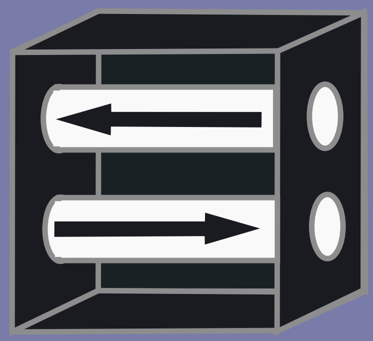


Certificates | IPsec

INGI2347: COMPUTER SYSTEM SECURITY (Spring 2014)

Marco Canini

UCL
Université catholique de Louvain

Lecture slides adapted from UCL INGI2347 by Gildas Avoine, Upenn CSE331 by Steve Zdancewic, and Dan Boneh, Stanford Cryptography I@Coursera. Reproduced with permission

Plan for today

Lecture 8

- Certificates
  - NEXT
  - Working with certificates
- VPN
- IPsec
  - Security Association (SA)
  - Authentication Header (AH)
  - Encapsulated Security Payload (ESP)
  - Transport and Tunnel Modes
  - Internet Key Exchange (IKE)

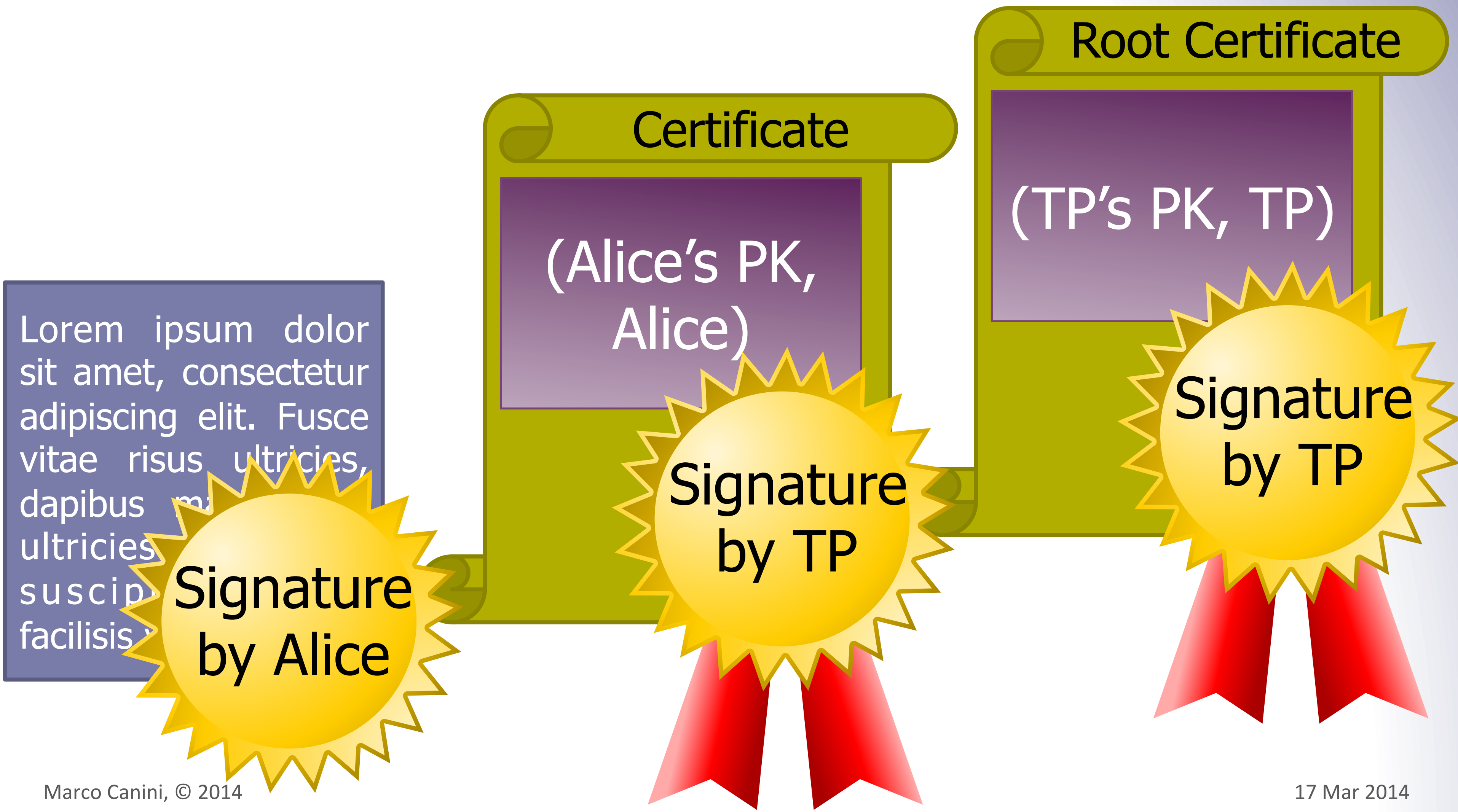
Certificates

What is a certificate?

Certificate's goal is to **link** a public key (PK) with its owner

- The pair (PK, owner) is signed by a trusted party (TP)
- The TP is named **Certification Authority** (CA)
- To check the signature, the CA's PK is needed
  - **Root certificate**: the pair (CA's PK, CA) is self-signed
  - The authenticity of the root certificate is fundamental (included in browsers)

Illustration



X.509 Certificates

X.509
- Standard from International Telecommunication Union (ITU), 1988
- Also IETF RFC-2459 (and updates)

Three required fields:

- TBS Certificate (TBS = “To Be Signed”)
  - The useful payload of the certificate
- CA signature algorithm
  - Identifier for the crypto algorithm used by the CA to sign this certificate
- CA signature value
  - Signature of the certificate by the CA

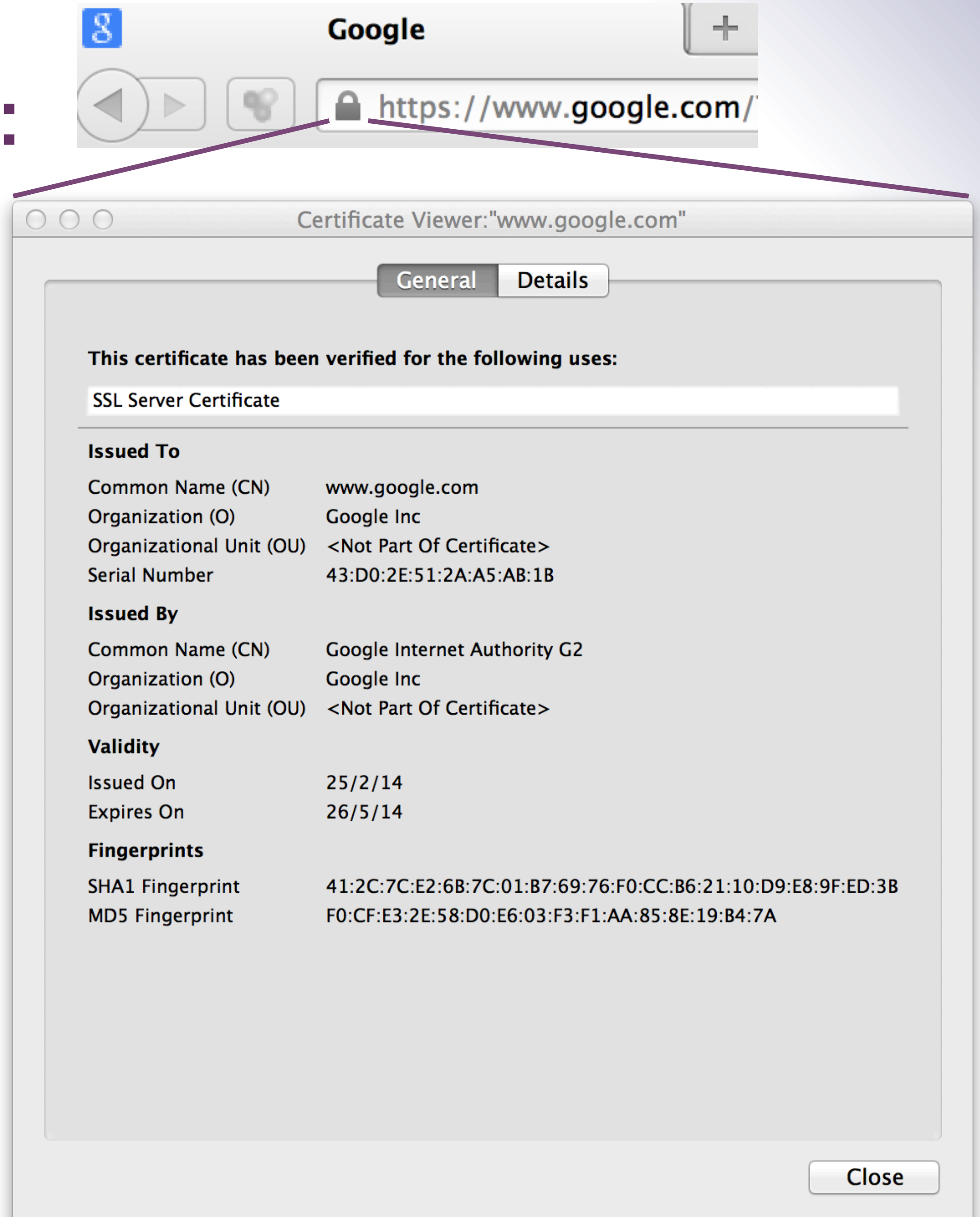
X.509 TBS

- Serial number
  - Unique number assigned by the CA to the certificate
- Issuer field
  - Identifies the entity who has signed and issued the certificate
- Subject
  - Identifies the entity associated with the public key
    - O: organization, C: country, OU: organization unit, CN: common name, ST: state, L: city, etc. no IP address

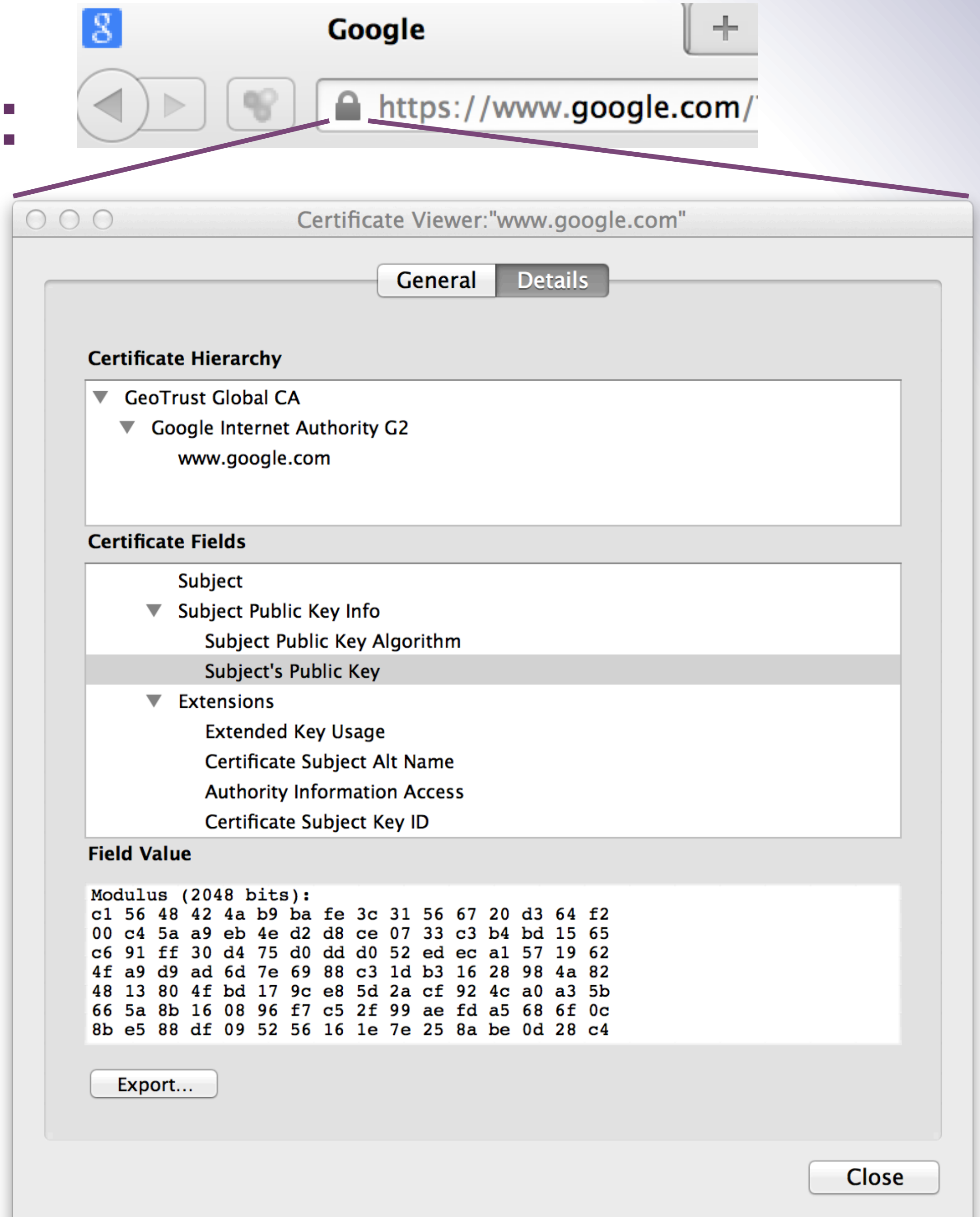
X.509 TBS (Continued)

- Validity
  - Not before
  - Not after
- Subject Public Key Info
  - Public key
  - Identifies the algorithm with which the key is used
    - e.g., RSA, DSA, or DH
- Etc.

Example:


Google
https://www.google.com/
Certificate Viewer:"www.google.com"
General
Details
This certificate has been verified for the following uses:
SSL Server Certificate
Issued To
Common Name (CN) www.google.com
Organization (O) Google Inc
Organizational Unit (OU) <Not Part Of Certificate>
Serial Number 43:D0:2E:51:2A:A5:AB:1B
Issued By
Common Name (CN) Google Internet Authority G2
Organization (O) Google Inc
Organizational Unit (OU) <Not Part Of Certificate>
Validity
Issued On 25/2/14
Expires On 26/5/14
Fingerprints
SHA1 Fingerprint 41:2C:7C:E2:6B:7C:01:B7:69:76:F0:CC:B6:21:10:D9:E8:9F:ED:3B
MD5 Fingerprint F0:CF:E3:2E:58:D0:E6:03:F3:F1:AA:85:8E:19:B4:7A
Close

Example:


Google
https://www.google.com/
Certificate Viewer:"www.google.com"
General
Details
Certificate Hierarchy
GeoTrust Global CA
Google Internet Authority G2
www.google.com
Certificate Fields
Subject
Subject Public Key Info
Subject Public Key Algorithm
Subject's Public Key
Extensions
Extended Key Usage
Certificate Subject Alt Name
Authority Information Access
Certificate Subject Key ID
Field Value
Modulus (2048 bits):
c1 56 48 42 4a b9 ba fe 3c 31 56 67 20 d3 64 f2
00 c4 5a a9 eb 4e d2 d8 ce 07 33 c3 b4 bd 15 65
c6 91 ff 30 d4 75 d0 dd d0 52 ed ec a1 57 19 62
4f a9 d9 ad 6d 7e 69 88 c3 1d b3 16 28 98 4a 82
48 13 80 4f bd 17 9c e8 5d 2a cf 92 4c a0 a3 5b
66 5a 8b 16 08 96 f7 c5 2f 99 ae fd a5 68 6f 0c
8b e5 88 df 09 52 56 16 1e 7e 25 8a be 0d 28 c4
Export...
Close

Working with Certificates

Certificate Authorities

- Issuers of certificates found on web servers

CA	Count [%]
GeoTrust	25.19
GoDaddy.com	13.65
Verisign	13.09
Thawte	9.79
Comodo Limited	7.12
Unknown	2.40
DigiCert	2.39
Network Solutions LLC	2.09
Comodo CA Limited	1.77
GlobalSign	1.64

NOTE: GeoTrust, Verisign, and Thawte are the same group

Source: https://secure1.securityspace.com/es/s survey/data/man.201002/casurvey.html (Feb 2010)

How to obtain a certificate

- Applicant registers with a CA
- CA (physically) authenticates the applicant
- CA asks applicant to generate public/private keys
- CA creates a certificate with the applicant's identity, PK, expiration date, etc., and the CA's signature
- CA provides a copy of its own PK to applicant

Registration Authority (RA)

- CA can delegate the registration of an applicant to the **registration authority** (RA)
- RA does not have CA's private key
- CA trusts the RA to authenticate the applicants
- After applicant is authenticated, applicant generates a pair of keys and sends the public key to the CA to create the certificate
- Technically RA sends a signed Certificate Signing Request (CSR) to the CA

CSR in practice

- Generate a 1024-RSA key-pair
  - openssl genrsa 1024 > mykey.key
- Generate a CSR
  - openssl req -new -key mykey.key -out myreq.csr
- Verify a CSR
  - openssl req [-text] [-noout] -verify -in myreq.csr
- Online checkers
  - http://support.ecenica.com/ssl-certificates/csr-checker/
  - https://ssl-tools.verisign.com/checker/

Certificate without CA

- Everyone can self-sign a certificate
- Distribute the certificate through an authenticated channel
- Makes sense in enterprise intranet
- Not really for public-facing services
- Rather get a free certificate…

Certificates in practice

- Generate a certificate
  - openssl x509 -req -in myreq.csr -signkey mykey.key -out mycert.crt
- View a certificate
  - openssl x509 -text -in mycert.crt
- Verify a certificate
  - openssl verify mycert.crt

Key escrowing

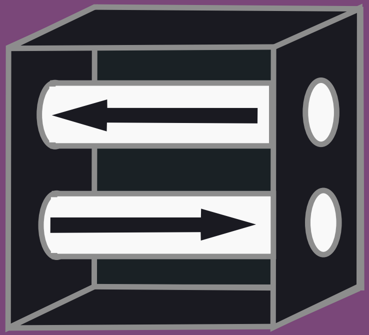
Keys are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys

Example:

- A company can provide two key pairs and certificates to each of is employees
  - One for signing | One for encrypting
- CA escrows a copy of the private encryption key
- Only employees can sign, but company can decrypt

Verifying a certificate

1. Verify the **certification path**
   - Performed locally
   - Delegated to a server: SCVP
     - Server-based Certificate Validation Protocol
2. Verify the **validity period**
3. Verify that the certificate is **not revoked**
   - Performed locally: CRL (certificate revocation lists)
   - Delegated to a server: OCSP
     - Online Certificate Status Protocol
     - Supported by all major browsers (enabled by default in Firefox and Safari)



VPN

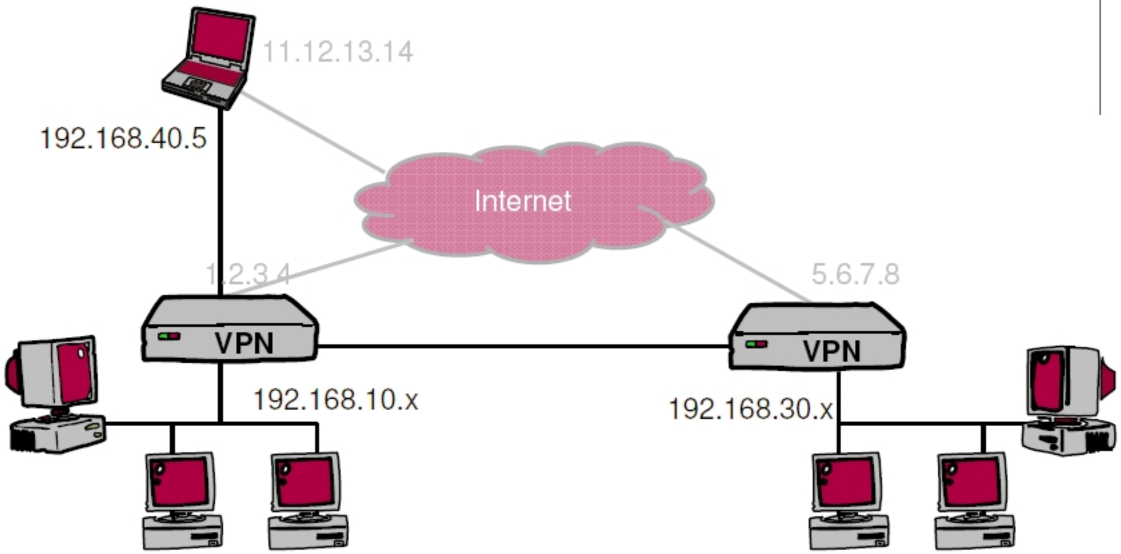
Virtual Private Network

Virtual Private Network (VPN)

Goal: extend a private network across a public network

- Scenarios:
  - Interconnection of remote sites through the Internet
  - Access to a company's network from a laptop connected to Internet

VPN Illustration



VPN basics

- VPN software on routers or PCs (e.g. laptop)
- Packet encapsulation across the Internet
- Encryption of data to guarantee confidentiality

Existing VPN Protocols

- Point to Point Tunneling Protocol (PPTP)
  - Microsoft
- Layer 2 Tunneling Protocol (L2TP)
  - IETF
  - Result of merging Cisco's Layer 2 Forwarding (L2F) protocol and Microsoft's PPTP protocol
- IP Security (IPsec)
  - IETF

IPsec

IP Security

IPsec Overview

- Open standard developed by the IETF
  - Public algorithms for confidentiality, authentication, integrity
- Authentication Headers (AH)
  - Provide connectionless **integrity** and origin **authentication** for IP packets
- Encapsulating Security Payloads (ESP)
  - Provide **confidentiality**, data-origin **authentication**, connectionless **integrity**
- Security Associations (SA)
  - Provide algorithms and parameters necessary to AH and/or ESP operations
- Internet Key Exchange (IKE)
  - Key exchange protocol
- Two operation modes: **tunnel**, **transport**

Security Association (SA)

- End hosts willing to exchange packets securely must first establish a Security Association (SA)
- A SA is simply the bundle of algorithms and parameters that is being used to encrypt and authenticate a particular flow in one direction
- SA memorizes algorithms, keys, validity periods, sequence numbers and peer's identity
- In normal bi-directional traffic (like TCP), flows are secured by a pair of SAs

Security Association (SA)

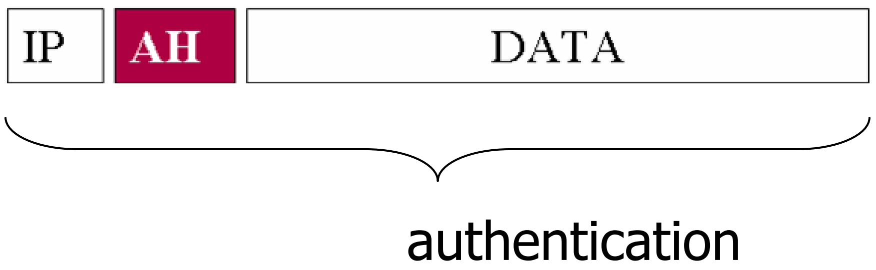
SAs are identified by a Security Parameter Index (SPI)

- The source indicates the SPI on all packets that it sends
- The destination uses the SPI to find the corresponding SA

- The source decides which packets must be processed with which SA
- One SA per destination, per protocol (AH or ESP), per flow

Authentication Header (AH)

- The addition of an authentication header allows verifying the packet's **authenticity** and **integrity**



Authentication Header (AH)

0	8	16	31
Next Header	Length	0	
Security Parameter Index (SPI)			
Sequence Number			
Integrity Check Value (ICV) (variable length)			

- Next Header: Specifies the encapsulated protocol (ICMP, TCP, UDP,…)
- Length: Size of this Authentication Header in 32-bit units, minus 2 (i.e., - 64 bits)
- Security Parameters Index: Contains a pseudo random value used to identify the security association for this datagram.
- Sequence Number: Monotonically increasing number to avoid replay-attacks.
- Integrity Check Value: Contains keyed-hash value

Authentication Header (AH)

- Authentication is calculated on:
  - Data that follow the AH
  - AH itself (with ICV set to zero)
  - Pseudo IP header
    - Source, destination, protocol, length, version, etc.
- The algorithm to be used to generate the authentication data is negotiated when the SA is created
- Two algorithms must be available:
  - HMAC-SHA-96
  - HMAC-MD5-96

Recall HMAC

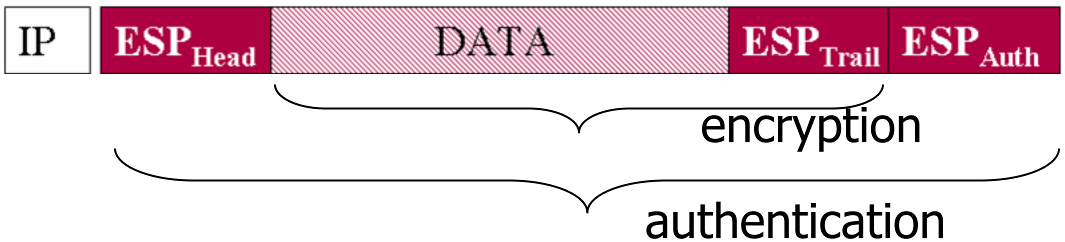
- Most widely used MAC on the Internet
  - Proposed by Bellare, Canetti, Krawczyk in 1996
  - Provably secure
  - Standards: FIPS 198-1, RFC 2104, ISO 9797-2
- Builds a MAC out of a hash function

HMAC: $S(k, m) = H\big(\, k \oplus \text{opad} \;||\; H(\, k \oplus \text{ipad} \,||\, m \,)\,\big)$

  - Examples:
    - HMAC-SHA256: H = SHA256 ; output is 256 bits
    - HMAC-SHA1-96: H = SHA1 ; output truncated to 96 bits

Encapsulated Security Payload (ESP)

- The ESP header allows packet encryption and authentication



- Encryption is done only on the encapsulated data and the trailer
- Encryption is done neither on the header's fields, nor on the authentication data
- Optional authentication is done on the ESP header and all that follow, but not on the IP header

Encapsulated Security Payload (ESP)

0 8 16 24 31

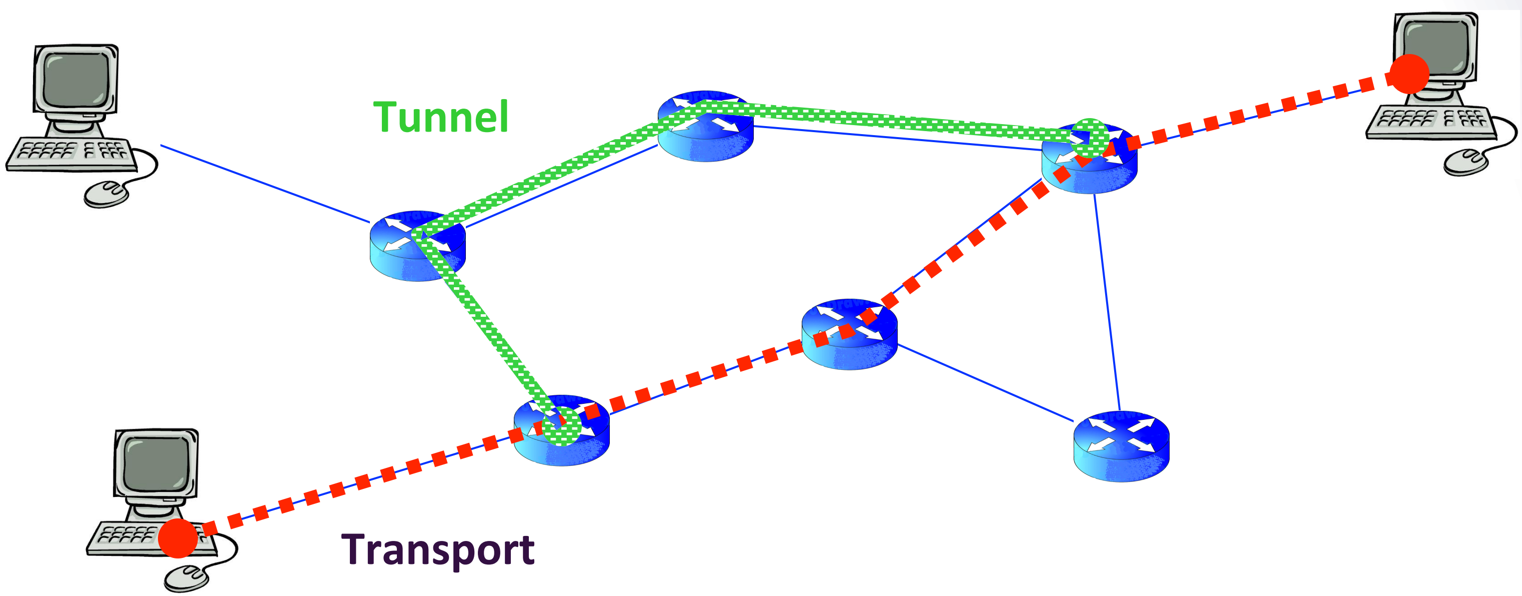
<table>
<tr><td>Next Header</td><td>Length</td><td colspan="2">0</td></tr>
<tr><td colspan="4">Security Parameter Index (SPI)</td></tr>
<tr><td colspan="4">Sequence Number</td></tr>
<tr><td colspan="4">Payload data (variable length)</td></tr>
<tr><td></td><td>Padding
(0 -255 bytes)</td><td>Pad Length</td><td>Next Header</td></tr>
<tr><td colspan="4">Integrity Check Value (ICV) (variable length)</td></tr>
</table>

- The mandatory algorithms are:
  - Encryption: DES-CBC, NULL (RFC 2410)
  - Authentication: HMAC-SHA-96 (RFC2404), HMAC-MD5-96 (RFC2403), NULL
- NULL encryption and NULL authentication in the same SA is not allowed

Transport and Tunnel Modes

Transport & Tunnel Modes

- **Transport**: only protects the packet's payload
- **Tunnel**: entire packet is encapsulated



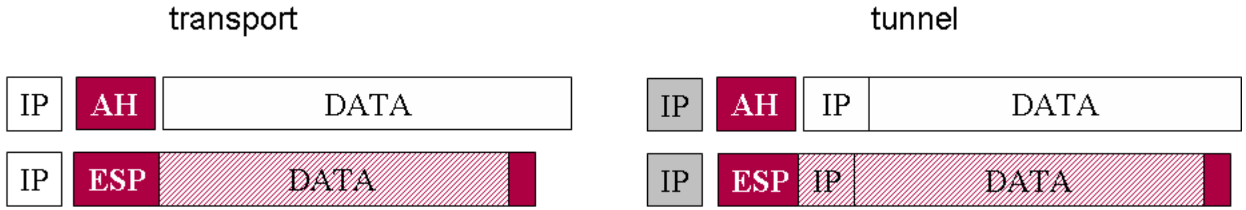
Transport & Tunnel Modes

Which packets need to be encrypted/authenticated?

- Each router contains a Security Policy Database
- SPD defines which packet needs to be secured
  - According to discriminators: destination address, source address, …
- Example:
  - Secure all HTTP traffic
  - Secure packets sent to remote sites but not to the Internet
  - Secure UDP
  - Secure TCP but not SSL

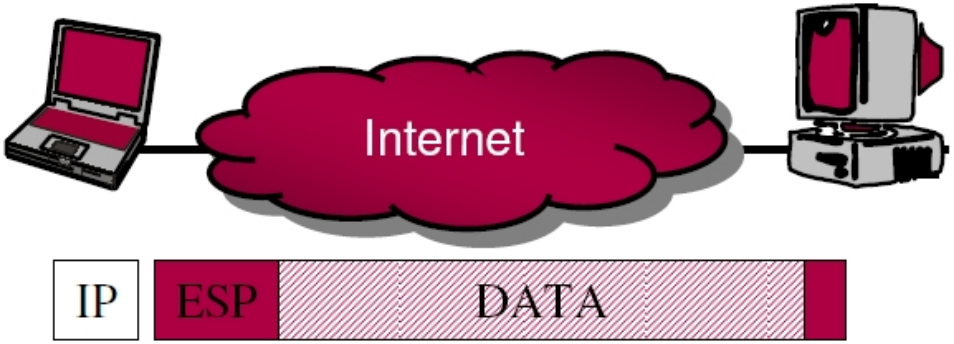
Transport & Tunnel Modes

- Transport mode:
  - Only IP packet **payload** is encrypted and/or authenticated
- Tunnel mode:
  - The entire packet is encapsulated in a new packet



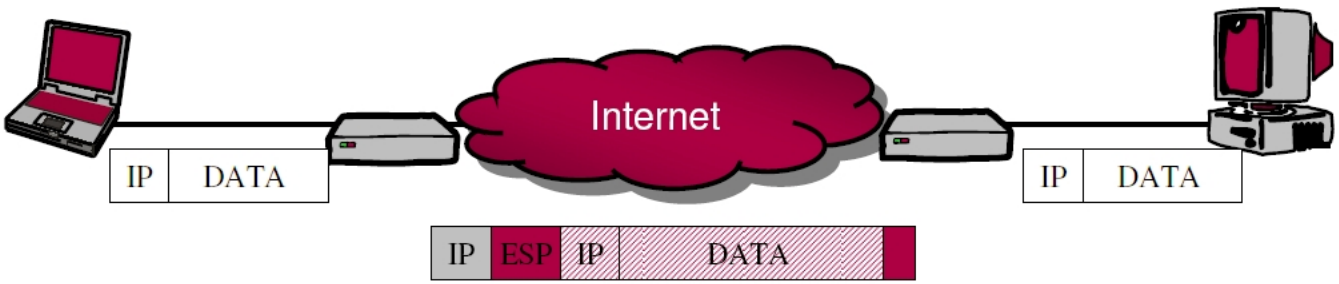
Transport Mode

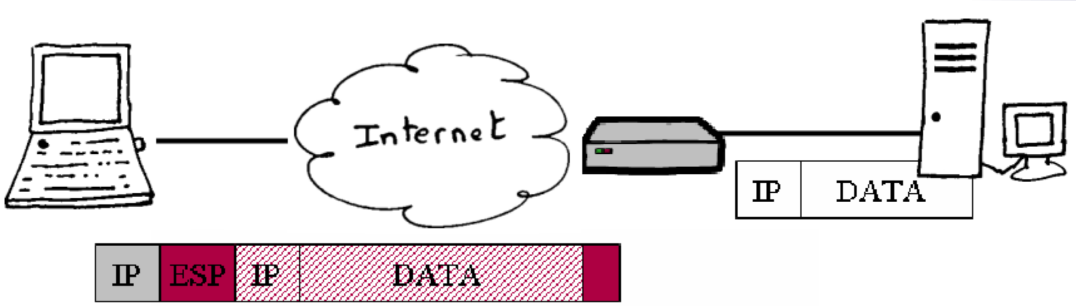
- Security is done end-to-end



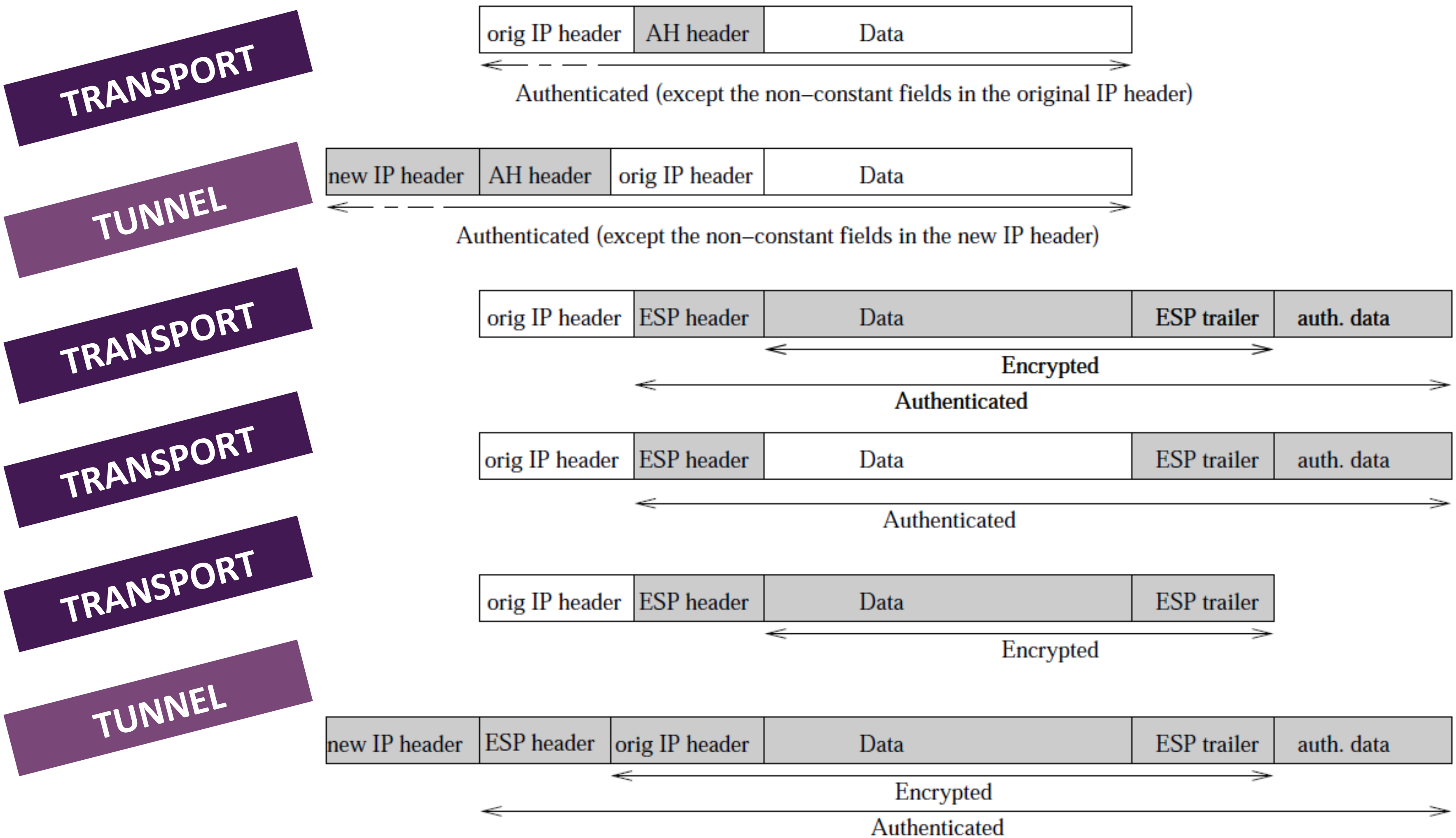
Tunnel Mode

- Security can be added by intermediate routers

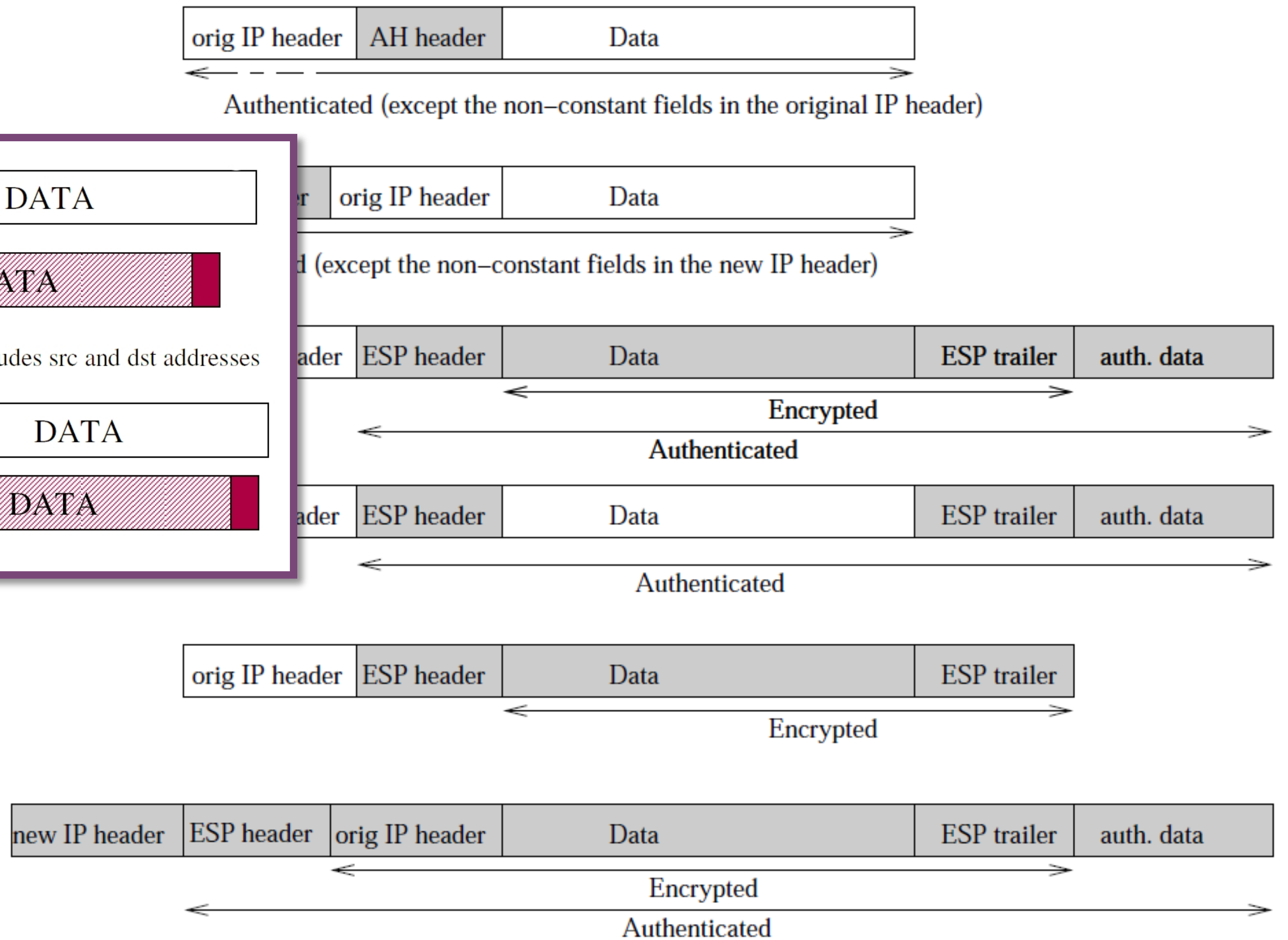


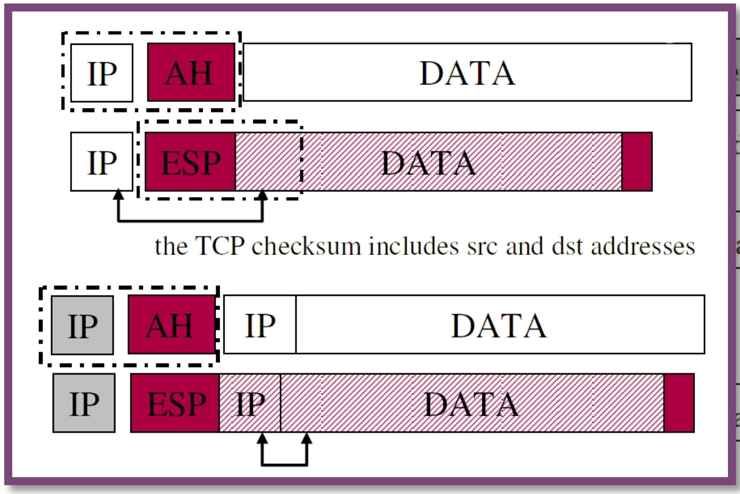


Quiz: What is the Applied Mode?



IPsec and NAT





IPsec and NAT

- The TCP and UDP checksum calculation includes a pseudo header made of src and dst IP addresses and ports
- When doing NAT, the checksum has to be readjusted every time the source IP address (and port) changes
- This does not work if the payload is encrypted or authenticated
- **NAT-T** mechanism: encapsulate IPsec in UDP to traverse NAT

Internet Key Exchange (IKE)

Internet Key Exchange (IKE)

Internet Key Exchange (IKE) is a protocol used to establish a SA between communicating partners

- IKE's aims:
  - Partner authentication
  - Key exchange between partners
  - Parameters negotiation
- IKE's result is a Security Association (SA)
  - SA is identified with a given Security Parameter Index (SPI)
- IKEv1 RFC 2409, 1998
- IKEv2 RFC 4306, 2005, updated RFC 5996, 2010

Two Phases of IKEv1

- Phase 1: set up an SA to protect the negotiations
- Phase 2: set up the SA for an ESP or AH flow

IKEv1: phase 1

- Authentication
  - Pre-shared secrets (PSS)
  - Public keys peer-exchanged
  - X.509 certificates
    - Require public key of certification authority
- Key Exchange
- Generate a shared key using Diffie-Hellman

IKEv1: phase 1

Parameter negotiation

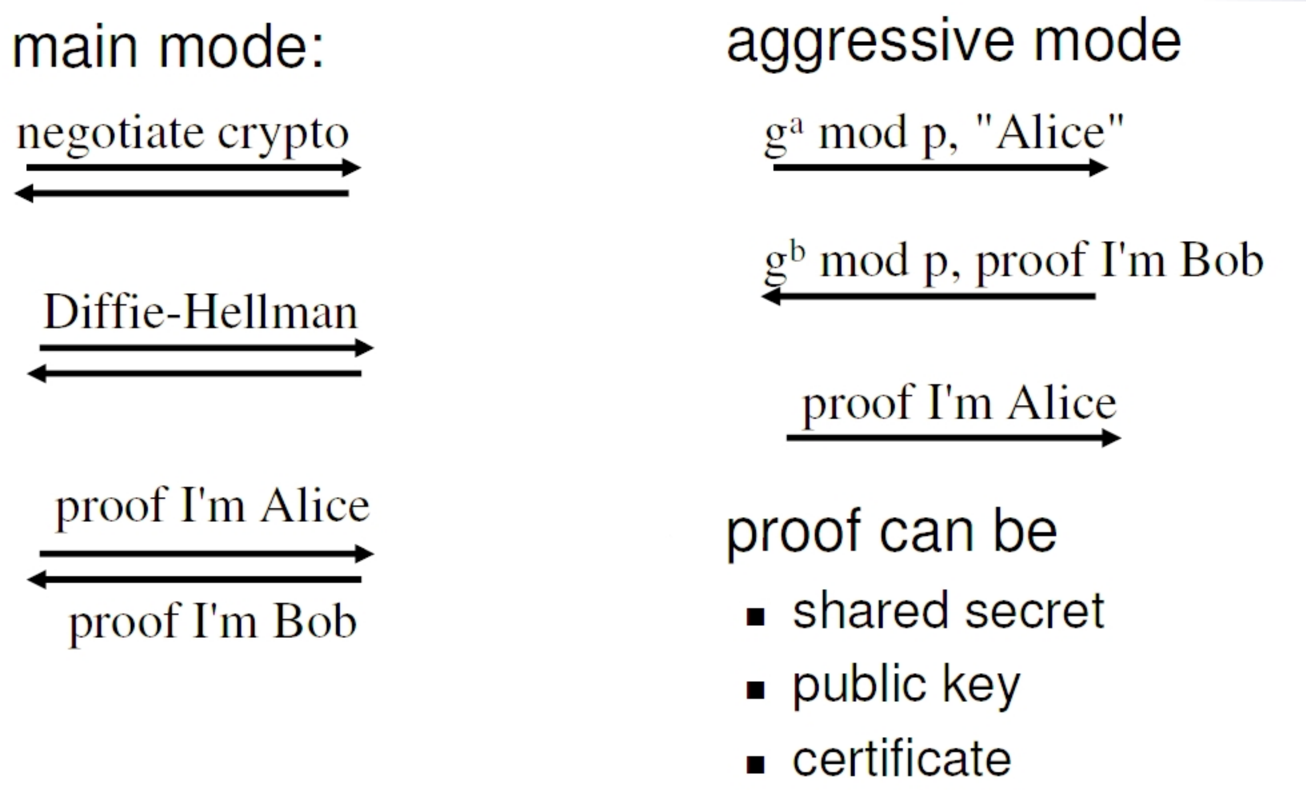
Main Mode

- More negotiation possibilities
  - E.g., DH parameters
- Protects the initiator's identity, and PSS's hash value (if used)

Aggressive Mode

- Faster but less negotiation possibilities
- Reveals the initiator's identity, and PSS's hash value (if used)

Main Mode & Aggressive Mode



Perfect Forward Secrecy (PFS)

A key-agreement protocol has PFS if it ensures that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future

- Forward secrecy is designed so that the compromise of one message cannot lead to the compromise of others
- Also that there is not a single secret value which can lead to the compromise of multiple messages

IKEv1: phase 2

- Only one mode, called “quick mode”
- Without PFS
  - Keys are periodically refreshed (typically every hour): session keys
  - Session keys are derived from the same secret
  - Stealing this secret compromises all keys
- With PFS
  - A Diffie-Hellman is done for each new session key (slower)
  - Stealing one key does not compromise the previous ones

Glossary

- VPN: Virtual Private Network
- PPTP: Point to Point Tunneling Protocol
- L2TP: Layer 2 Tunneling Protocol
- L2F: Layer 2 Forwarding
- IPsec: IP Security
- AH: Authentication Header
- ESP: Encapsulated Security Payload
- IKE: Internet Key Exchange
- SPI: Security Parameter Index

Any questions?

Stay tuned

Next time you will learn about

WEP