GitHub repository

A tantárgyhoz kapcsolódó anyagok és példaprogramok elérhetők az alábbi GitHub linken:

https://github.com/zbalogh/oe-internetes-alkalmazasok

Biztonságos kommunikáció az interneten

HTTPS, TLS/SSL és PKI alapjai

Miért kell HTTPS?

- Adatvédelem (Confidentiality)
- Adatintegritás (Integrity)
- Hitelesítés (Authentication)
- MITM támadások elleni védelem

HTTP vs HTTPS

- **HTTP**: Port 80, titkosítatlan
- **HTTPS**: Port 443, titkosított (**TLS** protokollra épül)
- Rétegek: TCP -> TLS -> HTTP
- Böngészők: kötelező HTTPS

Mi az a TLS/SSL?

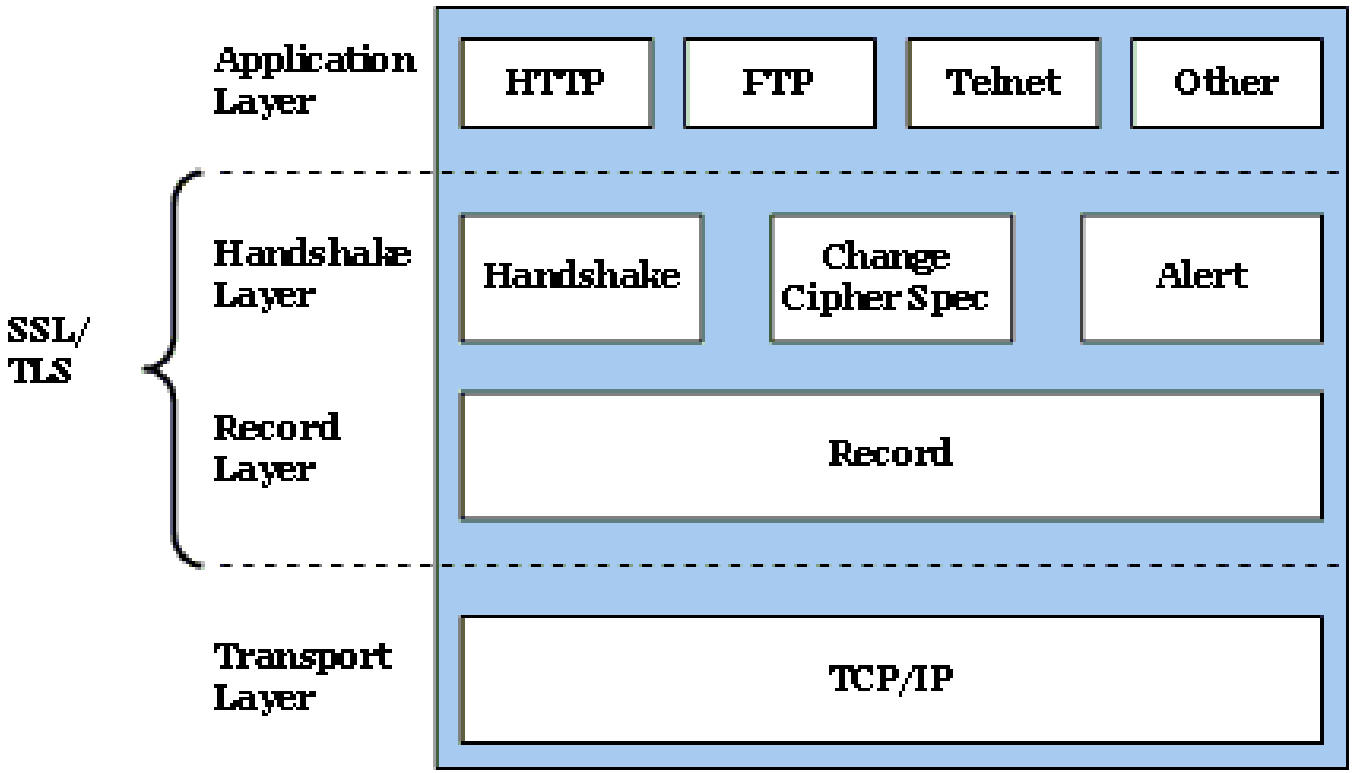
- **SSL** (Secure Socket Layer): régi protokoll, ma már elavult!
- **TLS** (Transport Layer Security) : modern verzió (1.2, 1.3)
- A **HTTPS** valójában: HTTP + TLS
- A TLS célja egy **biztonságos csatorna** létrehozása két végpont között

A TLS szerepkörei

- **Titkosítás**: Adatok bizalmas kezelése (AES)
- **Hitelesítés**: A szerver személyazonosságának igazolása tanúsítvánnyal
- **Integritás**: Az adatok sértetlenségének és hitelességének ellenőrzése (HMAC vagy AEAD)
- **Kulcscsere**: Közös titkos kulcs biztonságos létrehozása (RSA, Diffie–Hellman, ECDHE)

A TLS rétegei

- **Record Layer**: adat titkosítás és integritás
- **Handshake**: kapcsolatfelépítés és kulcscsere
- **Alert**: hibakezelés és megszakítás
- **ChangeCipherSpec**: titkosított kommunikációra váltás


Application Layer
HTTP
FTP
Telnet
Other
SSL/ TLS
Handshake Layer
Handshake
Change Cipher Spec
Alert
Record Layer
Record
Transport Layer
TCP/IP

PKI és a tanúsítványok

- Mi az a PKI?
- CA, Intermediate CA, Root CA
- CA által aláírt X.509 tanúsítvány
- Publikus és privát kulcs
- Tanúsítványlánc

Tanúsítvány érvényesség és visszavonás

- A tanúsítványok érvényességi idővel rendelkeznek
- A tanúsítvány visszavonható, például ha a privát kulcs kompromittálódott
- **CRL** (Certificate Revocation List)
- **OCSP** (Online Certificate Status Protocol)
- **OCSP** stapling

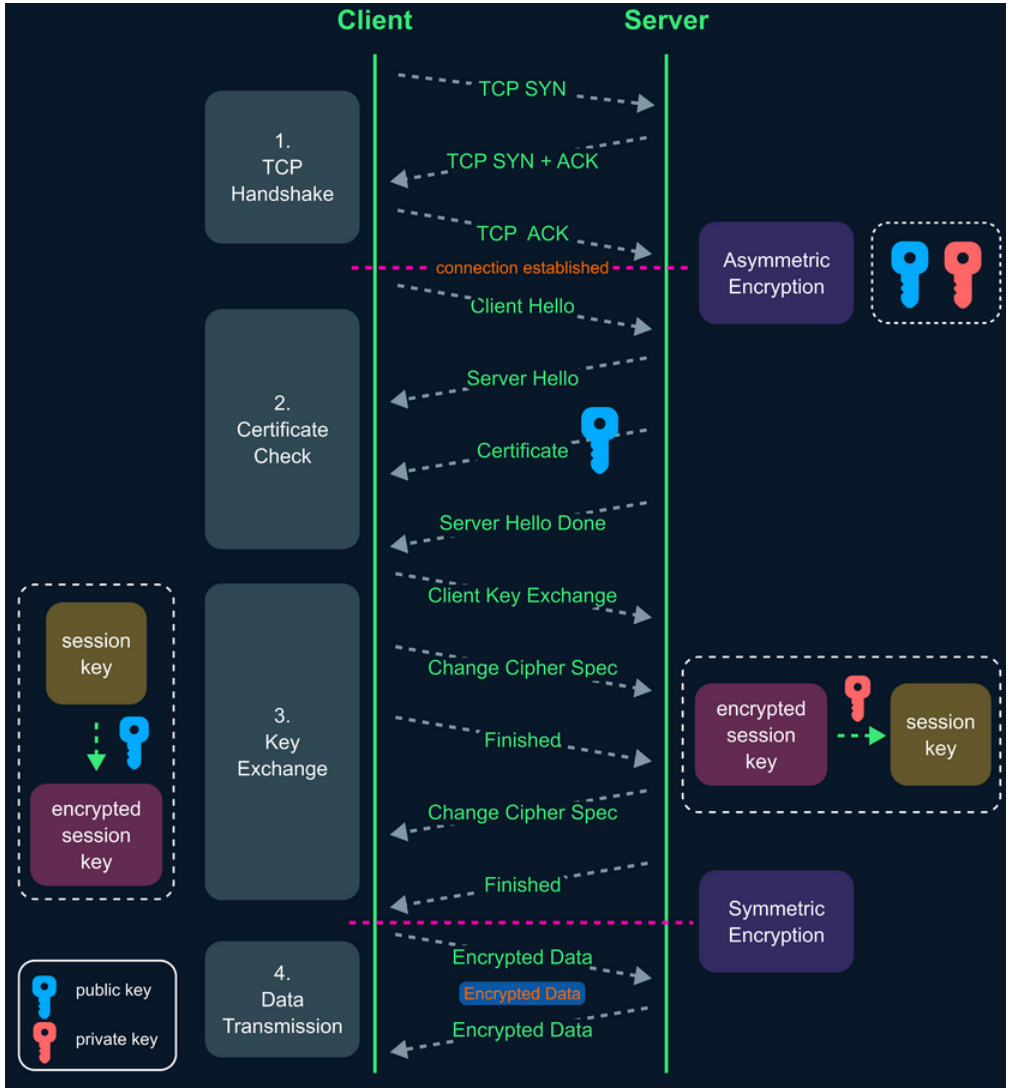
TLS Handshake (általános áttekintés)

A handshake céja, a **biztonságos csatorna felépítése**

- **Kliens** → Szerver: képességek, algoritmusok
- **Szerver** → Kliens: tanúsítvány, választott algoritmusok
- **Kulcscsere** (aszimmetrikus titkosítás)
- **Session key** (titkos kulcs) létrehozása
- **Titkosított kommunikáció** indul (szimmetrikus titkosítás)

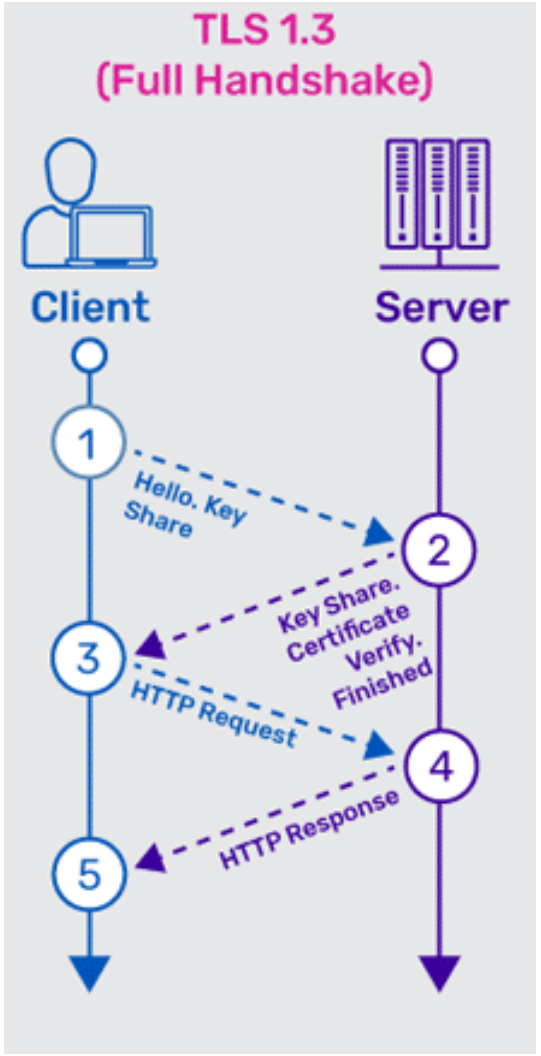
(HTTPS) TLS 1.2 Handshake

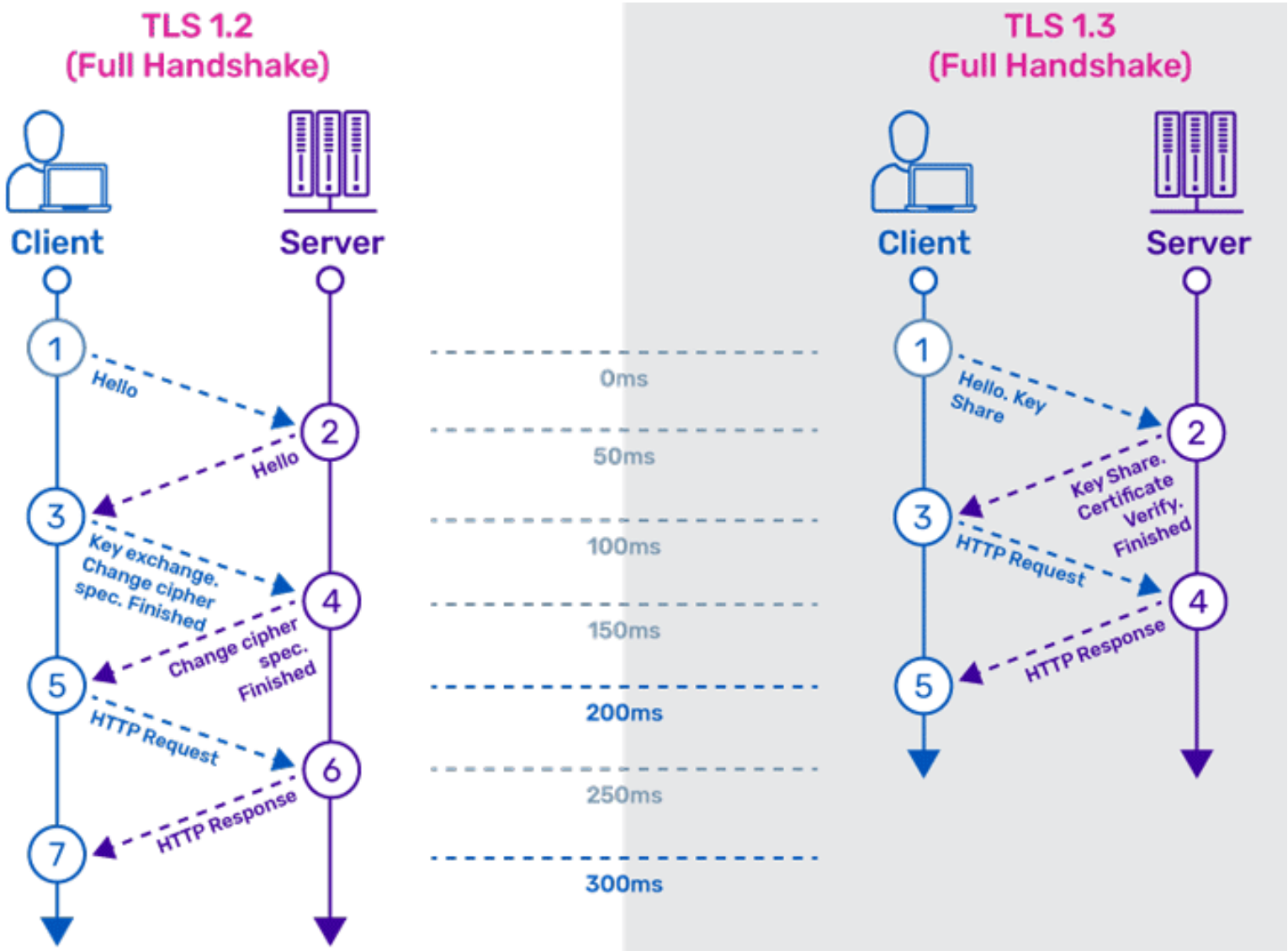
- ClientHello üzenet
- ServerHello üzenet
- Server Certificate küldése
- ServerKeyExchange üzenet (opcionális)
- ClientKeyExchange üzenet
- Session key generálása
- ChangeCipherSpec üzenetek
- Finished üzenetek



(HTTPS) TLS 1.3 újdonságok

- Egyszerűsített handshake
- Gyorsabb kapcsolatfelépítés
- Modern algoritmusok használata
- Minden kulcscsere ECDHE algoritmussal
- PFS (Perfect Forward Secrecy) kötelező




TLS 1.2
(Full Handshake)
Client
Server
1
Hello
2
Hello
3
Key exchange.
Change cipher
spec. Finished
4
Change cipher
spec.
Finished
5
HTTP Request
6
HTTP Response
7
0ms
50ms
100ms
150ms
200ms
250ms
300ms
TLS 1.3
(Full Handshake)
Client
Server
1
Hello. Key
Share
2
Key Share.
Certificate
Verify.
Finished
3
HTTP Request
4
HTTP Response
5

MITM támadás és védekezés

- MITM = Man In The Middle
- Tanúsítvány hamisítás – böngésző hibát jelez
- DNS eltérítés
- HSTS (HTTP Strict Transport Security)

Modern legjobb gyakorlatok

- TLS 1.3 előnyben
- Régi algoritmusok tiltása
- Erős algoritmusok (cipher suites) használata
- HTTPS automatikus kényszerítése
- Let’s Encrypt használata (ingyenes)

Összefoglalás

- HTTPS = titkosítás + hitelesség + integritás
- Tanúsítványok és PKI biztosítják a bizalom alapját
- TLS handshake a kapcsolat lelke
- MITM támadások jól kivédhetők