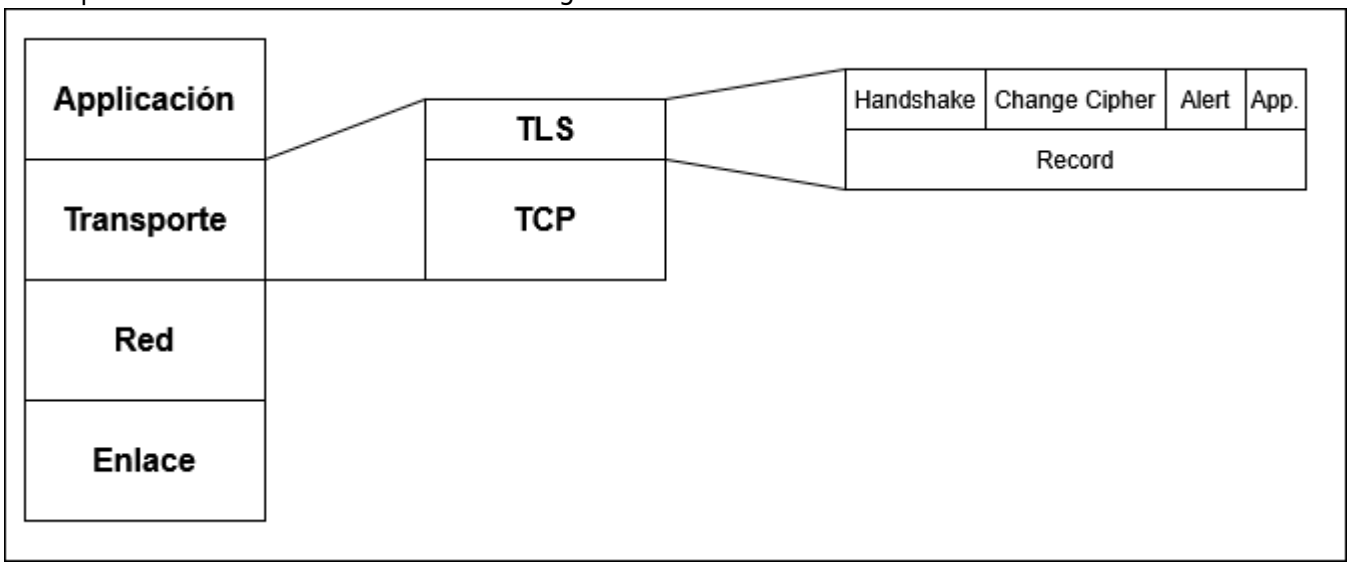
Seguridad a nivel de transporte [L4]

TLS (Transport Layer Security)

Es una evolución de SSL (Secure Socket Layer) para proveer comunicaciones seguras a través de infraestuctura insegura. Provee un canal seguro a un servicio arbitrario de internet. Garantiza autenticación, confidencialidad e integridad, tiene los siguientes objetivos:

- Seguridad Criptográfica
- Interoperabilidad
- Extensibildiad
- Eficiencia

La arquitectura de TLS en internet es la siguiente:

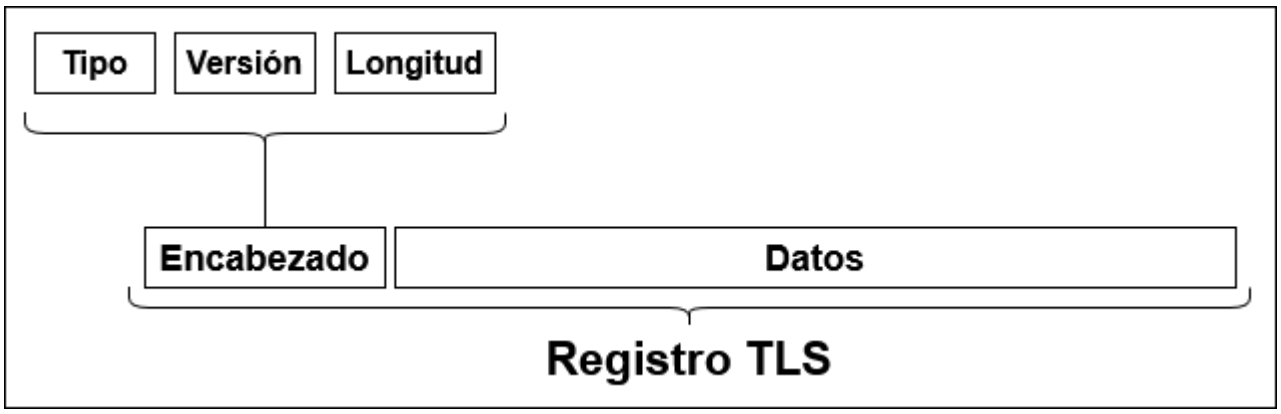


Competidores de TLS

- SSH (Secure Shell)
  - También en capa de aplicación
  - Usa cifrado de clave pública para la autenticación pero también puede usar contraseñas
  - Confiada basada en hosts conocidos e intercambio de claves en vez de PKI
  - Se suele usar para acceso remoto a servidores, transferencia de archivos o tunelado de otros protocolos.
- PGP(Pretty Good Privacy):
  - Confianza de web descentralizada en vez de usar PKI jerárquico.
  - Se suele usar para email, archivos y verificación de paquetes de software.

Protocolo TSL (1.2 y 1.3)

Transporta y, opcionalmente, cifra cada mensaje TLS entre 2 aplicaciones. Un registro TLS tiene la siguiente estructura:



- Trasnsporte de mensaje: Se transportan buffers opacos enviados por subcapas del protocolo superarioes. Puede fragmentar mensajes mayores de 16384 bytes y combinar varios mensajes pequeños en un solo regisotr.
- Cifrado y validación de integridad: Los primeros mensajes se transmiten en claro, una vez finaliza el handshake, se cifra y valida de acuerdo a los parámetros negociados
- Compresión: Ya no se usa, sujetoa a ataques de compresión de canal lateral.
- Extensibilidad: El protocolo de registro solo trata con el transporte y el cifrado, las demás tareas son llevadas a cabo por un subprotocolo. Hay 4 subprotocolos principales:
  - Handshake
  - Change cipher spec
  - Datos de aplicación
  - Alert.

Protocolo de Handshake

Responsable de la negociación de los parámetros de conexión y realizar la autenticación. Intercambia entre 6 y 10 mensajes, dependiendo de las características. Suelen haber 3 flujos comunes:

- Handshake completo con autenticación de servidor
- Handshake abreviado continuando una sesión anterior
- Handshake completo con autenticación mutua.

especificación_del_mensaje

```
struct {
  HandshakeType msg_type; //1 Byte
  uint24 length;
  HandshakeMessage message; //Depende del tipo de mensaje
} Handshake;
```