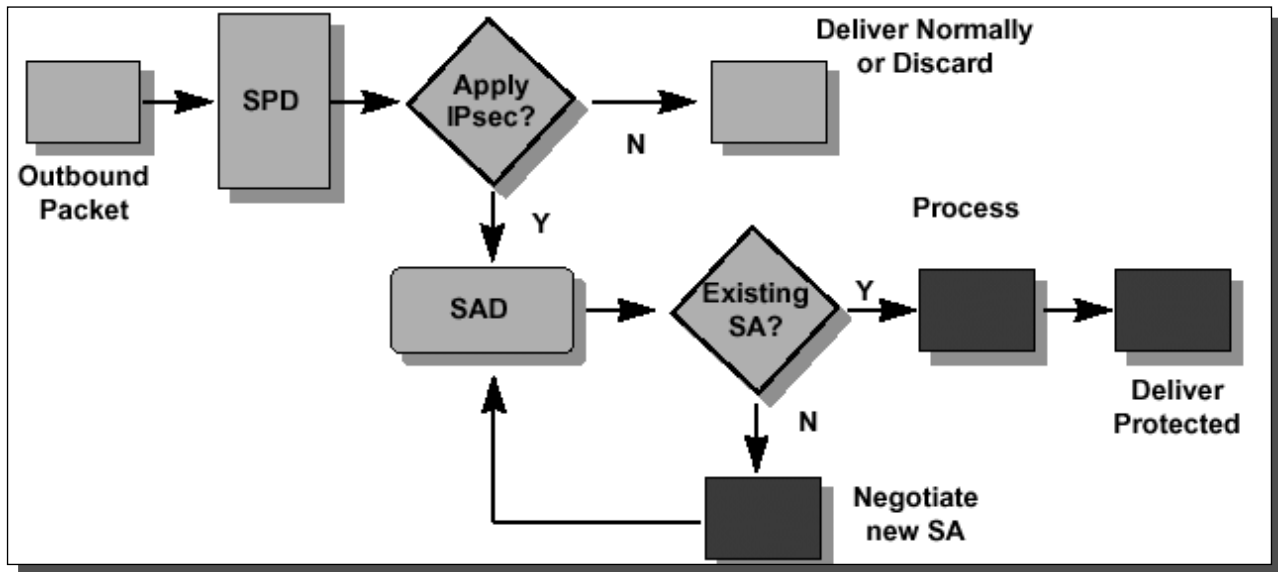
## Bases de Dados de SAs

---

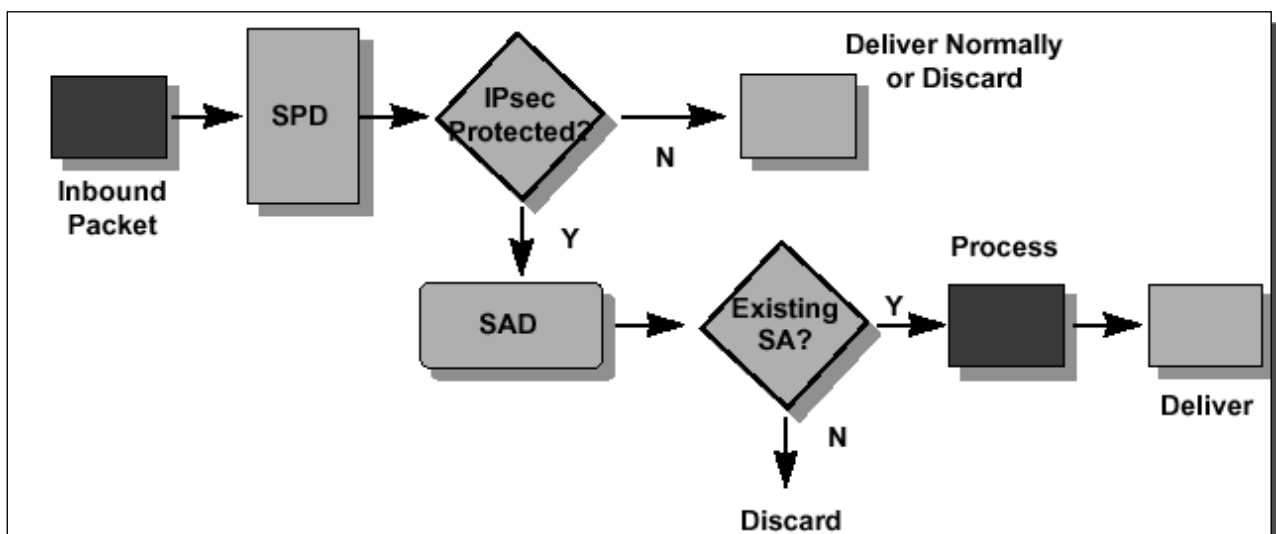
- » 2 bases de dados por cada interface IPSec → SPD, SAD
  
- » SPD, Security Policy Database
  - Lista ordenada de políticas de segurança. Selecção do tráfego IP a
    - 1) Eliminar;
    - 2) Processar pelo IPSec;
    - 3) Não processar por IPSec
  - Políticas descritas com base em
    - ◆ Tipo de endereços: origem, destino
    - ◆ Tipo de tráfego: inbound (de entrada na interface), outbound (de saída)
  - Políticas segurança ↔ Regras de filtragem (de pacotes) nas firewalls
  
- » SAD, Security Associations Database
  - Informação sobre as SAs estabelecidos
    - ◆ Protocolo, algoritmos de assinatura e encriptação



## *Processamento de Tráfego Outbound*



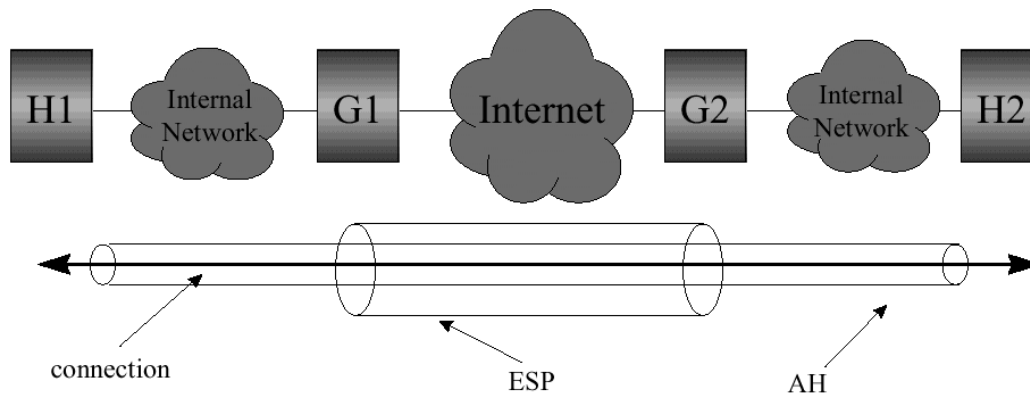
## *Processamento de Tráfego Inbound*



## *Aplicações Tipo do IPSec – VPN*

---

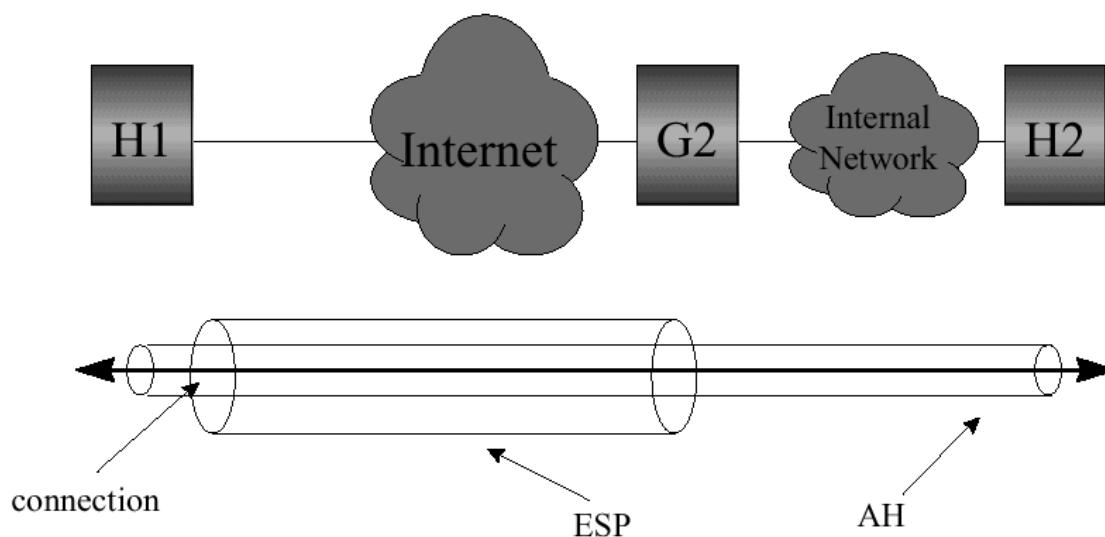
- ◆ VPN c/ segurança extremo a extremo
- ◆ ESP encripta dados sobre a Internet pública
  - Pode ser usado em modo túnel
- ◆ AH assegura integridade dos dados extremo a extremo



## *Aplicações Tipo do IPSec – Guerreiro da Estrada*

---

- ◆ Utilizador liga-se à empresa através da Internet pública
- ◆ ESP pode ser usado em modo túnel



## *Combinação de SAs*

---

- ◆ Número of SAs cresce rapidamente
  - » Número de ligações
  - » 1 par de SAs para cada ligação
  - » Combinação de protocolos IPSec (AH, ESP, AH sobre ESP)
  - » Modos de funcionamento
  
  - » Gateway VPN → centenas de SAs
  
- Gestão manual de SAs → complexa, impraticável
- Necessidade de mecanismos para
  - » Negociar, estabelecer e terminar SAs

## *IKE - Internet Key Exchange*

---

- ◆ Protocolo usado para
  - » Estabelecer e terminar SAs
    - Protocolos, algoritmos e chaves
  - » Autenticar as partes
  - » Gerir as chaves trocadas
  
- ◆ Sobre UDP, Porta 500. RFC 2409

## *IKE*

---

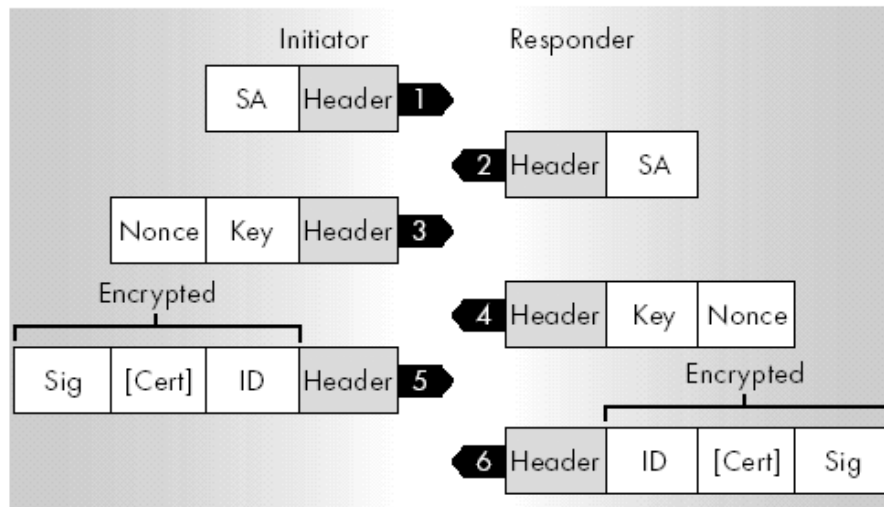
- ◆ Fases de negociação
  - » Fase 1 → partes estabelecem 1 canal seguro (SA IKE), em 3 passos
    - ◆ Negociação de tipos de resumo e algoritmos de cifra a usar
    - ◆ Troca de chaves públicas (método Diffie-Hellman)
      - Chaves de cifra comuns obtidas a partir de chaves públicas
    - ◆ Verificação de identidade do parceiro
  - » Fase 2 → negociação de SAs genéricas, através do SA IKE
- ◆ Modos de negociação
  - » Fase 1
    - Modo principal
    - Modo agressivo: mais simples, mais rápido; não fornece protecção de identidade
  - » Fase 2
    - Modo rápido

## *Modo Principal*

---

- ◆ Objectivo
  - » Negociar algoritmos de autenticação/confidencialidade, hashes e chaves
- ◆ 3 trocas bidireccionais entre as 2 partes
  1. Negociação de algoritmos básicos e hashes
  2. Troca de chaves públicas
    - método de Diffie-Hellman
    - Troca de nonces → número aleatório que a outra parte deve assinar e retornar
  3. Verificação das identidades
- ◆ As partes usam o valor de Diffie-Hellman recebido para
  - » Obter chave de geração de futuras chaves
  - » Gerar chave de autenticação
  - » Gerar chave de encriptação, usada no IKE SA

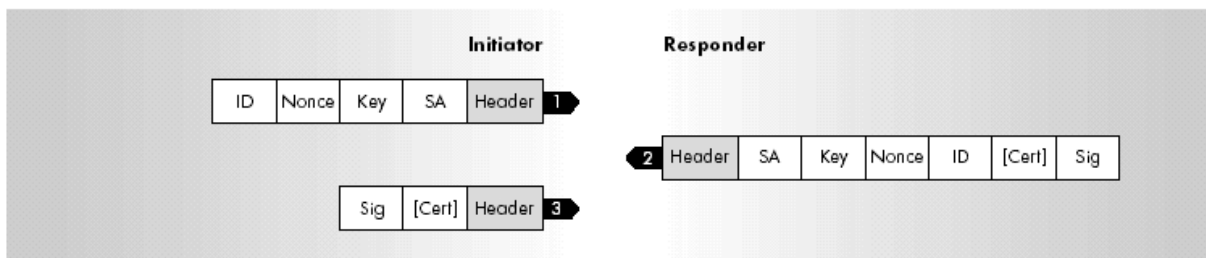
## Modo Principal



Cert: a certification payload  
 ID: an identification payload  
 Key: a key exchange payload  
 Nonce: a nonce payload  
 SA: a Security Association/proposal payload  
 Sig: a signature payload

## Modo Agressivo

- » 2 trocas
- » Partes trocam informação de identidade não cifrada encriptação
- » É possível conhecer a identidade dos comunicadores. É mais rápido

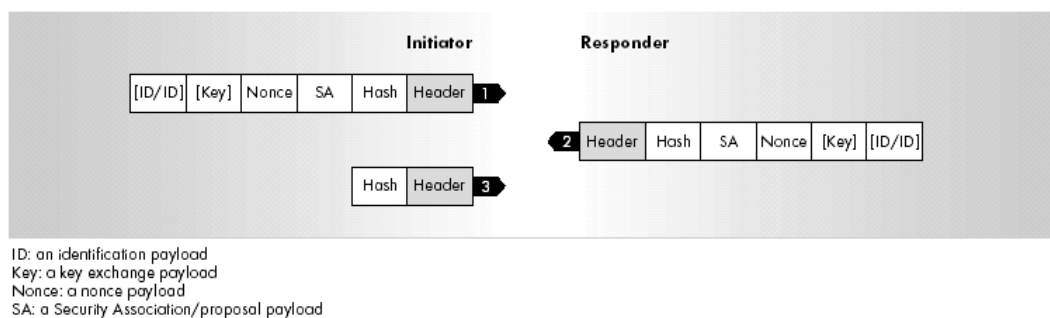


Cert: a certification payload  
 ID: an identification payload  
 Key: a key exchange payload  
 Nonce: a nonce payload  
 SA: a Security Association/proposal payload  
 Sig: a signature payload

## Modo Rápido

---

- » 2 objectivos: negociar nova SA; gerar chaves frescas
- » mensagens encriptadas
- » 3 pacotes
  - ◆ Initiator envia info de SA, nonce, hash sobre tudo
  - ◆ Responder envia SA, seu nonce, e hash feito também sobre nonce do initiator
  - ◆ Initiator envia hash sobre os 2 nonces
- » Initiator e Responder
  - ◆ Fazem hash sobre nonces, SPI, valores dos protocolos; com chave de geração
  - ◆ Novo hash → chave da nova SA



## Negociação de 1 SA

---

- ◆ Para gerar nova SA
  - » Initiator envia mensagem de Modo Rápido, protegida por IKE SA
- ◆ Uma negociação de SA resulta em 2 SAs
  - » Entrada e saída (para o Initiator)
  - » SAs são sempre formadas aos pares
- ◆ Pares têm os mesmos parâmetros
  - » Chaves, algoritmos de autenticação e encriptação
  - » SPI são diferentes!

## *IKE Authentication Methods*

---

Authentication method	How authentication is performed	Advantages	Disadvantages
Pre-shared keys	By creating hashes over exchanged information	<ul style="list-style-type: none"> <li>• Simple</li> </ul>	<ul style="list-style-type: none"> <li>• Shared secret must be distributed out-of-band prior to IKE negotiations.</li> <li>• Can only use IP address as ID</li> </ul>
Digital signatures (RSA or DSS)	By signing hashes created over exchanged information	<ul style="list-style-type: none"> <li>• Can use IDs other than IP address</li> <li>• Partner certificates need not be available before</li> </ul>	<ul style="list-style-type: none"> <li>• Requires certificate operations (inline or out-of-band)</li> </ul>
RSA public key encryption	By creating hashes over nonces encrypted with public keys	<ul style="list-style-type: none"> <li>• Better security by adding public key operation to DH exchange</li> <li>• Allows ID protection with aggressive mode</li> </ul>	<ul style="list-style-type: none"> <li>• Public keys (certificates) must be available before IKE negotiations</li> <li>• Performance-intensive public key operations</li> </ul>
Revised RSA public key encryption	Same as above	<ul style="list-style-type: none"> <li>• Same as above</li> <li>• Fewer public key operations by using an intermediate secret</li> </ul>	<ul style="list-style-type: none"> <li>• Public keys (certificates) must be available before IKE negotiations</li> </ul>

---

## *MIPv6 Seguro*

## Ameaça Principal e Soluções

- ◆ Ameaça principal → *Binding Update* falso
  
- ◆ Solução
  - » Túnel bidireccional obrigatório
  - » Protecção dos *Binding Updates* enviados ao HA
  - » Protecção dos *Binding Updates* enviados aos CNs

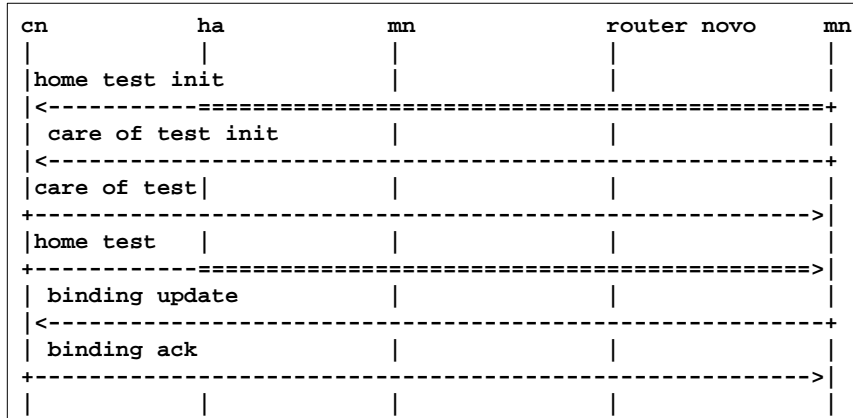
## *Binding Updates, MN ↔ HA*

- ◆ MN e HA
  - » Usam Associações de Segurança
    - para proteger integridade e autenticidade de
    - *Binding Updates, Binding Acknowledgements*
  - » ESP em modo de transporte, com autenticação
  
- ◆ Gestão automática de chaves de segurança com IKE



## Binding Update, MN $\leftrightarrow$ CN

- ◆ Utilização do procedimento Return Routability
- ◆ Complexo!



## Terminologia – Kcn, Nonces, Cookies, Tokens

- ◆ *Kcn*
  - » Pertence ao CN, segredo; usado para gerar *keygen token*
- ◆ *Nonce*
  - » CN gera nonces regularmente; identificado por índice
- ◆ *Cookies*
  - » Enviado por MN ao CN, devolvido por CN ao MN; não repetidos
  - » *Home init cookie* - enviado em Home Test Init; devolvido em Home Test
  - » *Care-of init cookie* - env em Care-of Test Init, devolv em Care-of Test
- ◆ *Tokens*
  - » Valores enviados por CN ao MN
  - » *home keygen token* – em Home Test, via HA
  - » *care-of keygen token* – em Care-of Test message

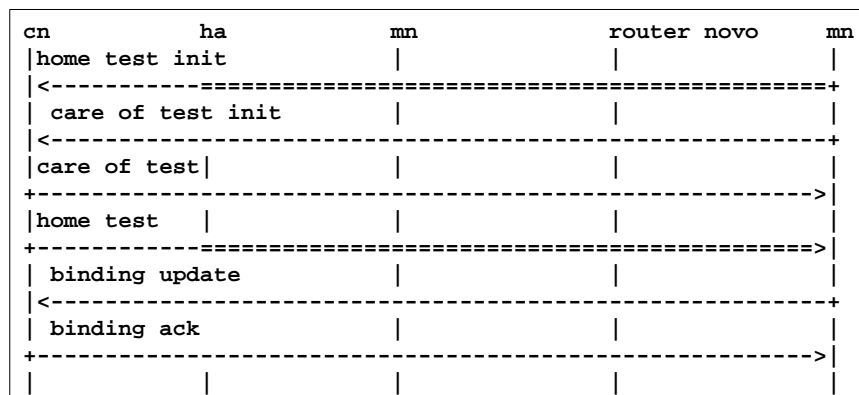
## *Funções Criptográficas*

---

- ◆ Função geradora de valores Hash → SHA1
- ◆ Códigos de autenticação de mensagens (MAC), gerados com
  - » HMAC\_SHA1(K,m) → MAC baseado em mensagem m e chave K.

## *Procedimento de Return Routability*

---



## *Procedimento de Return Routability*

---

- » Home Test Init
  - ◆ Source Address = home address; Destination Address = CN
  - ◆ Parâmetros → home init cookie; retornado por CN
  - ◆ Quando recebe Home Test Init, CN gera
    - *home keygen token* := First (64, HMAC\_SHA1 (Kcn, (home address | nonce | 0)))
- » Home Test
  - ◆ Source Address = CN; Destination Address = home address
  - ◆ Parâmetros: home init cookie, home keygen token, home nonce index
- » Care-of Test Init
  - ◆ Source Address = care-of address; Destination Address = CN
  - ◆ Parâmetros → care-of init cookie; retornado por CN
  - ◆ Quando recebe Care-of Test Init message, CN gera
    - *care-of keygen token* := First (64, HMAC\_SHA1 (Kcn, (care-of address | nonce | 1)))
- » Care-of Test
  - ◆ Source Address = CN; Destination Address = care-of address
  - ◆ Parâmetros: care-of init cookie, care-of keygen token, care-of nonce index
- » Quando MN recebe *Home Test* e *Care-of Test*
  - ◆ usa os 2 tokens para formar a chave de binding Kbm:
    - *Kbm* = SHA1 (home keygen token | care-of keygen token)
  - ◆ Usa Kbm para autenticar *Binding Update* e *Binding Ack*

## *Mensagens de Binding*

---

### ◆ *Binding Update*

- » Source Address = care-of address
- » Destination Address = CN
- » Parâmetros
  - home address; sequence number
  - home nonce index; care-of nonce index
  - First (96, HMAC\_SHA1 (Kbm, (care-of address | correspondent | BU)))

### ◆ *Binding Acknowledgement*

- » Source Address = CN
- » Destination Address = care-of address
- » Parâmetros:
  - sequence number (within the Binding Update message header)
  - First (96, HMAC\_SHA1 (Kbm, (care-of address | correspondent | BA)))
- » *Binding Ack* contém mesmo número sequência que *Bind Update*

