**Základy zabezpečení komunikace**

Základem zabezpečení komunikace po počítačové síti je tzv. kryptografie neboli šifrování. Podstata šifrování spočívá v tom, že zašifrovanou informaci nelze odhalit, případně je to možné pouze s použitím velkého (nereálného) množství výpočetního výkonu za hodně dlouhou dobu. Šifrovat můžeme informace:

* **obousměrně** – data zašifrujeme tajným klíčem tak, aby je mohl jejich příjemce podobným způsobem znovu dešifrovat
* **jednosměrně** – mluvíme o tzv. **hashování**, kdy se z původní informace ze zašifrované podoby zpětně získat nedá, jedná se pouze o tzv. otisk původní informace

**Obousměrné šifry pak dělíme na:**

* **symetrické** – data se šifrují i dešifrují pomocí **tajného klíče**, který je předem znám oběma komunikujícím stranám. Tajný klíč je potřeba chránit, aby se k němu nedostaly jiné osoby, které by pak mohly rozkrýt obsah šifrované komunikace. Typickým příkladem symetrického šifrování je šifra **AES** (American Encryption Standard).
* **asymetrické** – obě strany disponují dvojicí klíčů:
* **veřejný klíč (public key)** – je to klíč, který je předáván komunikující protistraně, jeho znalost neohrozí bezpečnost šifrovaných informací, slouží k zašifrování dat, ta se následně v šifrované podobě odešlou příjemci, který tento veřejný klič vygeneroval
* **privátní klíč (private key)** – nikdy neopouští počítač svého majitele, který jej vygeneroval, jeho prozrazení by ohrozilo bezpečnost přenášené informace. Pomocí privátního klíče dešifruje příjemce obsah zašifrované zprávy.

U asymetrické šifry nejprve obě strany vygenerují své veřejné a privátní klíče, následně si navzájem své veřejné klíče vymění. Pomocí vyměněných klíčů šifrují posílanou komunikace a tu opět svými privátními klíči dešifrují. Typickým příkladem je šifra **RSA**.

**Hashování**

Jednosměrná operace, kdy se zpracovaná informace zjednoduší a pomocí matematických algoritmů převede do podoby, z níž není možné ji získat zpět. Každá informace vytváří svůj jedinečný otisk tzv. hash. Používá se např. k ukládání přihlašovacích hesel, kdy na serveru nebo PC jsou uloženy pouze otisky hesel, nikoli hesla samotná a při přihlašování se zadané heslo též zahashuje a následně se porovnávají pouze hash hodnoty. Hojně používanými hashovacími algoritmy jsou MD5 (dnes již není považován za bezpečný) a SHA.

**Zabezpečení komunikace na webu a mailové komunikace**

**Certifikáty**

Probíhá prostřednictvím zabezpečených protokolů **SSL/TLS**, která chrání přenášená data proti vyzrazení jejich obsahu a pomocí **elektronického podpisu**, který jednoznačně identifikuje odesílatele a původ přenášené informace. Pro tyto účely se využívá tzv. certifikátů vydávaných tzv. certifikačními autoritami, tvoří hierarchickou strukturu:

* **kořenový (root) certifikát** – jde o certifikát kořenové certifikační autority, z hlediska hierarchie je nejvyšší možný, ověřují se pomocí něj certifikáty nižší hierarchické úrovně.
* **certifikáty nižší úrovně** – je navázaný a podřízený kořenové certifikační autoritě, může se jednat o certifikát zprostředkující certifikační autority nebo o certifikát uživatele

Certifikát nižší úrovně je vždy podepsán certifikátem vyšší úrovně, tím je zaručena jejich návaznost a následné ověřování. Princip návaznosti a správy hierarchie certifikátů se nazývá jako **PKI (Private Key Infrastructure)**.

**Certifikáty obsahují následující údaje:**

* jméno držitele
* jméno vystavitele
* začátek platnosti
* konec platnosti
* podpis vystavitele
* veřejný klíč držitele

Na certifikát je navázán i soukromý klíč držitele certifikátu, který je uložen na jeho PC v úložišti certifikátů, na bezpečnostní kartě nebo tzv. tokenu. Certifikáty se používají pro asymetrickou kryptografii.

**Vytvoření SSL/TLS komunikace (handshake)**

Obousměrná komunikace klienta se serverem:

1. Klient posílá na server zprávu **Client hello**, obsahuje podporované protokoly a šifry, podporovanou kompresi a náhodně generovaný řetězec.
2. Server posílá klientovi zprávu **Server hello** a náhodně generovaný řetězec a veřejnou část certifikátu.
3. Server posílá klientovi vybraný protokol a metodu šifrování a komprese, session ID, volitelně posílá požadavek na certifikát klienta, zpráva je ukončena jako **Server hello done**.
4. Klient použije veřejný klíč ze serverového certifikátu k zašifrování a odeslání **Pre-Master key**, který bude sloužit k vygenerování **Master key** klíče pro šifrování dat. Zprávou **Client key Exchange** odešle **Pre-Master key** na server.
5. Server obdrží **Pre-Master key** a spolu s klientem oba vygenerují za pomocí obou náhodně vygenerovaných řetězců a Pre\_Master key klíč **Master key** a klíče spojení (session key).
6. Klient posílá serveru zprávu **Change cipher specs**, kde informuje server, že začíná používat šifrované spojení, dále posílá zprávu **Client finished**.
7. Server přijímá zprávu **Changed cipher specs**, změní svůj stav na dohodnutou metodu šifrování s použitím vygenerovaných klíčů. Šifrování probíhá od této chvíle symetricky. Zároveň posílá zprávu **Server finished**.
8. Od této chvíle probíhá obousměrná komunikace mezi klientem a serverem šifrovaně.

**S/MIME, elektronický podpis**

Používají se pro zabezpečení mailové komunikace a ověření odesílatele. Pro tyto účely slouží hash a asymetrická kryptografie. V případě elektronického podpisu je postup otočen, k šifrování hashe se používá soukromý klíč odesílatele a k ověření jeho veřejný klíč, který je zveřejněn certifikační autoritou.

**Princip EP je rozdělen do dvou fází**

**Při odeslání zprávy** je ze zprávy vygenerován hash, ten je následně zašifrován pomocí soukromého klíče odesílatele zprávy. Tím vznikne vlastní elektronický podpis, který se spolu s původní zprávou a veřejnou částí certifikátu odesílatele pošle příjemci zprávy. Tím je zajištěno ověření identity (certifikát je možno ověřit pomocí certifikační autority) a nezměnitelnost odesílané zprávy (obsahuje hash).

**Při přijetí zprávy** příjemce pomocí veřejného klíče certifikátu odesílatele dešifruje data obsažená v elektronickém podpisu a získá z něj původní hash. Dále z přijaté zprávy sám znovu vygeneruje svůj hash a obě hash hodnoty porovná. Pokud se hodnoty hash shodují, je ověřena neporušenost zprávy.