

UNIT 7

Network Security

(5 Hours) — Information Security (CACS459)

Units 2 and 6 covered the building blocks — cryptographic algorithms and the threats they defend against. This unit shows how those building blocks are actually deployed in real network protocols to protect data as it travels: securing individual emails (S/MIME, PGP), securing entire application sessions over the web (SSL/TLS), securing traffic at the network (IP) layer itself (IPSec), and controlling which traffic is allowed to enter or leave a network at all (firewalls).

7.1 Overview of Network Security

Network security protocols can be deployed at different layers of the protocol stack, and the layer chosen affects both what is protected and how transparent the protection is to applications above it. Lower-layer protection (e.g., IPSec at the network layer) protects everything running above it automatically, without requiring individual applications to be modified; higher-layer protection (e.g., PGP at the application layer) is more application-specific but can be tailored precisely to that application's needs.

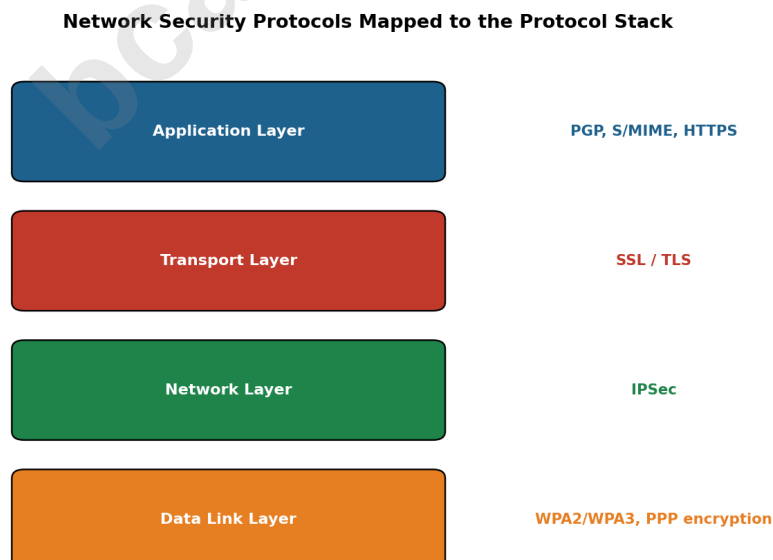


Fig. 7.1 — Major network security protocols mapped onto the layer of the protocol stack at which they operate.

Layer	What is Protected	Example Protocol
Application	Specific application data (email content, web forms)	PGP, S/MIME, HTTPS application data
Transport	An entire application session/connection, transparently to the application	SSL / TLS
Network	Every packet between two hosts or gateways, transparently to all applications and transport protocols above it	IPSec
Data Link	Traffic over a single physical/logical link (e.g., a Wi-Fi connection)	WPA2/WPA3, PPP encryption

Note: As a general security-engineering principle: protecting traffic at a lower layer is broader (covers more) but less flexible, while protecting it at a higher layer is more precise but must be implemented separately for every application that needs it.

7.2 Email Security: S/MIME and Pretty Good Privacy (PGP)

Ordinary email (SMTP) provides no security at all by default — messages travel in plaintext and can be read or altered by anyone with access to the mail servers or network links in between. Both S/MIME and PGP add confidentiality, integrity, authentication, and non-repudiation to email, using broadly the same underlying idea: a hybrid of symmetric and asymmetric cryptography, exactly as described in Unit 2.

7.2.1 Pretty Good Privacy (PGP)

PGP was created by Phil Zimmermann in 1991 and became the first widely available email encryption tool for ordinary users. PGP combines several services:

- **Confidentiality:** the message is encrypted with a randomly generated, one-time session key using a fast symmetric cipher (such as AES); the session key itself is then encrypted with the recipient's RSA public key, so only the recipient's matching private key can recover it.
- **Authentication / Digital Signature:** the sender computes a hash (e.g., SHA) of the message and encrypts that hash with their own RSA private key, producing a digital signature that the recipient can verify using the sender's public key.
- **Compression:** the message is typically compressed before encryption, both to save bandwidth and to make cryptanalysis slightly harder (compressed data has less exploitable redundancy).
- **Email compatibility:** since the resulting ciphertext is raw binary data, PGP applies radix-64 (Base64) conversion to make it safe to transmit through email systems that expect plain ASCII text.

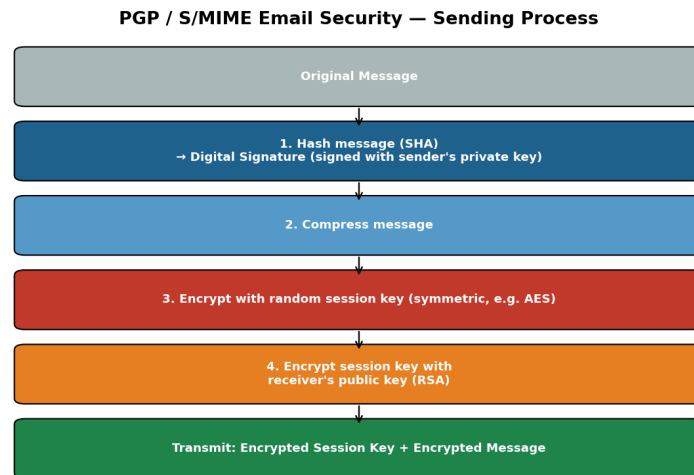


Fig. 7.2 — The PGP/S-MIME message-sending process: sign with a hash and the sender's private key, compress, encrypt with a random symmetric session key, then encrypt that session key with the recipient's public key.

PGP uses a decentralised 'Web of Trust' model for managing public keys: rather than relying on a centralised certificate authority, users sign each other's public keys directly to vouch for their authenticity, building a distributed trust network.

7.2.2 Secure/Multipurpose Internet Mail Extensions (S/MIME)

S/MIME extends the standard MIME email format with cryptographic security services, very similar in spirit to PGP (encryption, digital signatures, hybrid symmetric/asymmetric cryptography), but with two key practical differences:

Aspect	PGP	S/MIME
Trust Model	Decentralised 'Web of Trust' — users vouch for each other's keys	Centralised: relies on formal X.509 digital certificates issued by a trusted Certificate Authority (CA)
Typical Use Case	Historically popular with individual/technical users for personal email encryption	Widely built directly into corporate email clients (e.g., Outlook), favoured in enterprise environments
Standardisation Body	Originally independent; later an IETF open standard (OpenPGP)	An IETF standard, building on the existing X.509 PKI used across the web

Note: Both PGP and S/MIME rely on the same underlying mathematics from Unit 2 (RSA, hashing, symmetric ciphers) — their real difference lies in how trust in public keys is established and managed, not in the cryptography itself.

7.3 Secure Socket Layer (SSL) and Transport Layer Security (TLS)

SSL (developed by Netscape) and its successor TLS (the IETF standardised version) operate at the transport layer, providing a secure channel that any application protocol can run over — this is exactly what turns ordinary HTTP into HTTPS, and also secures protocols like FTP and SMTP when run 'over TLS'. TLS provides three core security services: server (and optionally client) authentication, confidentiality, and data integrity.

7.3.1 The TLS Handshake

Before any application data is exchanged, the client and server perform a handshake to agree on cryptographic parameters and establish a shared session key:

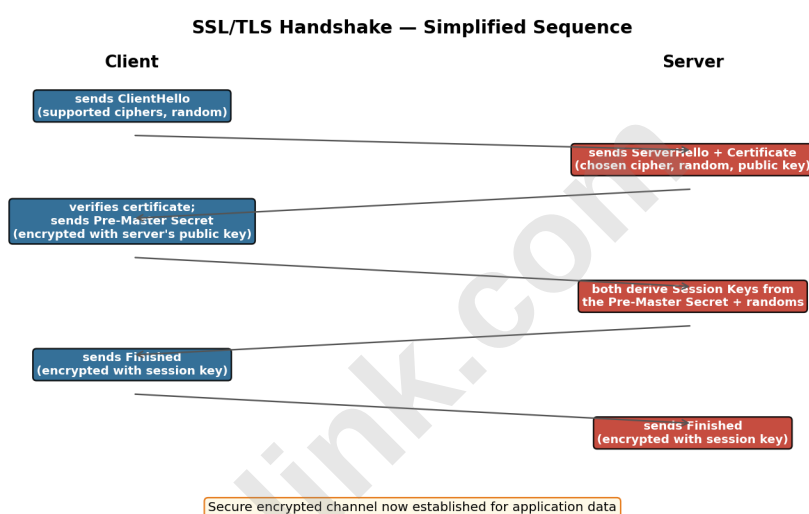


Fig. 7.3 — A simplified TLS handshake: the client and server exchange hello messages and a certificate, derive a shared session key from a securely exchanged pre-master secret, then confirm with a 'Finished' message before exchanging application data.

- **ClientHello:** the client proposes the TLS version, a list of supported cipher suites, and a random value.
- **ServerHello + Certificate:** the server selects a cipher suite, sends its own random value, and presents its digital certificate (containing its public key, verifiable via a trusted Certificate Authority) for authentication.
- **Key Exchange:** the client generates a pre-master secret and sends it to the server, encrypted using the server's public key (or, in modern TLS, the two sides perform an ephemeral Diffie-Hellman key exchange instead, for 'forward secrecy').
- **Session Key Derivation:** both sides independently compute the same set of symmetric session keys from the pre-master secret and the two random values exchanged earlier.
- **Finished:** each side sends a 'Finished' message, encrypted with the newly derived session key, confirming the handshake succeeded; all subsequent application data is then protected using fast symmetric encryption with the shared session key.

Note: This is a textbook example of hybrid cryptography exactly as introduced in Unit 2: slow asymmetric cryptography (RSA or Diffie-Hellman) is used only briefly, during the handshake, to safely establish a key; all the bulk application data afterward is protected using fast symmetric encryption (such as AES) with that shared key.

7.4 IP Security (IPSec)

IPSec is a suite of protocols that adds authentication and encryption directly at the IP (network) layer, meaning it can protect any traffic running over IP, completely transparently to the applications and transport protocols above it. IPSec is the technology underlying most enterprise VPNs (Virtual Private Networks).

7.4.1 IPSec Core Protocols

- **Authentication Header (AH):** provides data integrity and authentication of the packet's origin (it can prove the packet wasn't tampered with and really came from the claimed sender), but does not provide confidentiality (no encryption).
- **Encapsulating Security Payload (ESP):** provides confidentiality (encryption) in addition to integrity and authentication — ESP is by far the more commonly used protocol in practice, since it offers strictly more protection than AH.
- **Internet Key Exchange (IKE):** automatically negotiates and establishes the shared security parameters (algorithms, keys) that AH/ESP will use, so administrators do not need to manually configure keys on every device.

7.4.2 IPSec Modes of Operation

IPSec Transport Mode (Host-to-Host)



Only the payload is protected; original IP header stays in the clear

IPSec Tunnel Mode (Gateway-to-Gateway / VPN)



The entire original packet is encapsulated and protected inside a new outer packet

Fig. 7.4 — Transport mode protects only the payload of each packet between two hosts directly; tunnel mode wraps the entire original packet inside a new one, typically used between two security gateways for a site-to-site VPN.

Mode	What is Protected	Typical Use Case
Transport Mode	Only the payload (data) of the original IP packet; the original IP header is left visible/unprotected	End-to-end protection directly between two individual hosts
Tunnel Mode	The entire original IP packet (header and payload), encapsulated inside a brand-new outer IP packet	Gateway-to-gateway (site-to-site) VPNs, or remote-access VPNs connecting a host to a gateway

7.5 Firewalls and Their Types

A firewall is a system (hardware, software, or a combination) placed at a boundary between two networks of differing trust levels (most commonly between an organisation's internal network and the public internet), designed to control which traffic is permitted to cross that boundary, based on a configured set of security rules.

7.5.1 Types of Firewalls

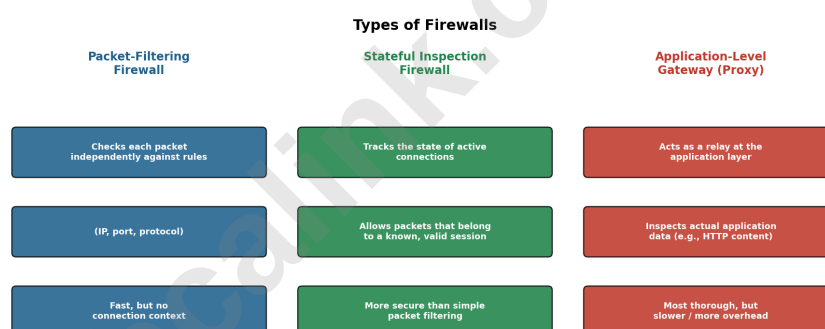


Fig. 7.5 — The three major firewall types, in increasing order of how deeply they inspect traffic: packet filtering (fastest, least context), stateful inspection (tracks connections), and application-level gateways (slowest, most thorough).

Type	How it Works	Strengths / Weaknesses
Packet-Filtering Firewall	Examines each packet in isolation against a static rule set (source/destination IP, port, protocol)	Fast and simple, but has no memory of past packets/connections, so it cannot detect attacks that span multiple packets
Stateful Inspection Firewall	Maintains a table of active, legitimate connections and only allows packets that genuinely belong to a recognised, established session	More secure than simple packet filtering, since it understands connection context, while remaining reasonably fast

Type	How it Works	Strengths / Weaknesses
Application-Level Gateway (Proxy)	Acts as an intermediary relay at the application layer, fully terminating and re-establishing each connection, and inspecting the actual application-layer content	Most thorough protection (can filter based on actual content, not just headers), but adds processing overhead and latency
Circuit-Level Gateway	Operates at the session layer, validating that a TCP handshake is legitimate before allowing a connection through, without inspecting the content of individual packets afterward	Faster than an application gateway, useful when content inspection is unnecessary but session validation is

7.5.2 Firewall Placement and the DMZ

Many organisations place a Demilitarized Zone (DMZ) between two firewalls: an external firewall facing the internet, and an internal firewall facing the organisation's trusted internal network. Public-facing servers (such as web servers or mail servers, which must be reachable from the internet) are placed inside the DMZ, where they are reachable from outside but are still firewalled away from the more sensitive internal network — so that even if a DMZ server is compromised, the attacker still faces the internal firewall before reaching genuinely critical internal systems.

Typical Firewall Placement — DMZ Architecture

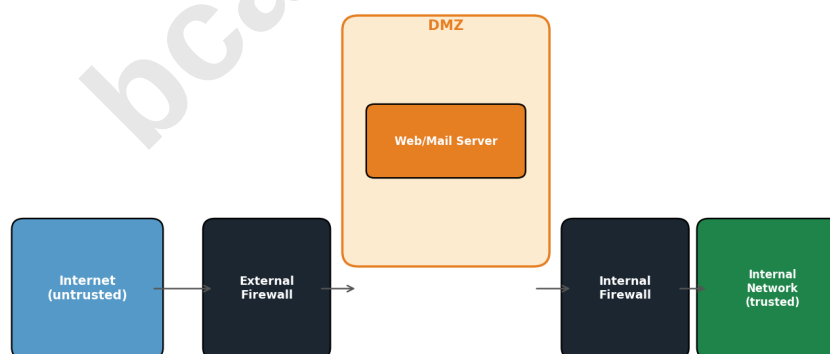


Fig. 7.6 — A typical DMZ architecture: public-facing servers sit between an external and an internal firewall, exposed to the internet but isolated from the trusted internal network.

Note: A firewall is only as effective as its rule configuration and cannot, by itself, stop attacks that don't cross the network boundary it guards — such as an infected USB drive, a malicious insider, or an attack carried out entirely within encrypted traffic that the firewall cannot inspect. This is exactly why firewalls are deployed as one layer within the broader 'Defense in Depth' strategy introduced in Unit 1.

Key Takeaways — Unit 7: Network Security

- ✓ Network security protocols can operate at different layers of the protocol stack — application (PGP, S/MIME), transport (SSL/TLS), network (IPSec), or data link — with lower layers offering broader, more transparent protection.
- ✓ PGP and S/MIME both secure email using hybrid cryptography (symmetric encryption for the message, asymmetric encryption for the session key, plus a digital signature for authentication); they differ mainly in trust model — PGP's decentralised Web of Trust versus S/MIME's centralised Certificate-Authority-based PKI.
- ✓ SSL/TLS secures an entire application session at the transport layer via a handshake that authenticates the server (via certificate), establishes a shared session key, then switches to fast symmetric encryption for all subsequent data — the basis of HTTPS.
- ✓ IPSec protects traffic at the IP layer using AH (integrity/authentication only) or ESP (integrity plus encryption), in either transport mode (protects only the payload, host-to-host) or tunnel mode (protects the entire original packet, used for VPNs).
- ✓ Firewalls control traffic crossing a network boundary, ranging from fast but context-blind packet filtering, to context-aware stateful inspection, to thorough but slower application-level gateways; a DMZ placed between two firewalls isolates internet-facing servers from the trusted internal network.

— *End of Unit 7* —