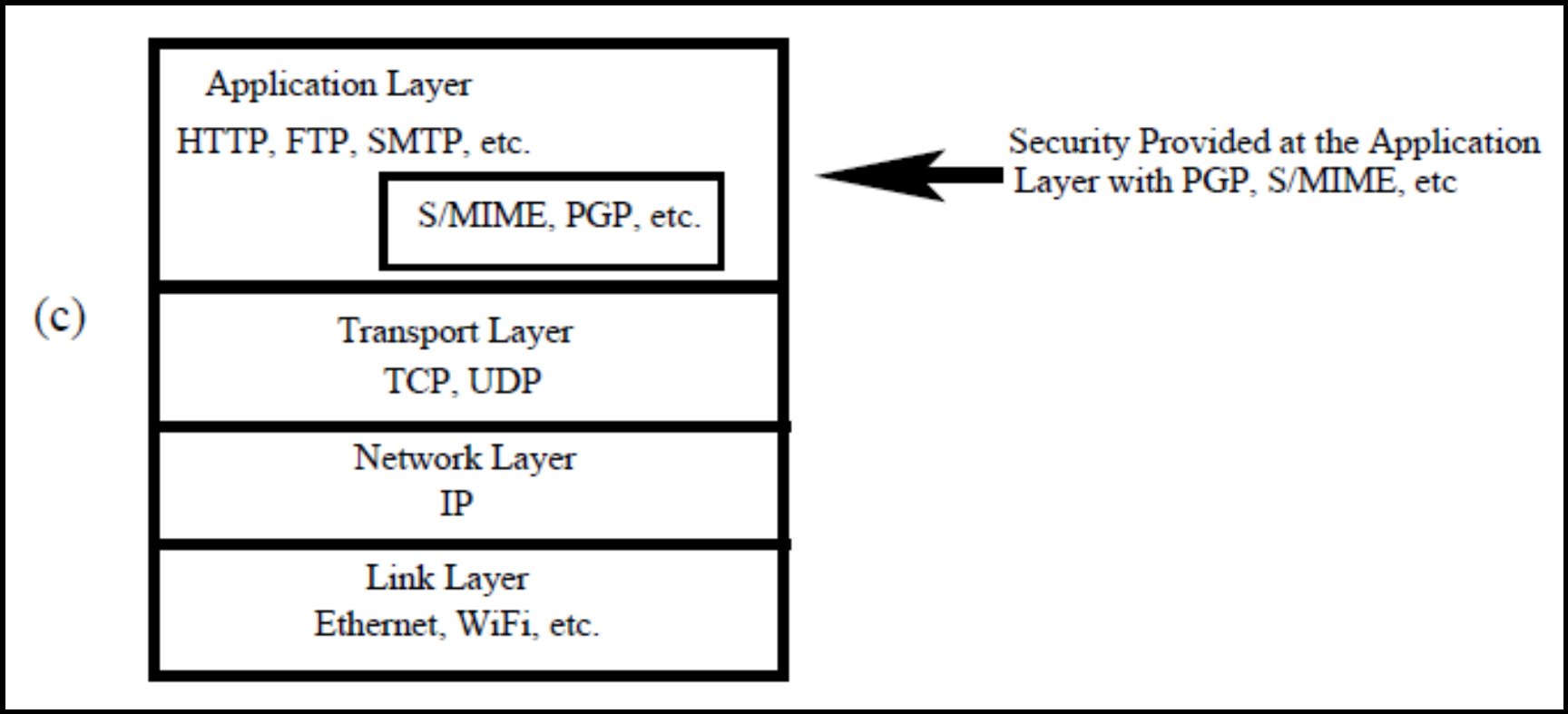
S/MINE, DKIM, IPSec, and SSL/TLS

Agenda

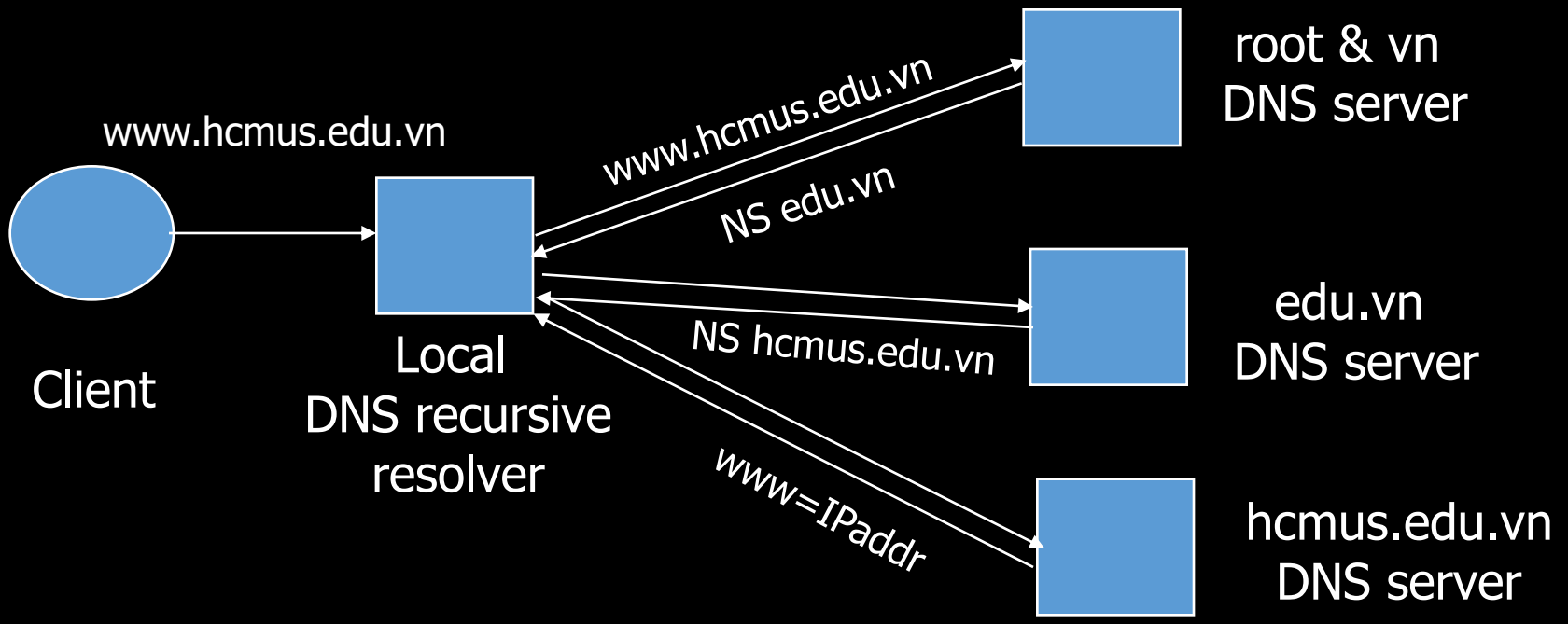
- S/MINE
- DKIM
- BGP
- DNS
- IPSec
- SSL/TLS

Application Layer



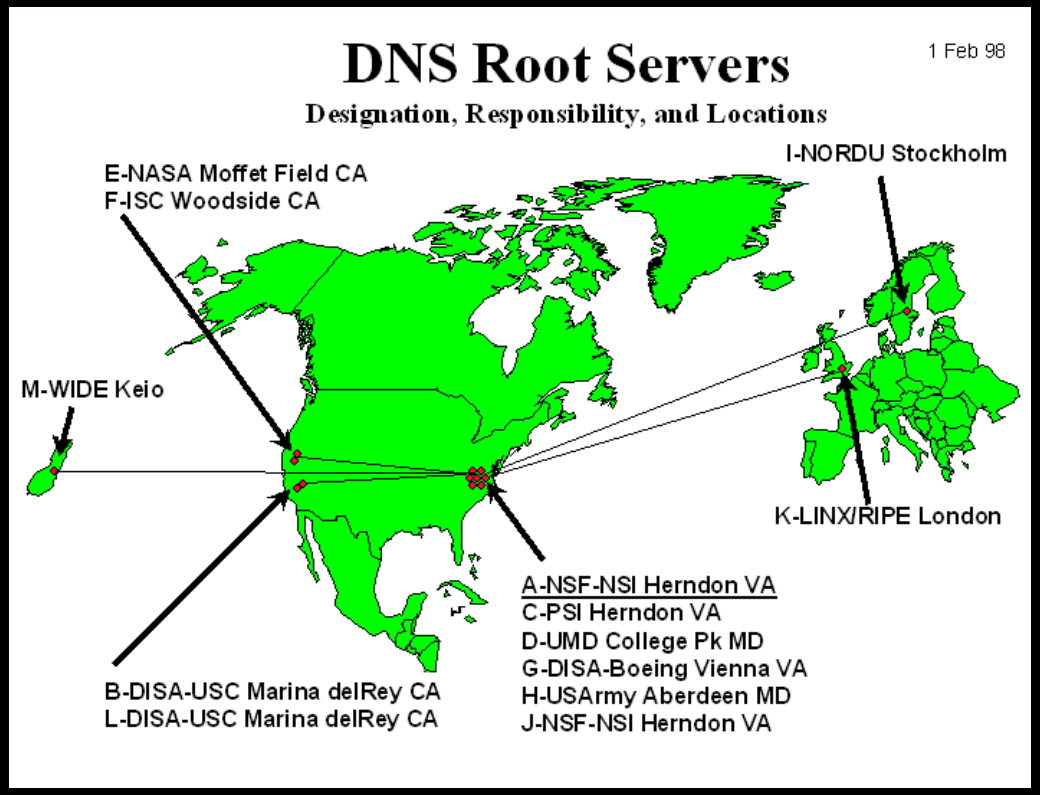
DNS: Domain Name Service

DNS maps symbolic names to numeric IP addresses
(for example, www.hcmus.edu.vn ↔ 172.29.1.16)



DNS Root Name Servers

- Root name servers for top-level domains
- Authoritative name servers for subdomains
- Local name resolvers contact authoritative servers when they do not know a name

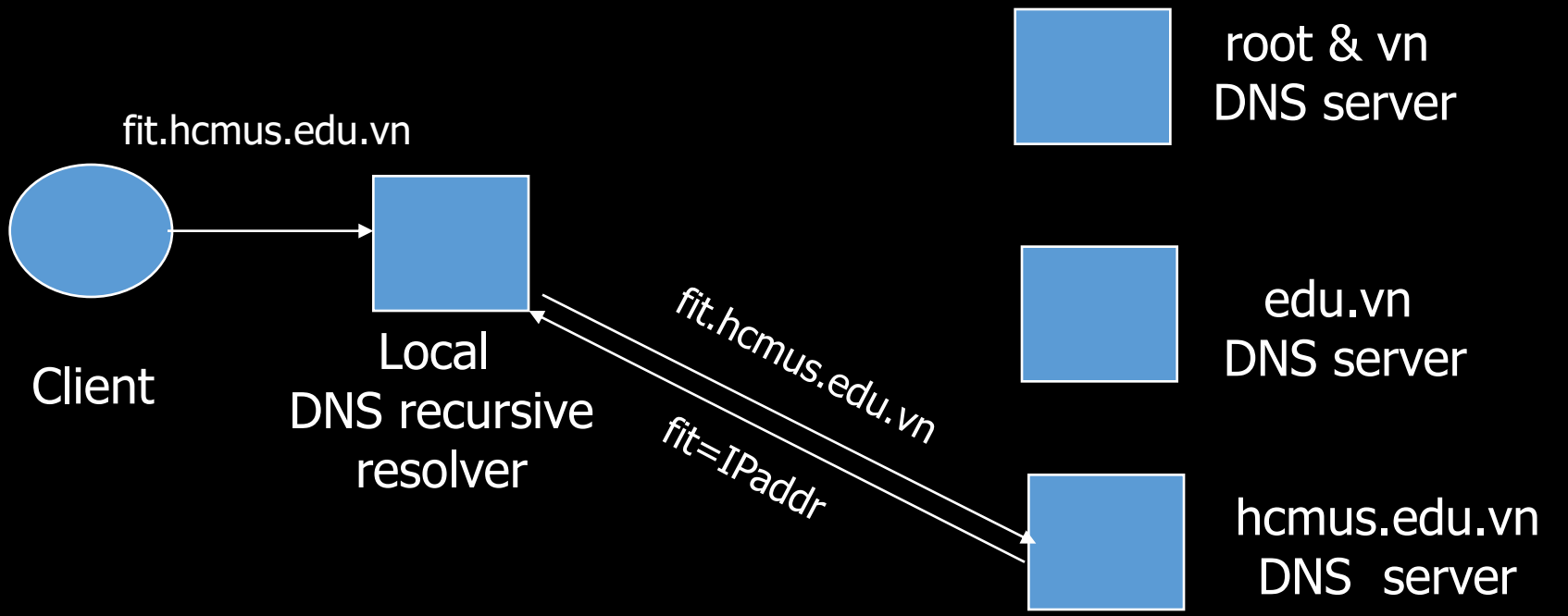


Feb 6, 2007: DoS attack on root DNS servers

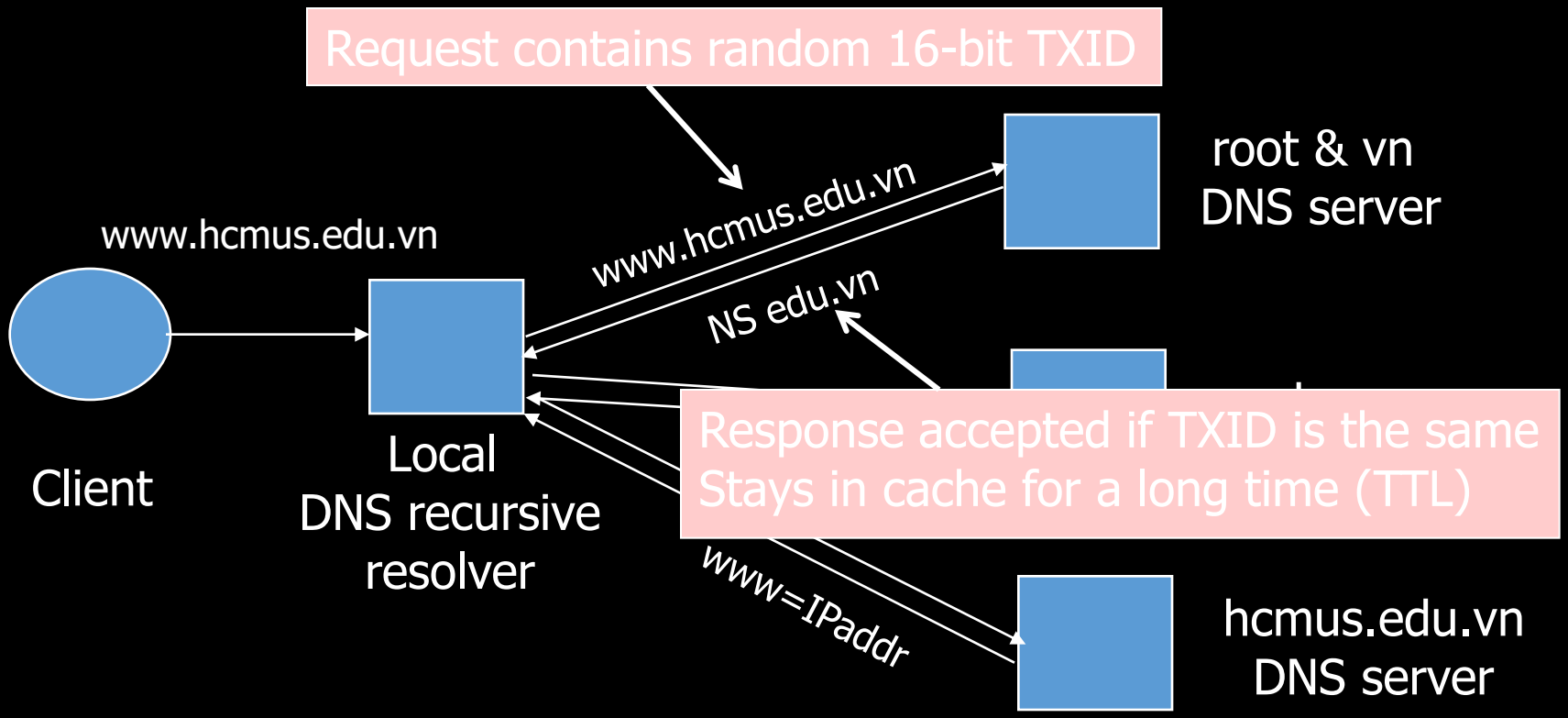
DNS Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Other queries may reuse some parts of lookup
    - NS records for domains
- DNS negative queries are cached
  - Don’t have to repeat past mistakes
    - For example, misspellings
- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed with every record

Cached Lookup Example



DNS “Authentication”



Bailiwick Checking, since 1997

```
$ dig @ns1.example.com www.example.com
;; ANSWER SECTION:
www.example.com.     120       IN     A     192.168.1.10

;; AUTHORITY SECTION:
example.com.         86400     IN     NS    ns1.example.com.
example.com.         86400     IN     NS    ns2.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.     604800    IN     A     192.168.2.20
ns2.example.com.     604800    IN     A     192.168.3.30
www.linuxjournal.com. 43200    IN     A     66.240.243.113
```

- This just means that any records that aren't in the same domain of the question are ignored.

Kaminsky's exploit 2008

- Typical DNS query and response to unknown www.example.com

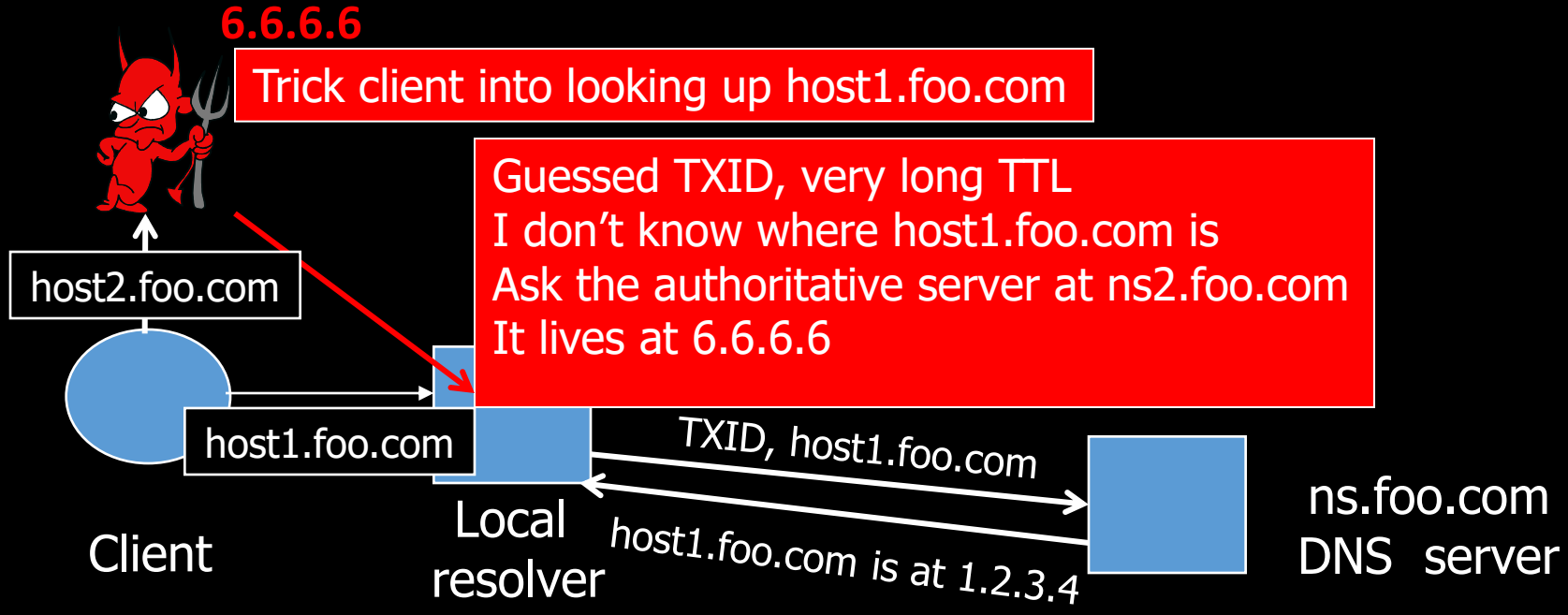
```
$ dig @ns1.example.com www.example.com
;; ANSWER SECTION:
www.example.com.    120     IN    A    192.168.1.10

;; AUTHORITY SECTION:
example.com.        86400   IN    NS   ns1.example.com.
example.com.        86400   IN    NS   ns2.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.    604800  IN    A    192.168.2.20
ns2.example.com.    604800  IN    A    192.168.3.30
```

DNS Spoofing

Several opportunities to win the race
If attacker loses, has to wait until TTL expires
… but can try again with host2.foo.com, host3.foo.com, etc.

Exploiting Recursive Resolving

[Kaminsky]



If attacker wins, all future DNS requests will go to 6.6.6.6
The cache is now poisoned… for a very long time!
No need to win future races!

Kaminsky's exploit

- asks for the IP address of doesnotexist.example.com
- Attacker replies:

```
$ dig doesnotexist.example.com
;; ANSWER SECTION:
doesnotexist.example.com.   120    IN   A     10.10.10.10

;; AUTHORITY SECTION:
example.com.                86400  IN   NS    www.example.com.

;; ADDITIONAL SECTION:
www.example.com.            604800 IN   A     10.10.10.20
```

Pharming

- Many anti-phishing defenses rely on DNS
- Can bypass them by poisoning DNS cache and/or forging DNS responses
  - Browser: give me the address of www.paypal.com
  - Attacker: sure, it's 6.6.6.6 (attacker-controlled site)
- Dynamic pharming
  - Provide bogus DNS mapping for a trusted server, trick user into downloading a malicious script
  - Force user to download content from the real server, temporarily provide correct DNS mapping
  - Malicious script and content have the same origin!

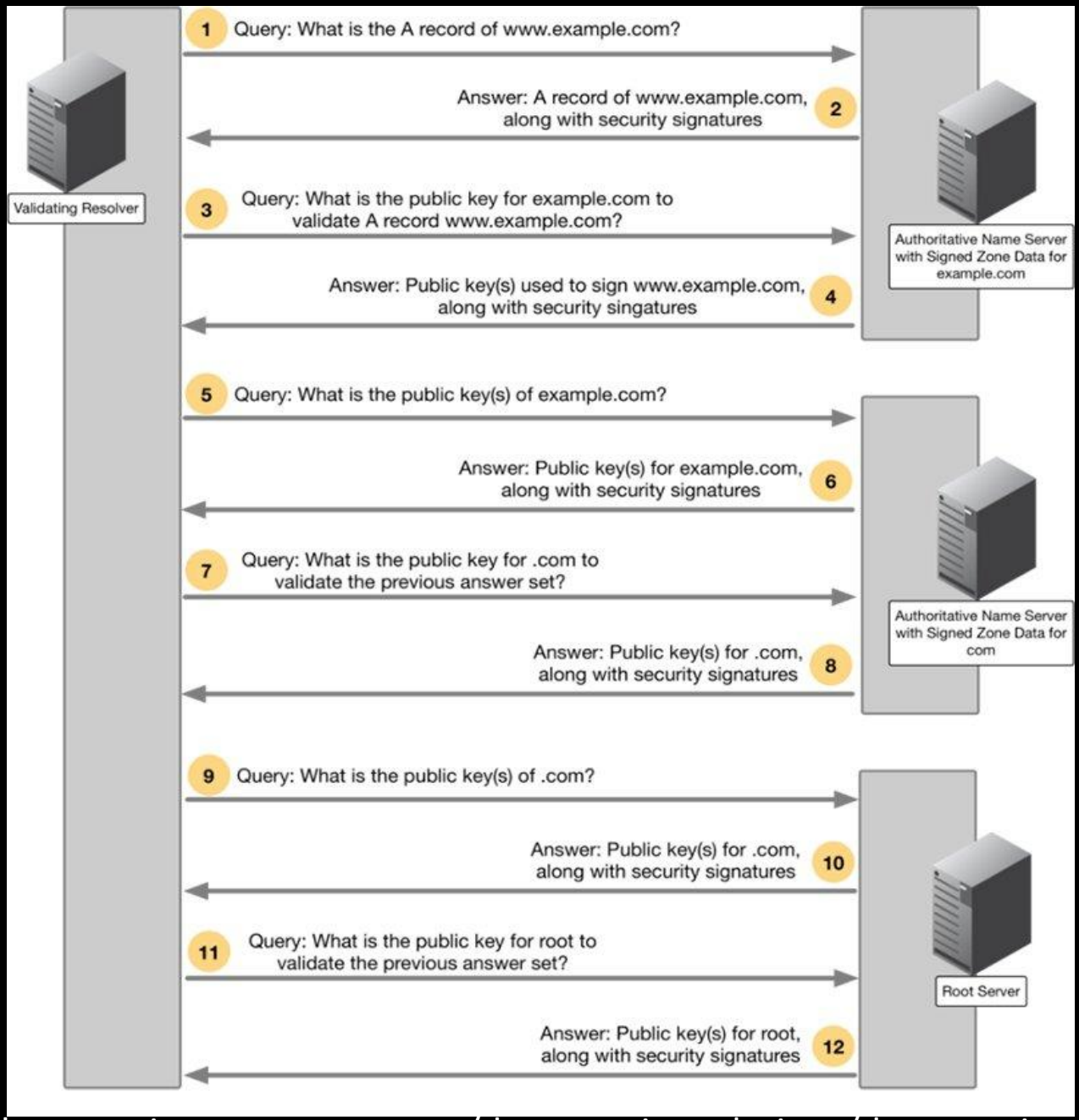
Solving the DNS Spoofing Problem

- Long TTL for legitimate responses
  - Does it really help?
- Randomize port in addition to TXID
  - 32 bits of randomness, makes it harder for attacker to guess TXID
- DNSSEC
  - Cryptographic authentication of host-address mappings

DNSSEC

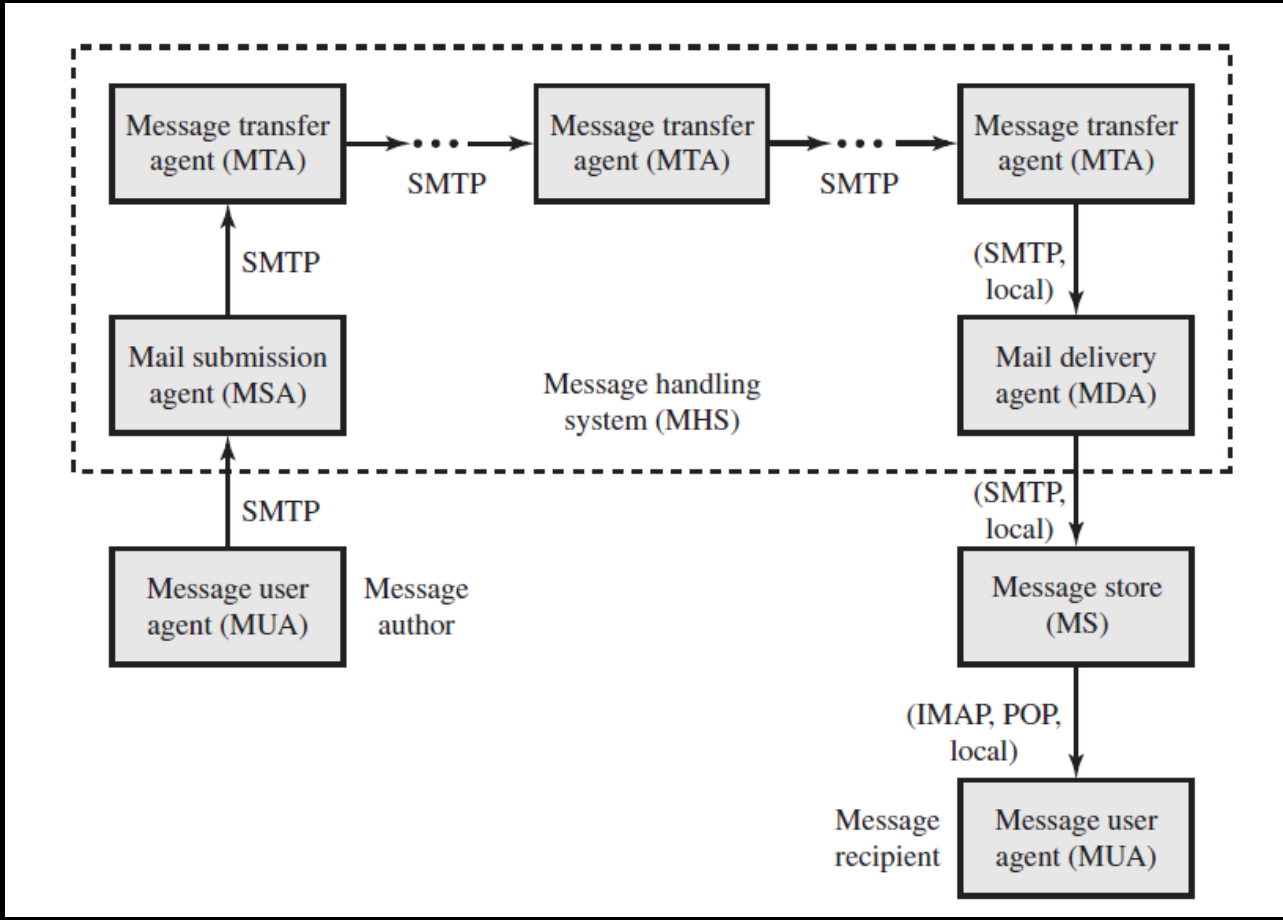
- Goals: authentication and integrity of DNS requests and responses
- PK-DNSSEC (public key)
  - DNS server signs its data (can be done in advance)
  - How do other servers learn the public key?
- SK-DNSSEC (symmetric key)
  - Encryption and MAC: $E_k(m, MAC(m))$
  - Each message contains a nonce to avoid replay
  - Each DNS node shares a symmetric key with its parent
  - Zone root server has a public key (hybrid approach)

DNSSEC



https://www.infoblox.com/dns-security-resource-center/dns-security-solutions/dns-security-solutions-dns-security-extensions-dnssec/

Internet Mail Architecture



Internet Mail Architecture

- Message User Agent (MUA)
  - Works on behalf of user actors and user applications.
  - The author use sender MUA to formats a message and performs initial submission into the MHS via a MSA.
  - The recipient MUA processes received mail for storage and/or display to the recipient user
- Mail submission agent (MSA)
  - Accepts the message submitted by an MUA and enforces the policies of the hosting domain and the requirements of Internet standards
- Message transfer agent (MTA)
  - Relays mail for one application-level hop.
- Mail delivery agent (MDA)
  - Responsible for transferring the message from the MHS to the MS.
- Message store (MS)

Domainkeys Identified Mail (DKIM)

- DomainKeys Identified Mail (DKIM) is a specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message in the mail stream.
- Message recipients (or agents acting in their behalf) can verify the signature by querying the signer's domain
- DKIM is a proposed Internet Standard (RFC 4871: *DomainKeys Identified Mail (DKIM) Signatures*).

DKIM Principle

1. S/MIME depends on both the sending and receiving users employing S/MIME. For almost all users, the bulk of incoming mail does not use S/MIME, and the bulk of the mail the user wants to send is to recipients not using S/MIME.

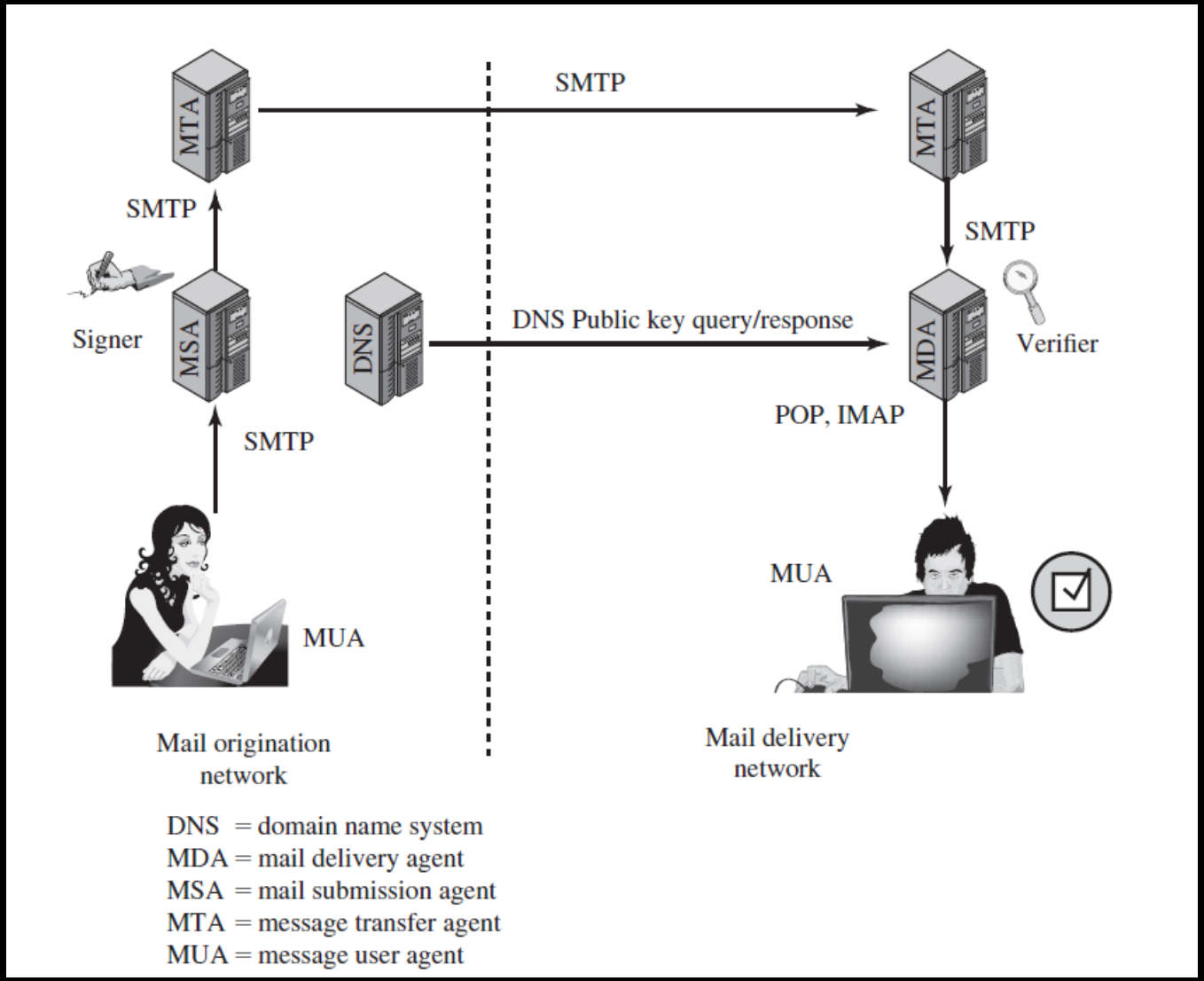
2. S/MIME signs only the message content. Thus, RFC 5322 header information concerning origin can be compromised.

3. DKIM is not implemented in client programs (MUAs) and is therefore transparent to the user; the user need take no action.

4. DKIM applies to all mail from cooperating domains.

5. DKIM allows good senders to prove that they did send a particular message and to prevent forgers from masquerading as good senders.

DKIM Operations



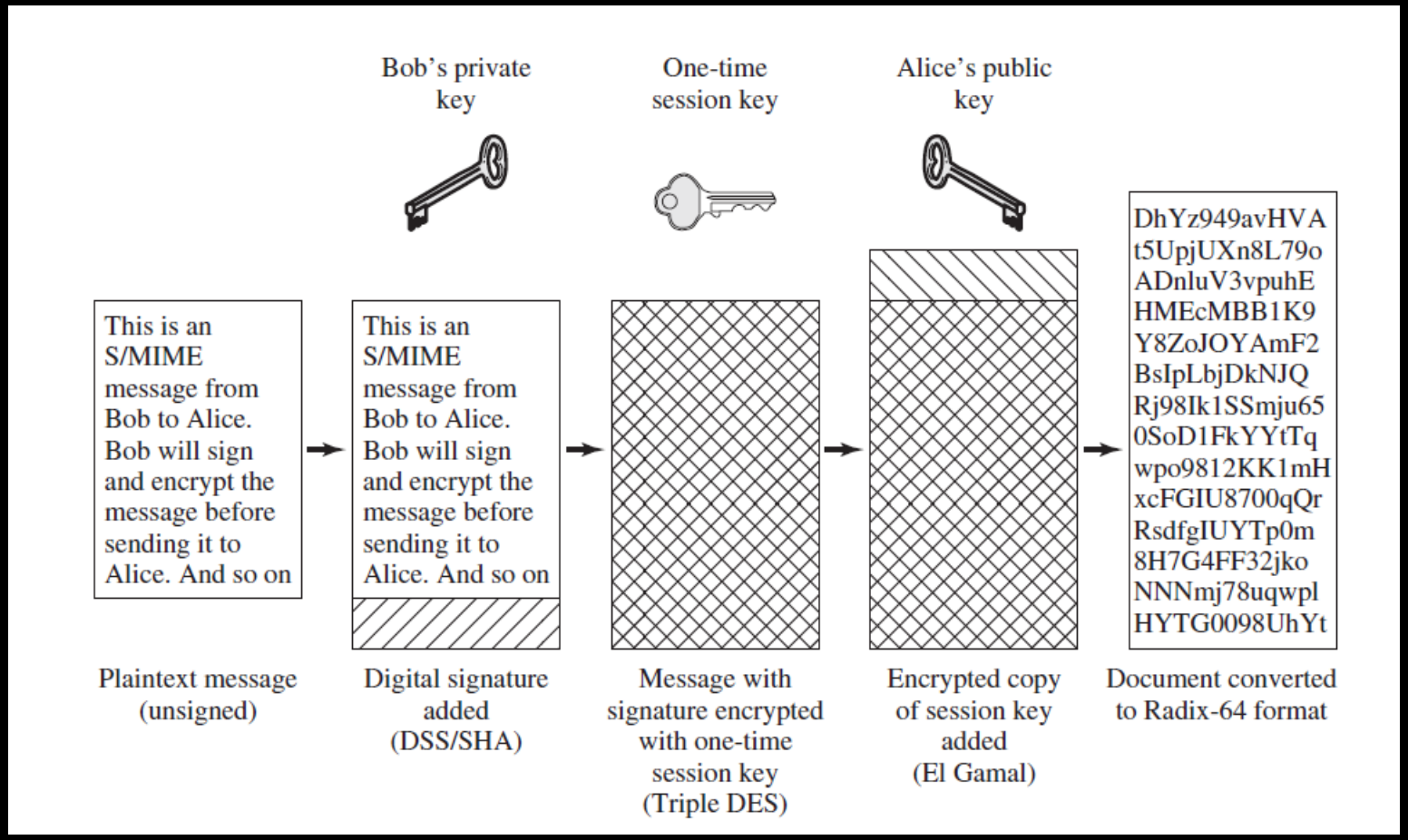
S/MIME

- MIME is an extension to the old RFC 822 specification of an Internet mail format.
- MIME provides a number of new header fields that define information about the body of the message.
- MIME defines a number of content formats, which standardize representations for the support of multimedia e-mail. Examples include text, image, audio, and video

S/MIME

- S/MIME is defined as a set of additional MIME content types and provides the ability to sign and/or encrypt e-mail messages
  - **Enveloped data**: This function consists of encrypted content of any type and encrypted-content encryption keys for one or more recipients.
  - **Signed data**: A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.
  - **Clear-signed data**: As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base64. As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature.
  - **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

S/MIME

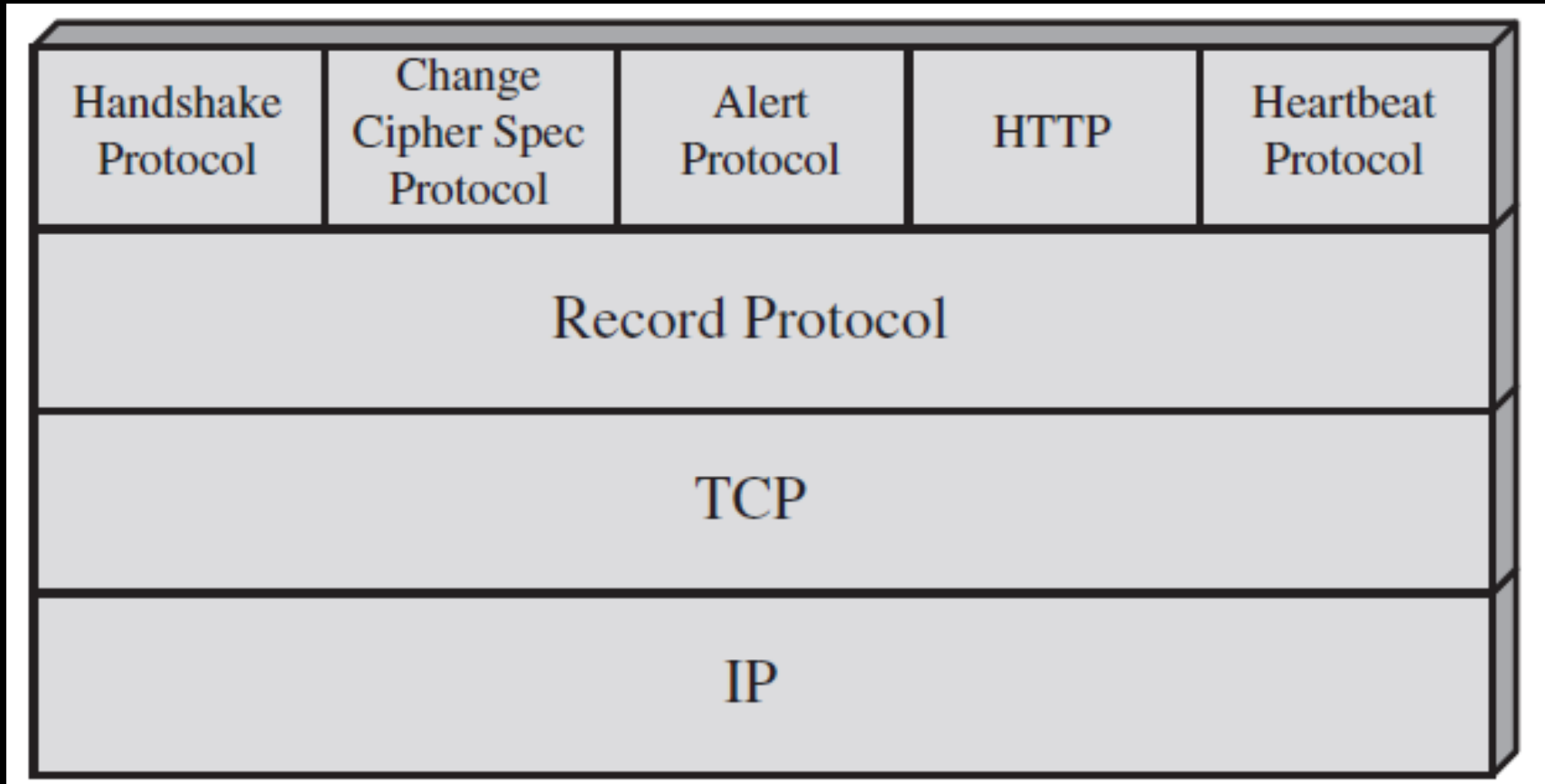

Bob's private key
One-time session key
Alice's public key
This is an S/MIME message from Bob to Alice. Bob will sign and encrypt the message before sending it to Alice. And so on
This is an S/MIME message from Bob to Alice. Bob will sign and encrypt the message before sending it to Alice. And so on
DhYz949avHVA
t5UpjUXn8L79o
ADnluV3vpuhE
HMEcMBB1K9
Y8ZoJOYAmF2
BsIpLbjDkNJQ
Rj98Ik1SSmju65
0SoD1FkYYtTq
wpo9812KK1mH
xcFGIU8700qQr
RsdfgIUYTp0m
8H7G4FF32jko
NNNmj78uqwpl
HYTG0098UhYt
Plaintext message (unsigned)
Digital signature added (DSS/SHA)
Message with signature encrypted with one-time session key (Triple DES)
Encrypted copy of session key added (El Gamal)
Document converted to Radix-64 format

S/MIME: Public-Key Certificates

- S/MIME uses certificates that conform to the international standard X.509v3.

SSL/TLS

- One of the most widely used security services is the Secure Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS).
- TLS is designed to make use of TCP to provide a reliable end-to-end secure service. TLS is not a single protocol but rather two layers of protocols


Handshake Protocol
Change Cipher Spec Protocol
Alert Protocol
HTTP
Heartbeat Protocol
Record Protocol
TCP
IP

SSL connection – SSL session

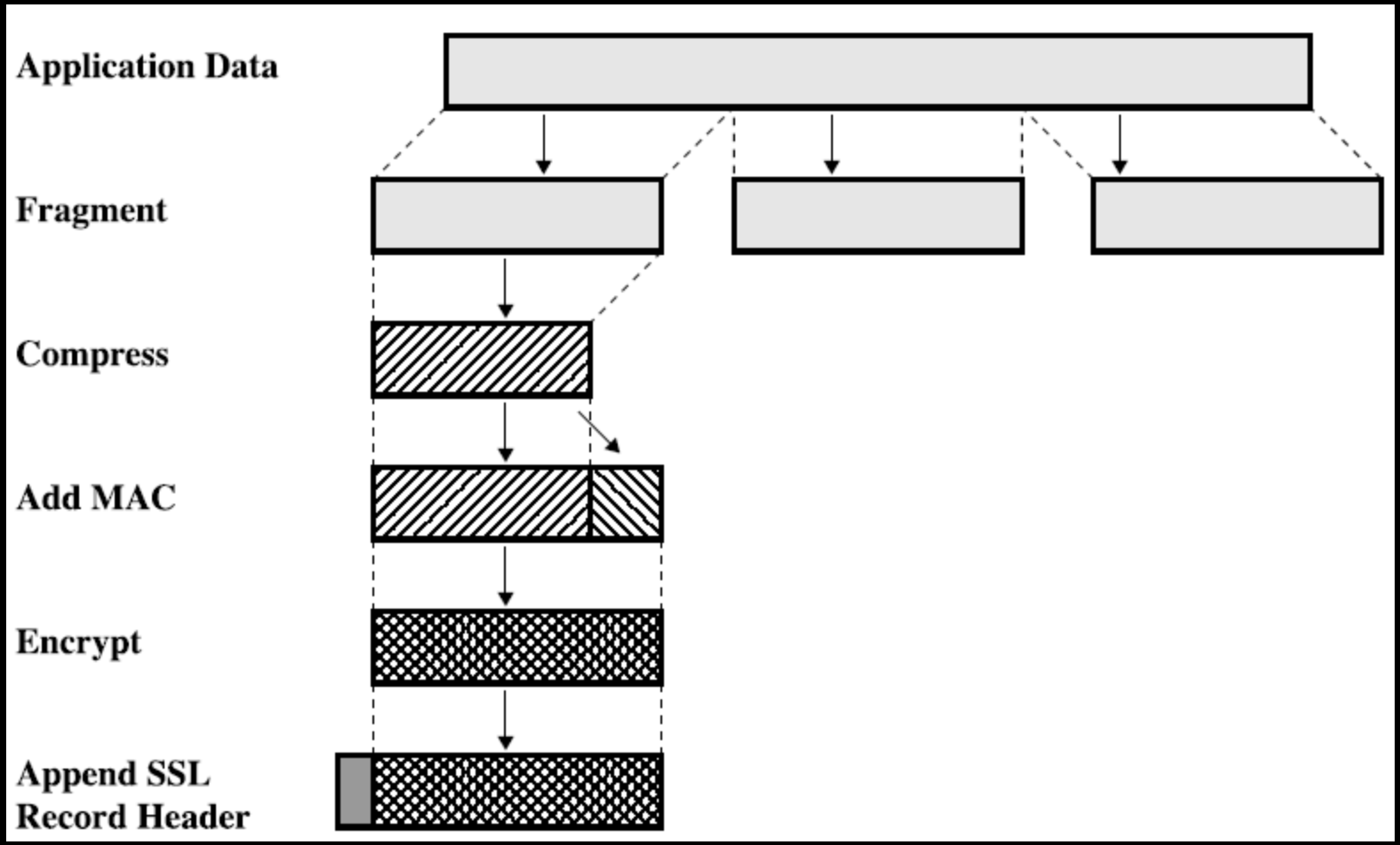
- Two important TLS concepts are the TLS session and the TLS connection, which are defined in the specification as follows:
  - **Connection**: A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For TLS, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.
  - **Session**: A TLS session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection

SSL Record Protocol

The SSL Record Protocol provides two services for SSL connections:

- **Confidentiality**: The Handshake Protocol defines a shared secret key that is used for symmetric encryption of SSL payloads.
- **Message integrity**: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

SSL Record Protocol



Change Cipher Spec Protocol

- It use the TLS Record Protocol
- This protocol consists of a single message, which consists of a single byte with the value 1.
- The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.

Alert Protocol

- convey TLS-related alerts to the peer entity
- alert messages are compressed and encrypted, as specified by the current state
- Each message in this protocol consists of two bytes. The first byte takes the value warning(1) or fatal(2) to convey the severity of the message
- If the level is fatal, TLS immediately terminates the connection

SSL Handshaking Protocol

- The most complex part of TLS
- SSL Handshake consist 4 phase:

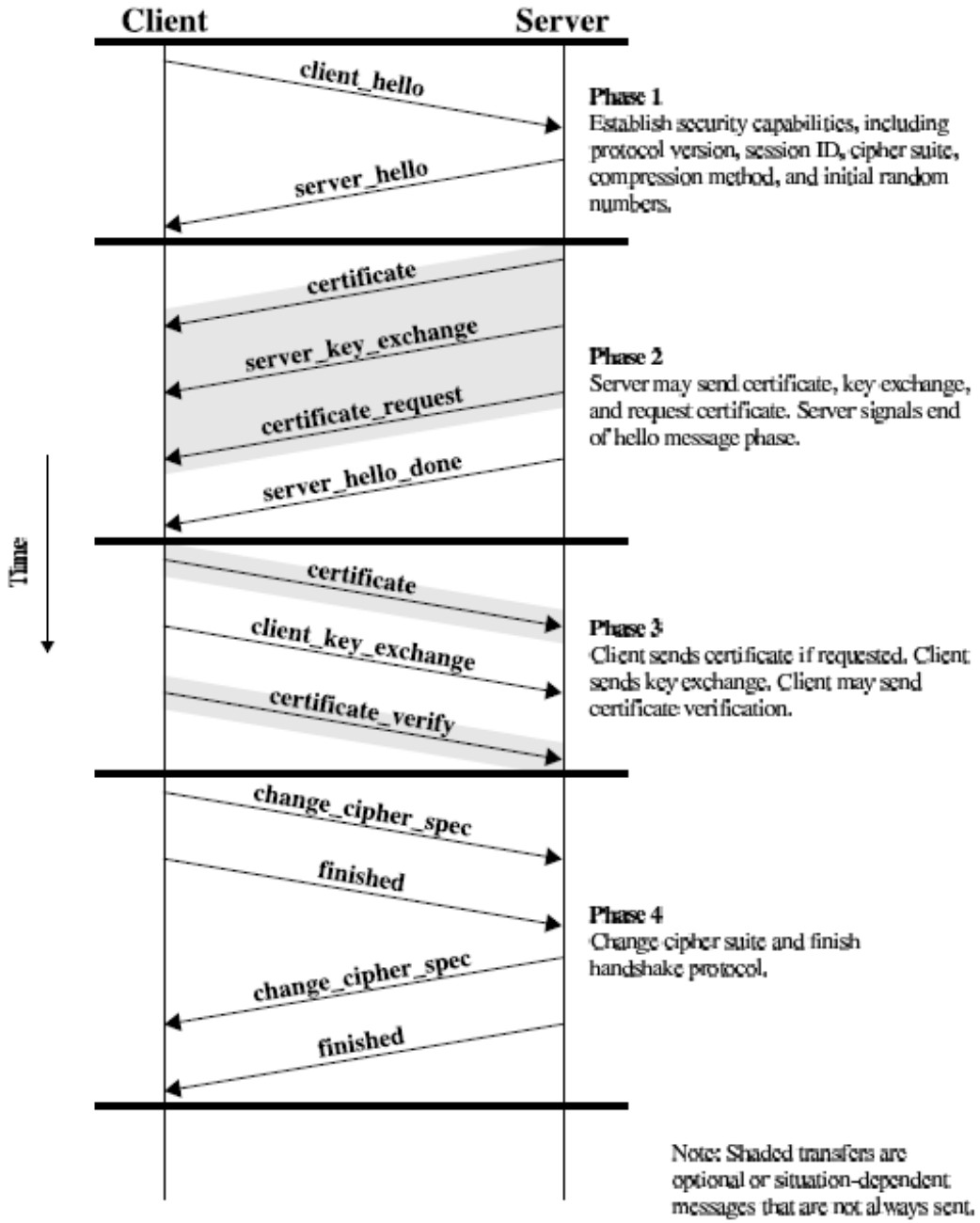
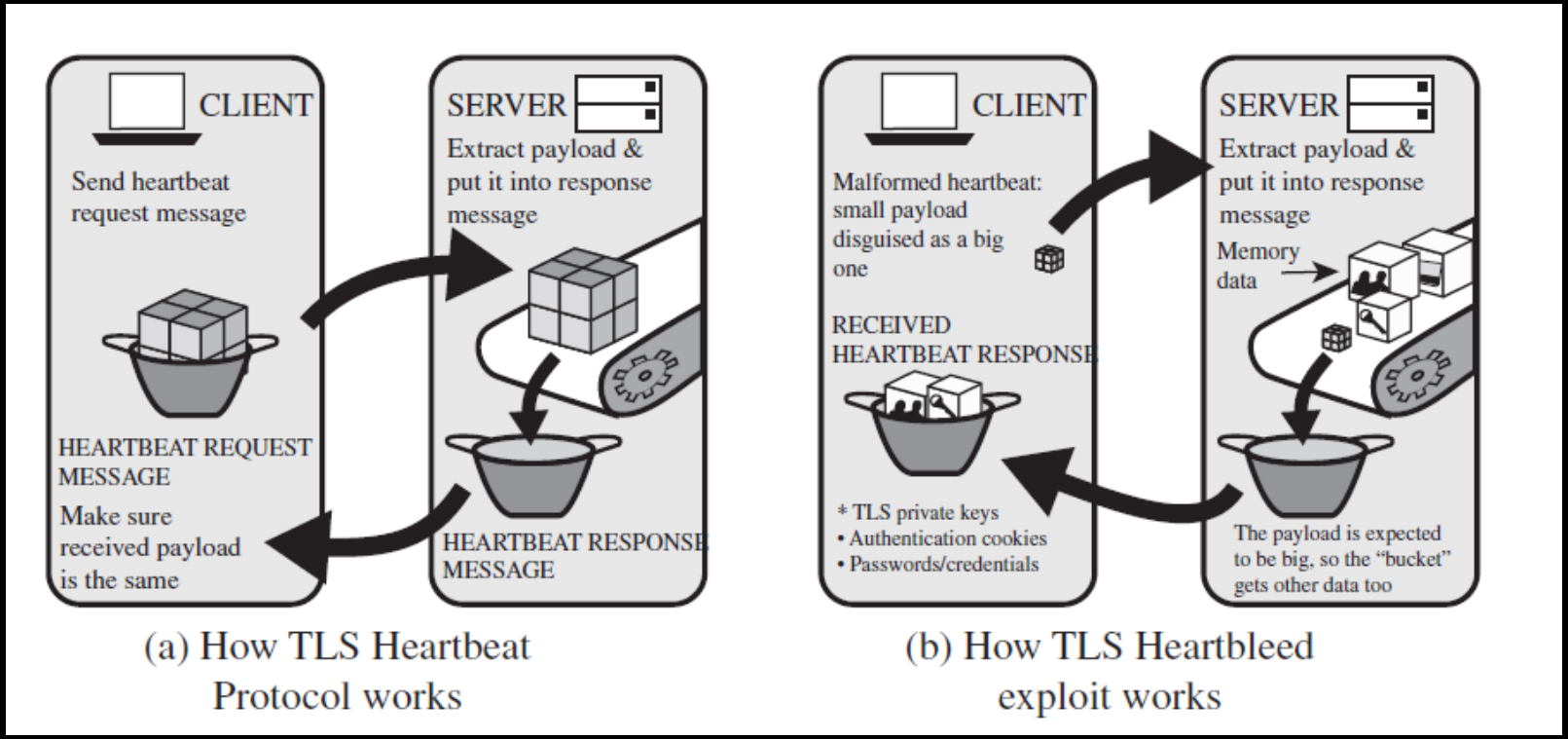


Figure 17.6 Handshake Protocol Action

Heartbeat Protocol

- The Heartbeat Protocol runs on the top of the TLS Record Protocol and consists of two message types: heartbeat_request and heartbeat_response



SSL/TLS Attacks

- Attacks on the Handshake Protocol.
- Attacks on the record and application data protocols:
  - BEAST
  - CRIME
  - …
- Attacks on the PKI:
  - Checking the validity of X.509 certificates is an activity subject to a variety of attacks.
- Other attacks:
  - DoS attack
  - Heartbleed attack
  - …

IPSec

- When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPsec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPsec is implemented in the firewall or router. Even if IPsec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPsec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.
- IPsec can provide security for individual users if needed. This is useful foroff-site workers and for setting up a secure virtual subnetwork within an organization for sensitive applications.

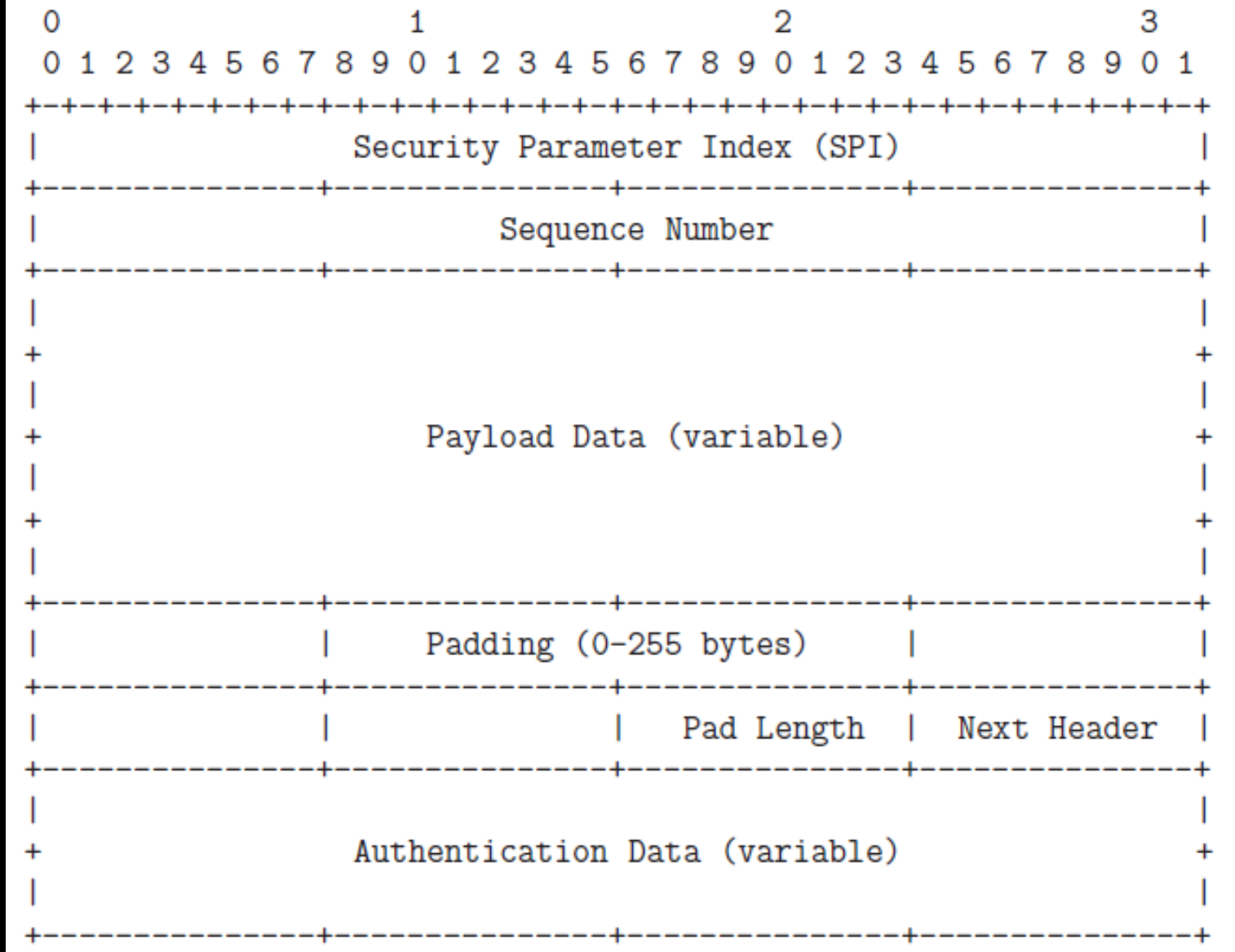
IPSec

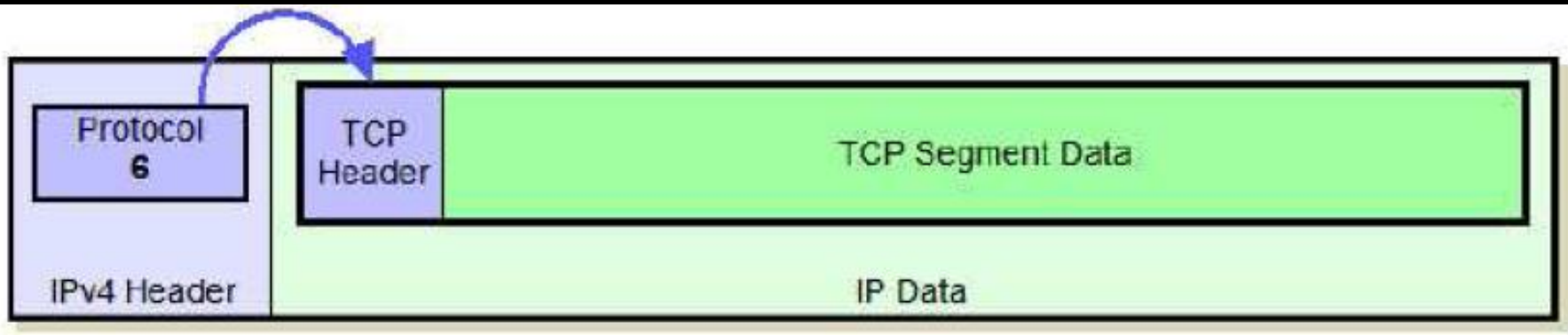
- IPsec provides two main functions:
  - a combined authentication/encryption function called Encapsulating Security Payload (ESP)
  - Key exchange function: Key management is provided by the Internet Key Exchange standard, IKEv2
- Note: There is also an authentication- only function, implemented using an Authentication Header (AH). Because message authentication is provided by ESP, the use of AH is deprecated

ESP: Security Associations

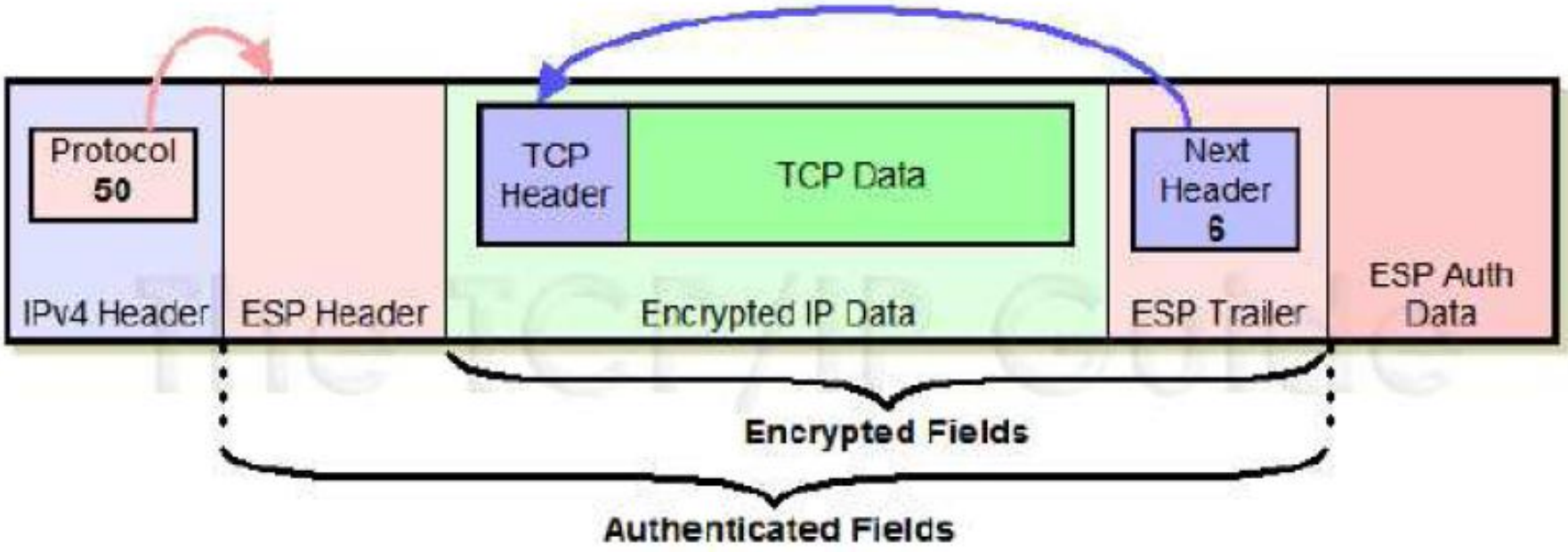
- A key concept that appears in both the authentication and confidentiality mechanisms for IP is the security association (SA).
- An association is a one-way relationship between a sender and a receiver
- An SA is uniquely identified by three parameters:
  - **Security parameter index (SPI)** A bit string assigned to this SA and having local significance only
  - **IP destination address**
  - **Protocol identifier**

Encapsulating Security Payload (ESP) Header

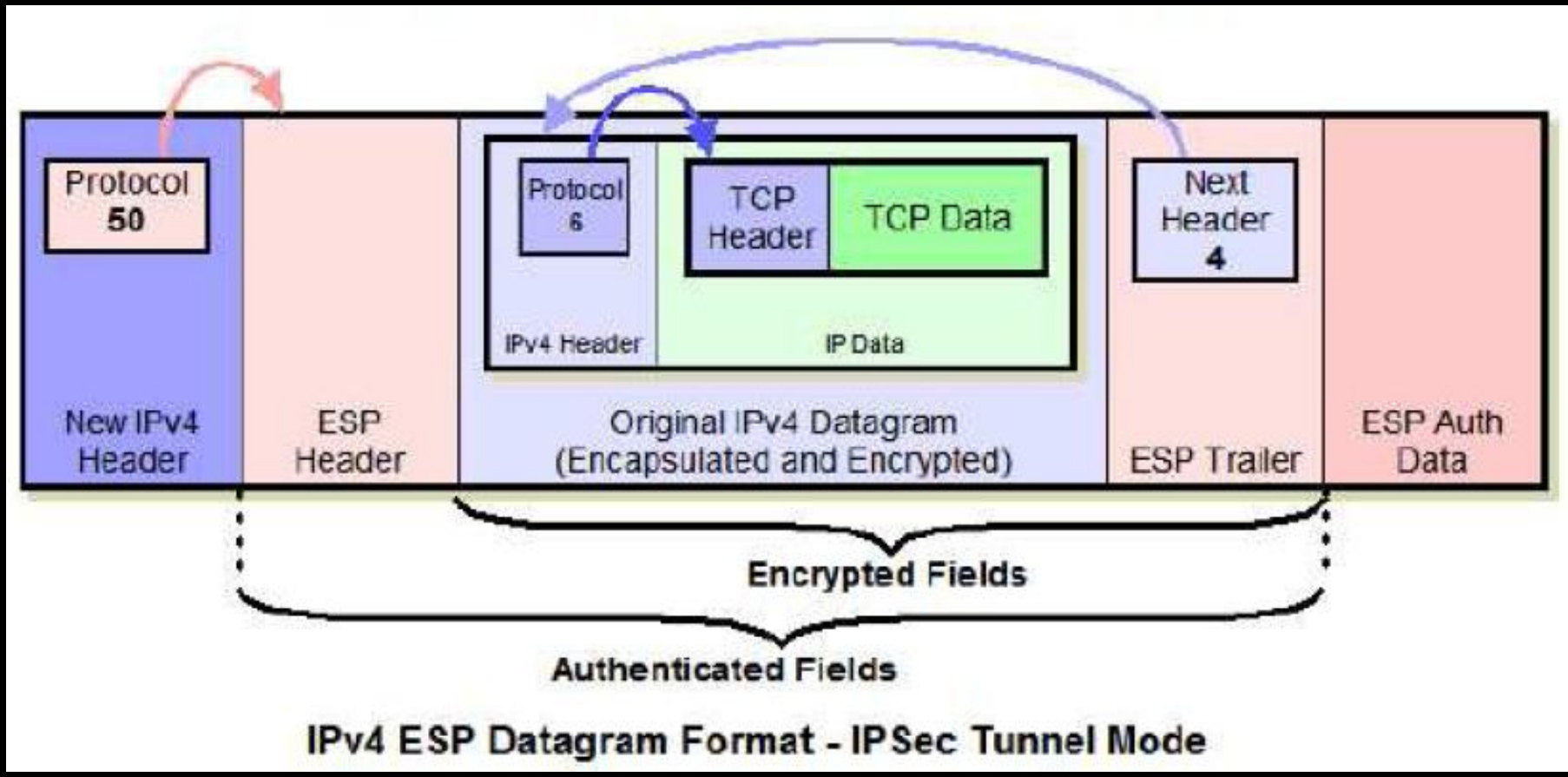




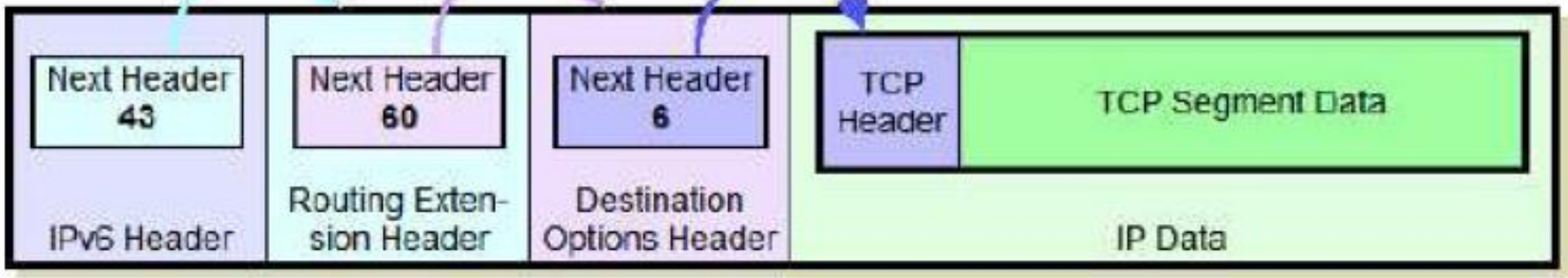
Original IPv4 Datagram Format



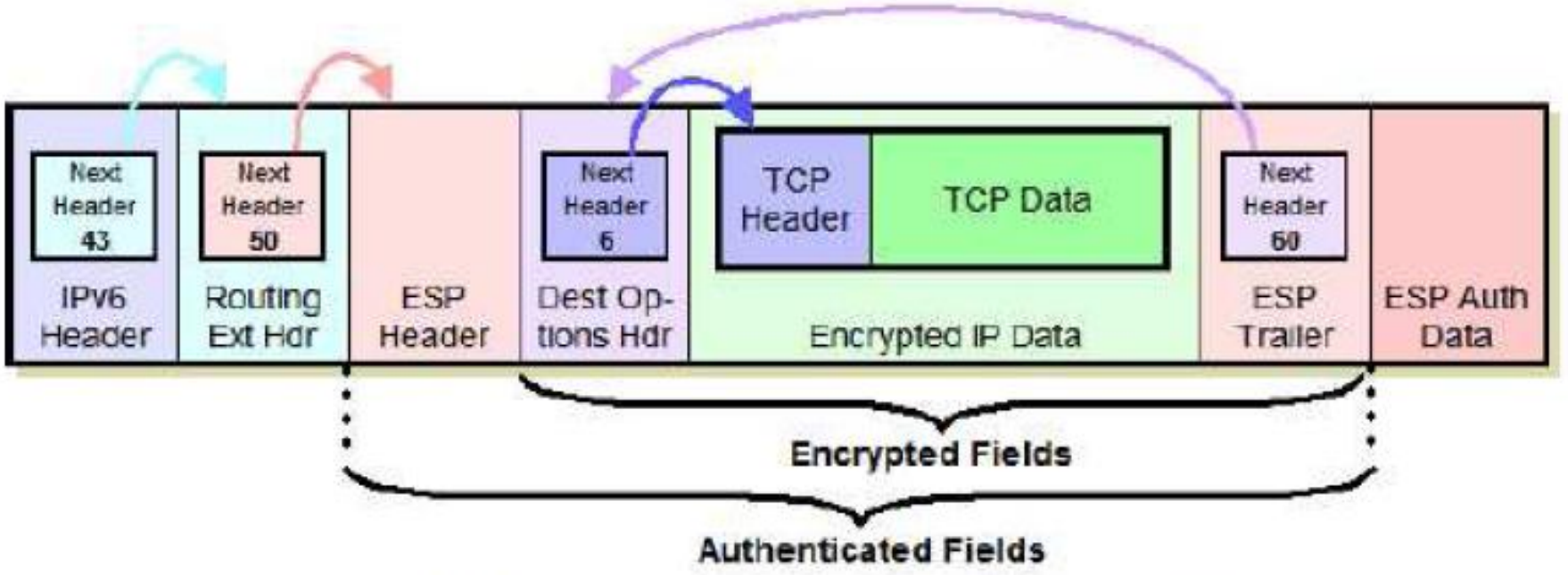
IPv4 ESP Datagram Format - IPSec Transport Mode



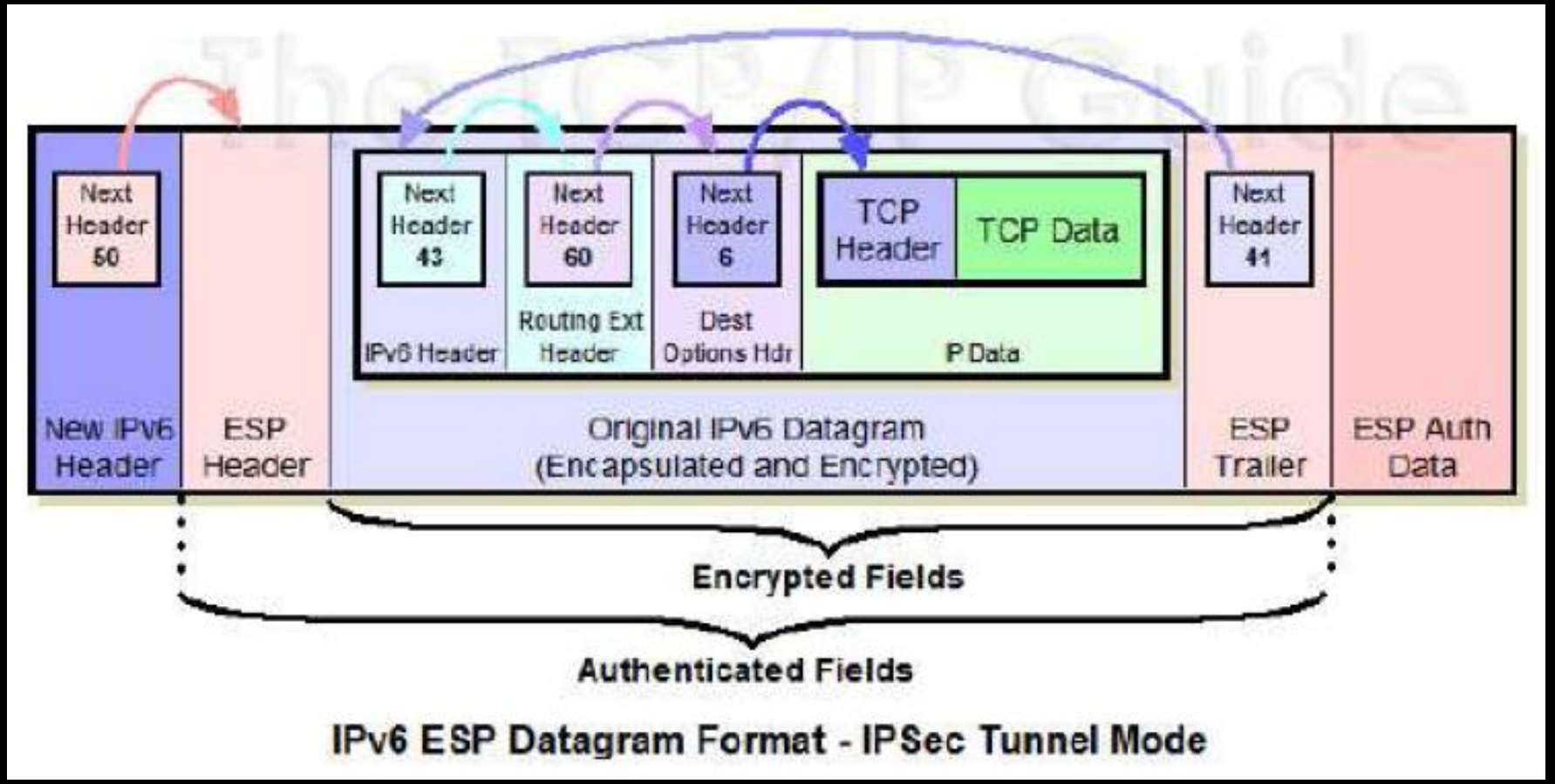
IPv4 ESP Datagram Format - IPSec Tunnel Mode



Original IPv6 Datagram Format (Including Routing Extension Header and Destination-Specific Destination Options Extension Header)



IPv6 ESP Datagram Format - IPSec Transport Mode



IPv6 ESP Datagram Format - IPSec Tunnel Mode