# SSL/TLS, RSA - Symetrické a asymetrické šifrování, SSL/TLS handshake. Certifikační autority. Princip ověření pravosti klíče

## SSL/TLS

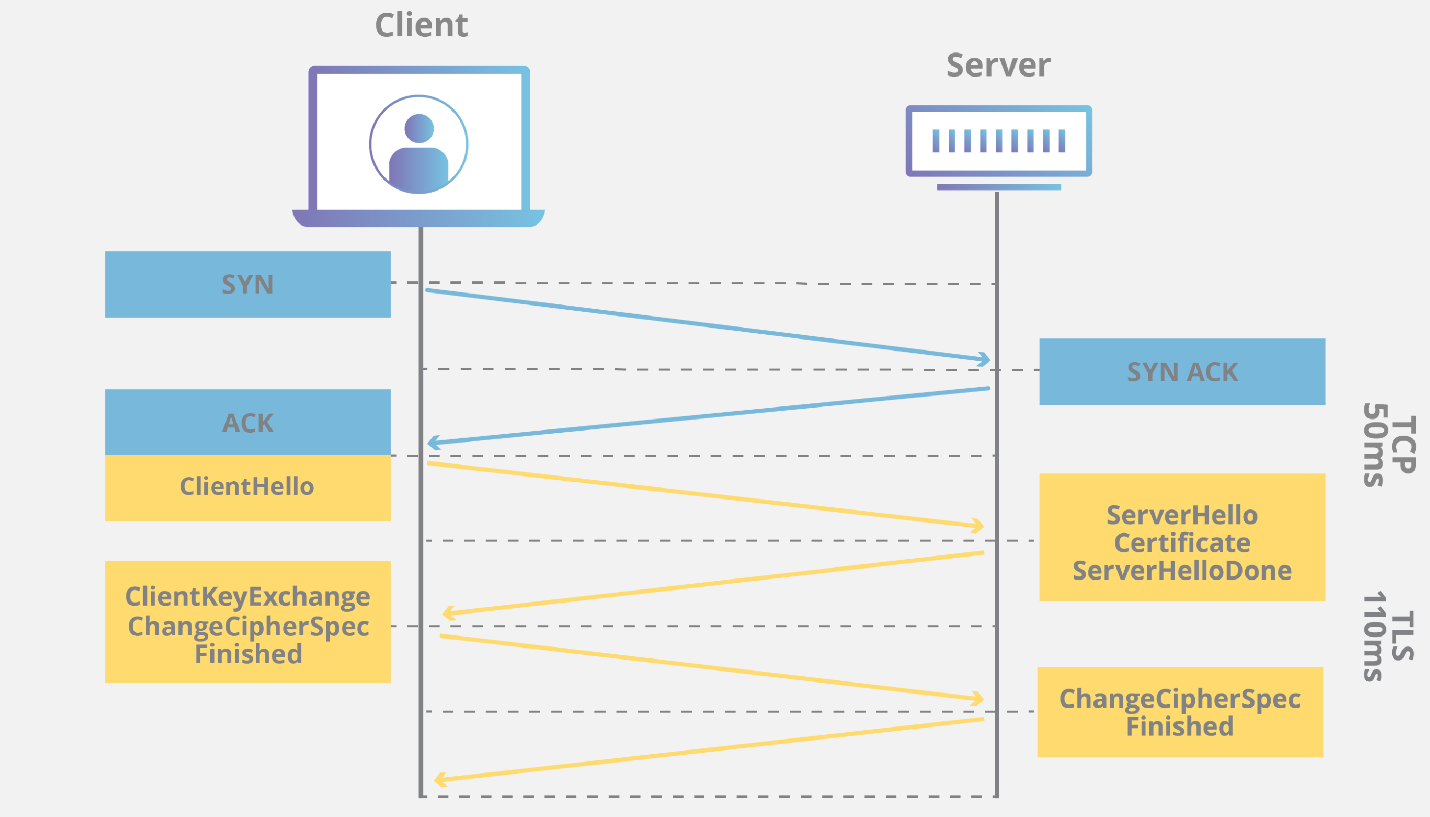
TLS (Transport Layer Security) jedná se o kryptografický protokol SSL (Secure Sockets Layer) je jeho předchůdcem a mají za úkol poskytnout bezpečnou komunikace přes internet. Tento protokol e velmi používaný v službách jako jsou email, VOIP (voice over ip), HTTPS. Samotné SSL se již moc nepoužívá možná jen v nějakých zastaralých aplikací, ale moderní svět postupně přešel na TLS.

TLS se zaměřuje na poskytování security, including privacy (confidentiality), integrity, a ověřování použitím cryptography, jako jsou například certifikáty, mezi dvěma nebo vice aplikacemi.

### TLS vs SSL

| **SSL** | **TLS** |
| --- | --- |
| SSL stands for[Secure Socket Layer](https://www.geeksforgeeks.org/secure-socket-layer-ssl/). | TLS stands for [Transport Layer Security](https://www.geeksforgeeks.org/transport-layer-security-tls/). |
| SSL (Secure Socket Layer) supports the **Fortezza** algorithm. | TLS (Transport Layer Security) does not support the **Fortezza** algorithm. |
| SSL (Secure Socket Layer) is the 3.0 version. | TLS (Transport Layer Security) is the 1.0 version. |
| In SSL( Secure Socket Layer), the Message digest is used to create a master secret. | In TLS(Transport Layer Security), a Pseudo-random function is used to create a master secret. |
| In SSL( Secure Socket Layer), the Message Authentication Code protocol is used. | In TLS(Transport Layer Security), Hashed Message Authentication Code protocol is used. |
| SSL (Secure Socket Layer) is more complex than TLS(Transport Layer Security). | TLS (Transport Layer Security) is simple. |
| SSL (Secure Socket Layer) is less secured as compared to TLS(Transport Layer Security). | TLS (Transport Layer Security) provides high security. |
| SSL is less reliable and slower. | TLS is highly reliable and upgraded. It provides less latency. |
| SSL has been depreciated. | TLS is still widely used. |
| SSL uses port to set up explicit connection. | TLS uses protocol to set up implicit connection. |

## TLS Handshake



### Proč probíhá TLS handshake

* Určení jaká verze TLS se bude používat (TLS 1.0, 1.2, 1.3, etc.)
* Určují jaký cipher suite se bude používat
  + Cipher suite je skupina algoritmů která vytváří šifrované komunikační spojní
  + key exchange algorithm, a bulk encryption algorithm, and a message authentication code (MAC) algorithm.
* Autentifikace identity za pomocí server public key a SSL certifikátu schváleného autoritou
* Generují session key za účelem symetrického šifrování poté co handshake proběhne

Průběh handshake bude záviset na tom jaký key exchange algoritmus se bude používat a jaký cipher suite podporují obě dvě strany. Všechny TLS handsahky využívají asymetrické šifrvání pro komunikaci.

#### Postup komunikace:

1. **The 'client hello' message:** Klient započíná handshake tím že posílá “hello” message server. Message obsahuje jakou TLS verzi klient podporuje, jakou cipher suite klient podporuje a řetězec náhodných bajtů … "client random."
2. **The 'server hello' message:** Jako odpověď na klientovo “hello” server posílá message obsahující jeho SSL certifikát, server podporovaný cipher suite, a "server random", znovu náhodně vygenerovaný tentokrát serverem.
3. **Authentication**: Klient ověří SSL certifikát za pomocí certifikační autority která ho používá. Tím potvrdí svojí totožnost a trustworthy server.
4. **The premaster secret:** Klient posílá jeden další náhodný řetězec bajtů … "premaster secret". “Premaster secret” je šifrovaný pomocí public key a může být dešifrovaný za pomocí private key který má server. (Public klíč dostane klient z SSL certifikátu)
5. **Private key used:** Server dešifruje “Premaster secret”.
6. **Session keys created:** Oba klient i server vygenerují session key z "client random", "server random" a “Premaster secret”, Měli by dojít k stejnému výsledku.
7. **Client is ready:** Klient pošle "finished" message která je zašifrovaná pomocí session key.
8. **Server is ready:** Server pošle “finished” message zašifrovanou pomocí session key.
9. **Secure symmetric encryption achieved:** Handshake je u konce, comunikace může pokračovat kde se bude využívat session key pro simetrické šifrování.

## Symetrické a Asymetrické šifrování

Principem šifrování je převod čitelných dat na nečitelná data. Zpětný převod z nečitelných dat na čitelná data se nazývá dešifrování. Šifrovat mohu jak data, která si chci ponechat na svém zařízení (aby si je nemohl přečíst případný útočník, který se jich zmocní), tak především data, která chci zaslat někomu dalšímu (šance na zachycení dat přenášených přes Internet je poměrně vysoká – šifrováním zajistím, že zachycená data nebude možné zneužít).

### Symetrické

Různé techniky šifrování stojí na odlišných matematických základech, ale většina z nich pracuje se šifrovacími klíči. V případě symetrického šifrování se tentýž klíč používá jak pro šifrování, tak i pro dešifrování. Postup symetrického šifrování je poměrně jednoduchý. Odesílatel a příjemce se nejprve domluví na klíči, tedy na sekvenci znaků (nejlépe bez reálného významu). Odesílatel zprávu zašifruje, zašle ji příjemci a ten si ji dešifruje.

### Asymetrické

Slabinu symetrického šifrování – tedy nutnost sdílení šifrovacího klíče – řeší asymetrické šifrování. Při něm se používají dva klíče – jeden pro šifrování a druhý pro dešifrování. Chce-li odesílatel poslat příjemci zašifrovanou zprávu, tak nejprve odesílatel požádá příjemce o jeho šifrovací klíč. Tímto klíčem zašifruje zprávu a odešle ji. Příjemce si pak zprávu svým dešifrovacím klíčem dešifruje. Znalost pouhého šifrovacího klíče neumožňuje dešifrování zpráv, a tak je možné tento klíč bez rizika poslat i přes běžné nezabezpečené připojení. Tomuto klíči se proto říká veřejný klíč. Dešifrovací klíč je naproti tomu potřeba uchovat v tajnosti – říká se mu proto soukromý klíč.

#### RSA (iniciály Rivest–Shamir–Adleman)

Jeden z nejstarších public-key kryptografických systémů, který se používá pro bezpečný přenos dat. V tomto algoritmu je šifrovací klýč public a liší se od dešifrovacího. Používá se i dnes, přičemž při dostatečné délce klíče je považován za bezpečný.

Je založen na principu že je složité rozložit velké číslo na součin prvočísel. Naproti tomu násobení dvou velkých čísel je elementární úloha.

Obsah obrázku text, Písmo, snímek obrazovky

Popis byl vytvořen automaticky

## Certifikační autorita

Certifikační autorita (zkratka CA) je v asymetrické kryptografii subjekt, který vydává digitální certifikáty (elektronicky podepsané veřejné šifrovací klíče), čímž usnadňuje využívání PKI (Public Key Infrastructure) tak, že svojí autoritou potvrzuje pravdivost údajů, které jsou ve volně dostupném veřejném klíči uvedeny. Na základě principu přenosu důvěry (viz níže) tak můžeme důvěřovat údajům uvedeným v digitálním certifikátu za předpokladu, že důvěřujeme samotné certifikační autoritě.

Certifikát autorita podepisuje za pomocí privátního klíče. Následně pak když si chci ověřit že je podpis důvěryhodný porovnám ho s publik key který už v dnešní době mohou být přímo v prohlížeči.

Vše zpočívá na důvěře v Certifikační autoritu když chceme zjistit validitu SSL certifikátu.

1. Stahování certifikačních řetězců: Spolu s certifikátem obdržíte také certifikační řetězce. Certifikační řetězec obsahuje certifikát vydávající autority (tzv. kořenové certifikáty), které jsou považovány za důvěryhodné. Prohlížeč potřebuje tyto kořenové certifikáty, aby mohl ověřit důvěryhodnost certifikátu, který jste obdrželi.
2. Verifikace digitálního podpisu: Prohlížeč zkontroluje platnost digitálního podpisu na certifikátu. Proces zahrnuje ověření, zda digitální podpis byl vygenerován pomocí privátního klíče vydávající autority, a zda je certifikát platný.
3. Kontrola důvěryhodnosti vydávající autority: Prohlížeč zkontroluje, zda je vydávající autorita, která podepsala certifikát, uvedena v certifikačním řetězci a zda je považována za důvěryhodnou. Pokud je vydávající autorita důvěryhodná, prohlížeč přijme certifikát jako důvěryhodný.

Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky