NTNU | Norwegian University of Science and Technology

TLS 1.3 AND IP SECURITY

TTM4135 - Lecture 14

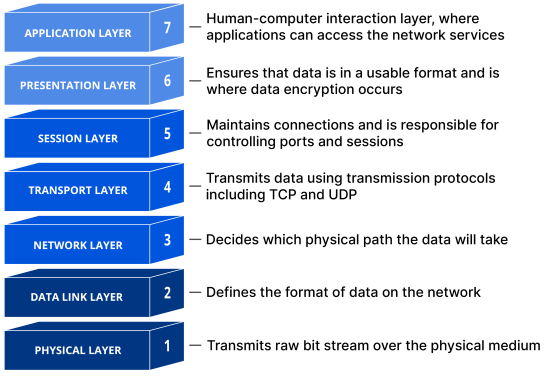
Tjerand Silde

23.02.2026

Motivation

- TLS 1.3 is the latest version of the Transport Layer Security protocol and introduces significant changes that affect both security and efficiency.
- Internet Protocol Security (IPsec) is a framework for ensuring secure communications over IP networks.
- IPsec provides security services similar to TLS, but at a lower layer of the protocol stack. Security can be added to both IPv4 and IPv6.

OSI Model - TLS vs. IPsec



TLS operates at the application layer; IPsec operates at the network layer.

Contents

Why Was TLS 1.3 Needed?

Efficiency: Earlier TLS versions required at least two round-trip times (RTTs) before application data could be sent.

Security: Multiple issues existed in earlier versions:

- Protocol complexity was high.
- Outdated and weak cipher suites were still supported.

- TLS 1.3 was designed according to sound cryptographic principles and aims for *provable security*. Previous versions were more ad-hoc.
- First drafted in 2014 through collaboration between academics, practitioners, and developers.
- Published as Internet Proposed Standard RFC 8446 in January 2018.
- Today supported by around 75% of popular web servers.

Protocol Support

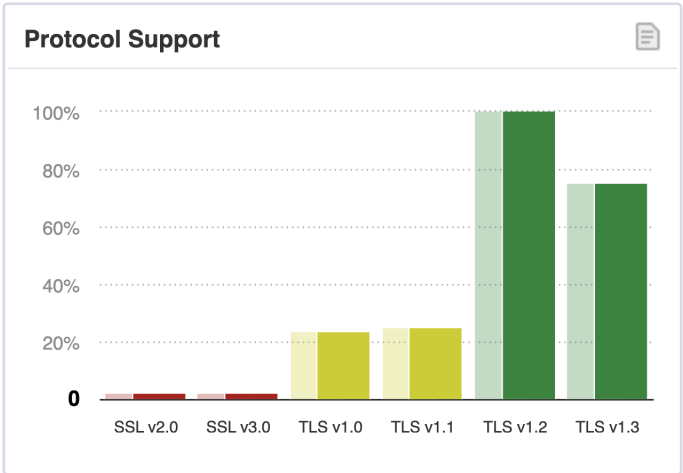


Figure: https://www.ssllabs.com/ssl-pulse

Changes from TLS 1.2 to TLS 1.3

Removed:

- Static RSA and DH key exchange
- Renegotiation
- SSL 3.0 negotiation
- DSA (finite field)
- Data compression
- Non-AEAD cipher suites

Added:

- 0-RTT mode using PSKs
- Post-handshake client authentication (CertificateVerify)
- More AEAD cipher suites

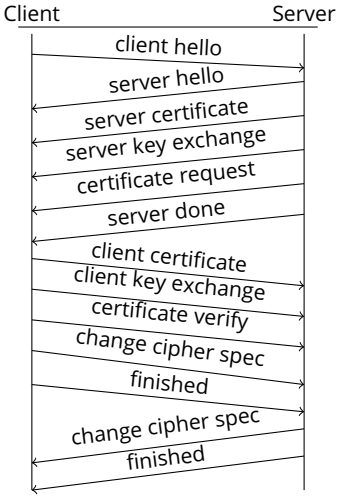
TLS 1.3 Handshake: Hello Messages

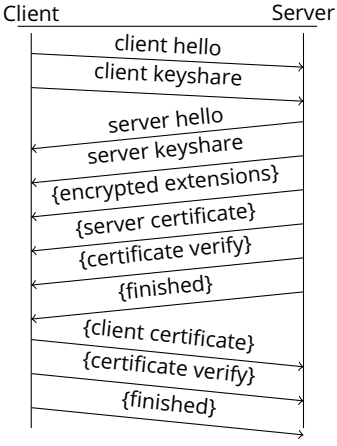
- Client sends *keyshare* in `ClientHello` for one or more anticipated cipher suites since all key exchange algorithms are ephemeral Diffie-Hellman.
- Server can compute a session key on receipt of `ClientHello` if:
  - It accepts one of the client's proposed cipher suites, and
  - The client's keyshare matches the accepted suite.
- Otherwise, the server sends a *HelloRetryRequest* and the client retries with acceptable parameters. This design usually saves one full RTT.

TLS 1.3 Handshake: Other Messages

- Only `ClientHello`/`ServerHello` are not cryptographically protected; all later handshake messages are protected with handshake traffic keys.
- Key derivation uses HMAC-based Extract-and-Expand Key Derivation Function (HKDF) which was standardized in 2010 as RFC5869.
- Key types derived from the master secret:
  - *Handshake traffic keys* to protect the handshake protocol
  - *Application traffic keys* for client-server data
  - *Early data keys* for 0-RTT data
- Compatibility mechanisms are included to interoperate with devices that only accept earlier TLS versions (e.g., some middleboxes).

Handshake: TLS 1.2 (left) vs. TLS 1.3 (right)



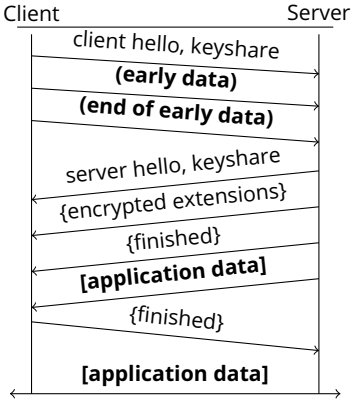


{} protected by handshake traffic keys

Client Authentication

- In both TLS 1.2 and TLS 1.3, client authentication is optional.
- The `CertificateVerify` message includes a signature that can be verified using the public key in the client's certificate.
- TLS 1.3 adds a *post-handshake client authentication* extension, allowing the server to request client authentication at any time after the handshake completes.

0-RTT in TLS 1.3



() protected by early data keys
{} protected by handshake traffic keys
[] protected by application traffic keys

- Possible only with a pre-shared key (PSK).
- PSK can be provisioned externally or via a previous TLS session (NewSessionTicket).
- Early data is optional and does not provide forward secrecy.

TLS 1.3 Cipher Suites

- The handshake always uses (EC)DHE; cipher suites specify only:
  - AEAD cipher for the record layer
  - Hash function for HKDF
- TLS 1.2 (and lower) cipher suite codes cannot be used with TLS 1.3.
- Mandatory ciphersuite to implement: `TLS_AES_128_GCM_SHA256`.
- Recommended ciphersuites:
  - `TLS_AES_256_GCM_SHA384`,
  - `TLS_CHACHA20_POLY1305_SHA256`,
  - `TLS_AES_128_CCM_SHA256`.

ChaCha20/Poly1305

- ChaCha20 is a stream cipher with Poly1305 MAC specified in RFC 8439.
- Available in TLS via RFC 7905. Designed by Daniel J. Bernstein (2008).
- Faster than AES on CPUs without AES hardware acceleration.
- Add-Rotate-Xor construction with 256-bit key and 512-bit keystreams.

TLS 1.3 - Main Improvements

Efficiency:

- Saves one RTT in the handshake
- Follow-on sessions can use 0-RTT

Security:

- Only forward-secret key exchange is allowed
- Many legacy cipher suites removed
- Renegotiation removed
- Formal security analysis and proofs

Quantum-Secure TLS

- There is a current RFC draft about including quantum-safe algorithms in the TLS handshake to prevent the "harvest now, decrypt later" attack: `https://www.ietf.org/archive/id/draft-ietf-uta-pqc-app-00.html`.
- This is already implemented in most modern browsers today.
- Roughly 65 % of HTTPS traffic at Cloudflare is quantum-safe now.
- The most common ciphersuite is the hybrid: `X25519MLKEM768`.
- Quantum-safe authentication is still further down the road.

Contents

IPsec: Introduction

- Standardized in RFCs 4301-4304 (2005), with cryptographic algorithms updated in subsequent RFCs.
- Provides protection for any higher-layer protocol.
- Uses encryption, authentication, and key management algorithms.
- Commonly used to build Virtual Private Networks (VPNs).
- Provides a security architecture for both IPv4 and IPv6.

Security Services

Message confidentiality Prevent unauthorized disclosure via encryption.

Message integrity Detect modifications using a MAC or authenticated encryption.

Limited traffic analysis protection Conceal who communicates, how often, and how much (e.g., by concealing some IP datagram details).

Message replay protection Prevent reuse and unacceptable reordering of packets.

Peer authentication Each IPsec endpoint confirms the identity of the other endpoint.

Gateway-to-Gateway Architecture

- Provides secure communications between two networks.
- Routes site-to-site traffic through the IPsec connection, protecting data in transit.
- Protects data only between the two gateways.
- Common when linking branch offices and headquarters communication.
- Can be less costly than private Wide Area Network (WAN) circuits.

Host-to-Gateway Architecture

- Commonly used to provide secure remote access.
- Organization deploys a VPN gateway on its network.
- Each remote user establishes a VPN connection between the local computer (host) and the gateway.
- The VPN gateway may be a dedicated device or integrated into another network device.
- Typical for connecting hosts on unsecured networks to resources on secured networks.

Host-to-Host Architecture

- Used for specific needs (e.g., administrators managing a single server).
- The only model that provides end-to-end protection for the entire transit.
- Resource-intensive to deploy and maintain (user and host management).
- All participating systems need VPN software installed/configured.
- Key management is often manual / hard to automate.

IPsec Protocol Types

Encapsulating Security Payload (ESP) Provides confidentiality, authentication, integrity, and replay protection.

Authentication Header (AH) Provides authentication, integrity, and replay protection, but not confidentiality; now deprecated.

Internet Key Exchange (IKE) Negotiates, creates, and manages session keys in *security associations*.

Setting Up an IPsec Connection

- Key exchange uses IKEv2 which is specified in RFC 7296 (2014).
- IKEv2 uses Diffie-Hellman authenticated by signatures (public keys in X.509 certificates).
- Cookies mitigate DoS: client must return a time-dependent cookie value (*proof of reachability*) before expensive cryptographic processing.

Security Associations

- A security association (SA) contains information needed by an IPsec endpoint to support a connection.
- May include cryptographic keys and algorithms, key lifetimes, security parameter index (SPI), and a protocol identifier (ESP or AH).
- SPI is included in the IPsec header to associate a packet with the corresponding SA.
- An SA tells the endpoint how to process inbound packets or generate outbound packets. SAs are required in each direction of a connection.
- IKEv2 is used to establish keys to be used in SAs.

Cryptographic Suites

- Similar to TLS cipher suites, IPsec offers standardized suites combining public-key and symmetric algorithms when sending data.
- DH groups are available over finite fields and on elliptic curves in IPsec.
- IPsec uses 3DES or AES for data encryption (e.g., CBC or GCM modes).
- HMAC for integrity if authenticated encryption (e.g., GCM) is not used.

IPsec Modes of Operation

- Each protocol (ESP or AH) can operate in **transport** or **tunnel** mode.
- **Transport mode:** Keep the original IP header; protect the payload - commonly used for host-to-host.
- **Tunnel mode:** Encapsulate the original packet as payload of a new outer packet - commonly used for gateway-to-gateway.
- The next figures are shown for IPv4 (there are slight differences for IPv6).

IPsec Protocol Components (ESP)

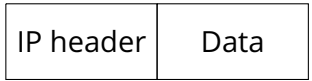
ESP header Contains SPI and sequence number.

ESP trailer Contains padding and padding length; extra padding may be included to hinder traffic analysis.

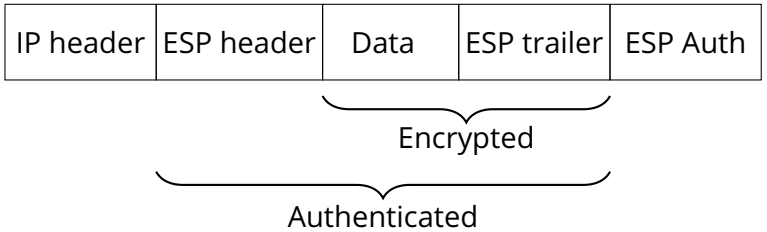
ESP Auth Contains a MAC of the encrypted data and ESP header (may be omitted if an authenticated encryption mode is used).

Transport Mode ESP

- **Original IP packet**



- **IP packet protected by Transport-ESP**

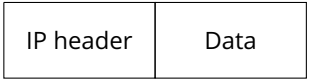


ESP in Transport Mode: Outbound Processing

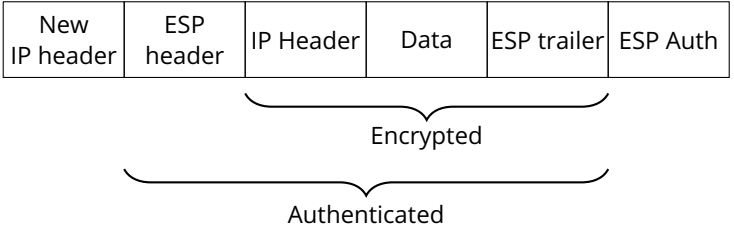
- Data after the original IP header is padded by adding an ESP trailer and then encrypted using the symmetric cipher and key in the SA.
- An ESP header is prepended. If the SA uses authentication, an ESP MAC is calculated over the data so far and then appended.
- The original IP header is prepended, but some IP header fields must be updated as following:
  - The protocol field changes to ESP.
  - The total length is updated to reflect added ESP data.
  - Checksums are recalculated as needed.

Tunnel Mode ESP

- **Original IP packet**



- **IP packet protected by Tunnel-ESP**



ESP in Tunnel Mode: Outbound Processing

- The entire original packet is padded (ESP trailer) and then encrypted using the symmetric cipher and key in the SA. An ESP header is prepended.
- If authentication is used, an ESP MAC is calculated and appended.
- A new outer IP header is prepended:
  - The inner IP header carries the ultimate source/destination.
  - The outer IP header may carry gateway addresses.
  - The outer IP header protocol field is set to ESP.

IPsec Security Considerations

- Active attacks have been demonstrated for *encryption-only* ESP; encryption without integrity is insecure.
- The 2005 IPsec revisions do not require support for encryption-only mode (though it may still be allowed).
- In common ESP usage, encryption is applied before the MAC.
- Using AH, a MAC can be applied before encryption (as in TLS); attacks have been shown against certain configurations.
- Formal analyses indicate no significant weaknesses in the IKEv2 key exchange protocol design.

Questions?