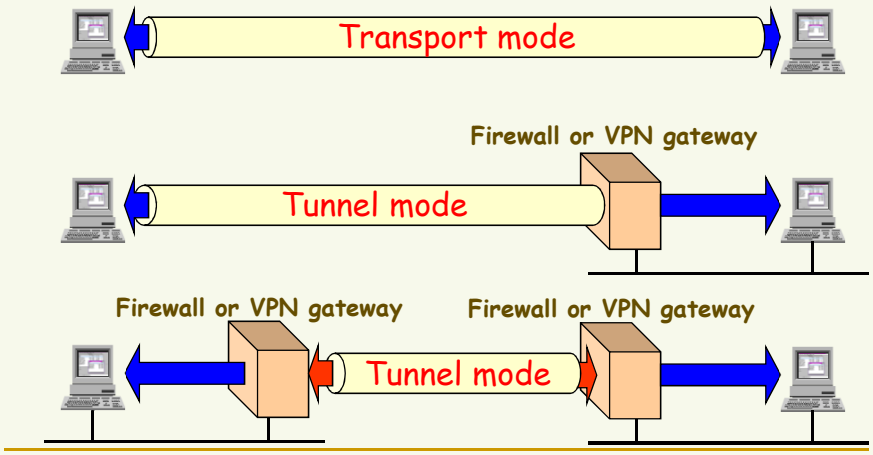
IPSec (IP Security)

IPSec (IP Security): Goals

- Protect IP traffic
  - Datagram confidentiality, datagram integrity control
- Two operational modes:
  - Transport mode
    - Uses original (cleartext) IP headers
  - Tunnel mode
    - Encapsulates (possibly encrypting) original IP header

IPSec:
Operational scenarios

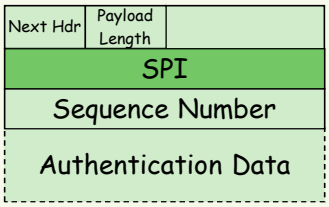


IPSec:
Mechanisms

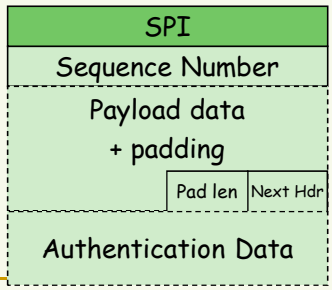
- Security Associations (SA, RFC 4301)
  - Security policies, mechanisms and crypto parameters used to secure the communication between a pair of hosts
  - Security Parameter Index (SPI)
    - SA identifier
    - Indexes the SA that should be used to validate an IPSec datagram
- Extra optional fields for the IP header
  - Authentication Header (AH, RFC 4302)
    - Has an SPI
    - Keyed hash (MAC) of the whole IP datagram
  - Encapsulating Security Payload (ESP, RFC 4303)
    - Has an SPI
    - Authenticated cryptogram of the IP datagram payload

IPSec:
AH and ESP mechanisms

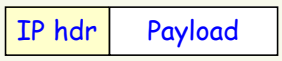
- AH goals
  - Connectionless integrity
  - Data origin authentication
  - Optional anti-replay service



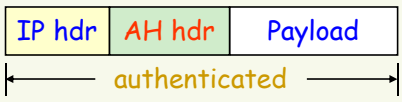
- ESP goals
  - Confidentiality (encryption)
  - Limited traffic flow confidentiality
  - Optional connectionless integrity
  - Optional data origin authentication
  - Optional anti-replay service



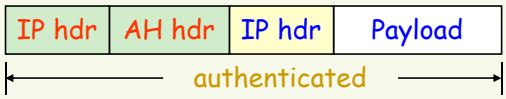
IPSec:
Authentication Header (AH)



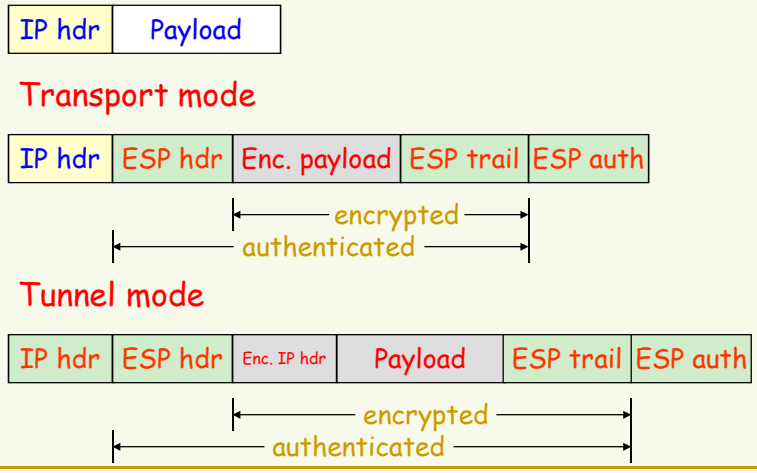
Transport mode



Tunnel mode



IPSec:
Encapsulating Security Payload (ESP)



IPSec:
Cryptographic algorithms (RFC 4305)

- AH
  - Must be implemented (mandatory)
    - HMAC-SHA-1
  - Should/may be implemented (optional):
    - AES-XCBC-MAC, HMAC-MD5
- ESP
  - Must be implemented (mandatory)
    - AES-CBC, 3DES-CBC, NULL
    - HMAC-SHA-1, NULL
  - Should/may be implemented (optional):
    - AES-CTR
    - AES-XCBC-MAC, HMAC-MD5

IPSec: Operation

- If sender has an SA to destination IP:
  - Use AH and/or ESP according to SA
  - Changes the IP accordingly
    - Adds AH and/or ESP headers
    - Replaces plaintext header/payload by an encrypted version
- If receiver has an SA with the headers' SPI:
  - Validates IPSec headers according with their SA
  - Upon a validation failure the datagram is discarded
    - Silently

IPSec: Issues

- Detection of useless/wrong Security Associations
  - Silent drop of invalid IPSec packets
- Overhead
  - Upper protocol layers can resent the same content
  - To much security (if already provided at upper layers)
- Anti-replay mechanisms
  - Each IPSec packet has a SA managed sequence number
    - Mandatory for the source
    - Opcionalmente validado pelo repector
    - Pode descartar datagramas correctos fora da janela de aceitação

IPSEC:
ESP não impede IP *spoofing*

- Ataque com repetição e IP *spoofing*

No.	Source	Destination	Protocol	Info	Length	Time	Delta Time
1	10.0.0.120	10.0.0.50	ESP	ESP (SPI=0x138ff4c0)	98	0.000000	0.000000
2	10.0.0.50	10.0.0.120	ESP	ESP (SPI=0x2dac6d96)	98	0.002724	0.002724
3	10.0.0.121	10.0.0.50	ESP	ESP (SPI=0x138ff4c0)	98	48.145456	48.142732
4	10.0.0.50	10.0.0.121	ICMP	Echo (ping) reply	74	48.148015	0.002559

- Para minimizar este problema pode-se usar ESP com cifra e autenticação em modo túnel
  - Mas não impede o *spoofing* do cabeçalho IP exterior

IPSec:
Setup of SAs

- Manual
  - With line-oriented or graphical tools
  - With libraries
- Automatic with protocol
  - ISAKMP (meta-protocol) (RFC 2408)
  - IKE (RFC 2409)

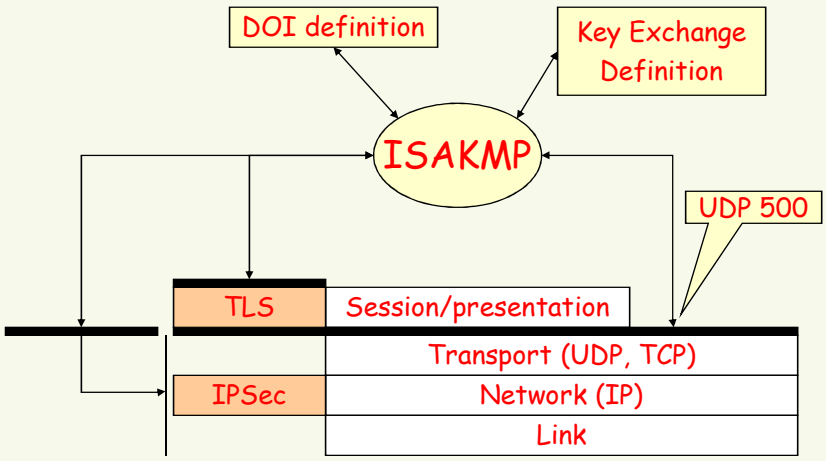
ISAKMP (*Internet Security Association and Key Management Protocol*, RFC 2408)

- Generic protocol (meta-protocol or framework)
  - Allows key negotiations and peer authentications
    - No specific techniques
    - 5 exchange types
      - Base Exchange
      - Identity Protection Exchange
      - Authentication Only Exchange
      - Aggressive Exchange
      - Informational Exchange
  - Two-phase protocol

    1 – Setup of an ISAKMP SA

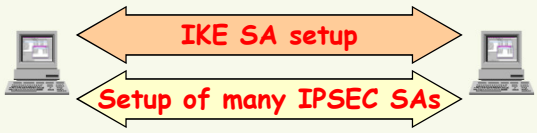
    2 – Setup of an SA for other protocols (e.g. IPSec)
- Applicational protocol
  - Can be used to setup SAs for any protocol

ISAKMP: Integration



IKE (Internet Key Exchange, RFC 2409)

- The only mandatory for IPSec
  - ISAKMP compliant
- Operation mode
  Phase 1: setup of a bidirectional IKE SA
  Phase 2: setup of unidirectional IPSEC SAs
    - Protected by IKE SA
    - Many IPsec SAs can reuse the same IKE SA



IKE:
Peer authentication

- Host (IP authentication)
  - Digital signatures and X.509 certificates
    - Distributed inline
  - Pre-distributed public keys of asymmetric key pairs
  - Pre-shared secret key
    - aka KEK (Key Encryption Key)

IKE:

Negotiation modes

- Main (Phase 1)
  - Setup of a bidirectional IKE SA
  - Identities are encrypted
  - 6 messages
    - Parameters agreement
    - Diffie-Hellman
    - Authentication
- Aggressive (Phase 1)
  - Equal to "main mode" without encrypted identities
  - 3 messages
- Quick (Phase 2)
  - Setup of two IPSec SAs
    - Ingress traffic
    - Egress traffic
  - Protected by an IKE SA
  - New key material
    - with DH
    - with key derivation
  - 3-4 messages

IPSec:

Negotiation example

No. .	Source	Destination	Protocol	Info	Length	Time	Delta Time
1	00:30:84:3e:c6:25	ff:ff:ff:ff:ff:ff	ARP	Who has 10.0.0.50? Tell 10.0.0.120	60	0.000000	0.000000
2	00:60:1d:f1:2a:93	00:30:84:3e:c6:25	ARP	10.0.0.50 is at 00:60:1d:f1:2a:93	60	0.002762	0.002762
3	10.0.0.120	10.0.0.50	ISAKMP	Identity Protection (Main Mode)	258	0.002862	0.000100
4	10.0.0.50	10.0.0.120	ISAKMP	Identity Protection (Main Mode)	150	0.805827	0.802965
5	10.0.0.120	10.0.0.50	ISAKMP	Identity Protection (Main Mode)	226	0.912235	0.106408
6	10.0.0.50	10.0.0.120	ISAKMP	Identity Protection (Main Mode)	226	0.950597	0.038362
7	10.0.0.120	10.0.0.50	ISAKMP	Identity Protection (Main Mode)	110	0.992395	0.041798
8	10.0.0.50	10.0.0.120	ISAKMP	Identity Protection (Main Mode)	110	1.017310	0.024915
9	10.0.0.120	10.0.0.50	ISAKMP	Quick Mode	430	1.020704	0.003394
10	10.0.0.50	10.0.0.120	ISAKMP	Quick Mode	182	1.035273	0.014569
11	10.0.0.120	10.0.0.50	ISAKMP	Quick Mode	94	1.035684	0.000411
12	10.0.0.50	10.0.0.120	ISAKMP	Quick Mode	126	1.061640	0.025956
13	10.0.0.120	10.0.0.50	ICMP	Echo (ping) request	98	1.062270	0.000630
14	10.0.0.50	10.0.0.120	ICMP	Echo (ping) reply	98	1.064842	0.002572

No. .	Source	Destination	Protocol	Info	Length	Time	Delta Time
1	10.0.0.120	10.0.0.50	ISAKMP	Identity Protection (Main Mode)	258	0.000000	0.000000
2	10.0.0.50	10.0.0.120	ISAKMP	Identity Protection (Main Mode)	150	0.107144	0.107144
3	10.0.0.120	10.0.0.50	ISAKMP	Identity Protection (Main Mode)	226	0.173738	0.066594
4	10.0.0.50	10.0.0.120	ISAKMP	Identity Protection (Main Mode)	226	0.211771	0.038033
5	10.0.0.120	10.0.0.50	ISAKMP	Identity Protection (Main Mode)	110	0.234565	0.022794
6	10.0.0.50	10.0.0.120	ISAKMP	Identity Protection (Main Mode)	110	0.238690	0.004125
7	10.0.0.120	10.0.0.50	ISAKMP	Quick Mode	422	0.239454	0.000764
8	10.0.0.50	10.0.0.120	ISAKMP	Quick Mode	182	0.245744	0.006290
9	10.0.0.120	10.0.0.50	ISAKMP	Quick Mode	94	0.246074	0.000330
10	10.0.0.50	10.0.0.120	ISAKMP	Quick Mode	126	0.250026	0.003952
11	10.0.0.120	10.0.0.50	ESP	ESP (SPI=0xdc262791)	98	0.250707	0.000681
12	10.0.0.50	10.0.0.120	ESP	ESP (SPI=0xc4891de3)	98	0.253161	0.002454

IPSec:
Issues with NAT (RFC 3715)

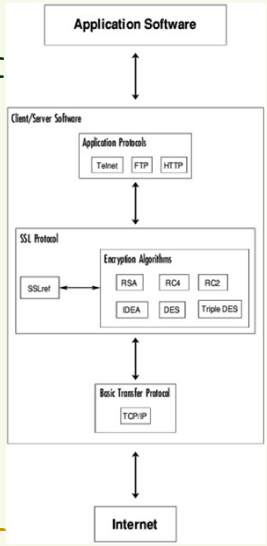
- NAT is not a "clean solution" for the IP end-to-end paradigm
  - Its a "survival" hack
- NAT impact is twofold
  - Network (changes src or dst IP)
  - Transport (changes src or dst port)
- Impact in IPSec
  - Doesn't work with transport mode
    - AH prevents IP header changes,
    - ESP prevents transport header changes
  - SPI-based multiplexing can fail for multiple reasons
  - ISAKMP/IKE doesn't work
    - Uses exclusively port 500
- Solution: NAT-T (NAT Traversal)
  - IPSec encapsulation into UDP (RFC 3948)
  - Detection of NAT box between end-hosts
  - Identification of NAT-capable end-hosts

NAT-T (RFC 3947):

- Another hack
  - With several weaknesses
- NAT-T support announcement
  - RFC 3947 hash within Vendor ID of first Phase 1 messages
- Detection of NAT
  - NAT-D payload w/ hashes of addresses and ports
- Adoption of a different ISAKMP port ASAP
  - To minimize ambiguities caused by different NAT behaviors
  - Keepalives to maintain NAT mappings stable
- Negotiation of NAT-T encapsulation
  - Tunnel / transport
- NAT-OA payload with original IP addresses
  - To transport UDP-encapsulation (required to update TCP checksums)

TLS (*Transport Layer Security*, RFC 2246): Objectivos

- Protocolo de comunicação segura sobre TCP/I
  - Padrão criado a partir do SSL V3 (*Secure Sockets Layer*)
  - Gere sessões seguras sobre TCP/IP por aplicação
    - Inicialmente concebido para tráfego HTTP
    - Actualmente usado para outros tráfegos
- Mecanismos de segurança
  - Confidencialidade e integridade da comunicação
  - Distribuição de chaves
  - Autenticação de interlocutores
    - Servidores (ou, usualmente, serviços)
    - Utilizadores cliente



TLS/SSL: Protocolos

- *Handshake Protocol*
  - Distribuição de chaves
    - Master secrets (48 octetos)
      - Calculado com DH; ou
      - Escolhido apenas pelo cliente, enviado para o servidor cifrado com chave pública do destinatário
    - Chaves de sessão (client/server read/write keys)
      - Calculadas a partir de um *master secret* e de nonces trocados no protocolo
  - Autenticação de interlocutores
    - Cifra assimétrica com chaves de longa duração ou efémeras
    - Certificados de chaves públicas de longa duração
- *Record Protocol*
  - Criação e análise de mensagens seguras
  - Compressão, cifra, controlo de integridade

TLS/SSL:
Gestão da operação

- Modelo cliente-servidor
  - Tal como numa ligação TCP
- As aplicações definem a estratégia
  - Autenticação do interlocutor
    - Se é necessária e como tem que se efectuar
  - Algoritmos
    - Ambos apresentam as suas possibilidades
    - O servidor escolhe
  - Gestão de chaves de sessão
    - Tempo de vida de um *master secret*
    - Tempo de vida de chaves de sessão
      - Igual ou menor que o tempo de vida de uma ligação TCP

TLS/SSL Handshake #1:
No Session Identifier, No Client Authentication, Certificate-based Server Authentication

C S	Mensagem	
→	Client-Hello	challenge-data, cipher-specs
←	Server-Hello	connection-id, server-certificate, cipher-specs
→	Client-Master-Key	cipher-kind, clear-master-key, $\{\text{secret-master-key}\}_{\text{server-public-key}}$
→	Client-Finish	$\{\text{connection-id}\}_{\text{client-write-key}}$
←	Server-Verify	$\{\text{challenge-data}\}_{\text{server-write-key}}$
←	Server-Finish	$\{\text{session-id}\}_{\text{server-write-key}}$

TLS/SSL Handshake #2:
Session Identifier Used, No Client Authentication

C S	Mensagem	
→	Client-Hello	challenge-data, session-id, cipher-specs
←	Server-Hello	connection-id, session-id-hit
→	Client-Finish	$\{connection\text{-}id\}_{client\text{-}write\text{-}key}$
←	Server-Verify	$\{challenge\text{-}data\}s_{erver\text{-}write\text{-}key}$
←	Server-Finish	$\{session\text{-}id\}_{server\text{-}write\text{-}key}$

TLS/SSL:
Utilização

- É apenas um protocolo padrão
  - Não existe uma API fixa
- Interfaces mais utilizadas
  - API de referência para o SSL
    - SSLref
  - API de realizações públicas do SSL
    - SSLeay, OpenSSL
- Utilização por servidores
  - Duas interfaces
    - Uma convencional, como o porto de transporte habitual
    - Uma sobre SSL, com um porto de transporte novo
  - O cliente define as suas intenções consoante a interface que usa

Protocolos	Porto normal	Porto TLS
HTTP	80	443
IMAP	143	993
POP3	110	995
SMTP	25	465
LDAP	389	636

SSH (*Secure Shell*): Objectivos

- Uma aplicação e um protocolo de comunicação segura sobre TCP/IP
  - Permite estabelecer sessões remotas seguras
  - Permite multiplexar outros fluxos de informação sobre uma sessão segura
    - Encapsulamento seguro de tráfego TCP/IP
- Mecanismos de segurança
  - Confidencialidade e integridade da comunicação
  - Distribuição de chaves
  - Autenticação de interlocutores
    - Do servidor (ou, normalmente, da máquina servidora)
    - Do utilizador cliente

SSH: Protocolos

- *Transport Layer Protocol*
  - Distribuição de chaves
    - Chave de sessão calculada com valores efémeros de DH
  - Autenticação do servidor
    - Assinatura de valores efémeros trocados e da chave calculada com DH
    - A chave pública do servidor pode ser transferida na altura se ainda não estivar na posse do cliente
  - Criação e análise de mensagens seguras
  - Compressão, cifra e controlo de integridade
- *Authentication Protocol*
  - Autenticação de utilizadores perante a máquina remota
  - Tipicamente com senha
- *Connection Protocol*
  - Multiplexagem de fluxos de informação sobre uma sessão segura

Correio electrónico seguro: PGP e PEM

- Mesma filosofia de funcionamento
  - Transformam objectos embebidos em mensagens RFC 822
  - Os objectos podem:
    - Ser assinados pelo remetente
    - Cifrados para o destinatário
      - Com cifra mista e a chave pública do destinatário
  - A diferença está no modelo de gestão de chaves públicas pessoais
    - No PGP todos podem ser entidades certificadoras
    - No PEM existe uma única hierarquia universal de entidades certificadoras

Negociação de chaves de sessão e autenticação de interlocutores

Protocolo	Chaves de sessão			Autenticação	
	algoritmo	SFP	Renegoc.	Tipo	Entidades
PGP	Off-line PK	---	---	Obrigatória Simétrica Unilateral ou mútua	Pessoas (emissor ou receptor)
PEM	Off-line PK Off-line SK				
SSH	DH PK	✔	✔	Opcional Assimétrica Unilateral ou mútua	Servidor (sshd) Utente cliente
TLS / SSL			✔	Opcional Simétrica Unilateral ou mútua	Cliente ou servidor (máquinas, pessoas ou serviços)
IPSEC IKE	DH Off-line SK + nonces	✔ ✘	✔	Obrigatória Simétrica Mútua	Máquinas
GSM	Off-line SK + nonces (desafios)	✘	✔	Obrigatória Unilateral	Cartões Operadoras
WEP	---	✘	✘	Opcional Unilateral	Pessoas (ou grupos)