Otázka 22
Zadání

Základy kryptografie
Teorie složitosti, teorie informace a pravděpodobnost
Základy kryptoanalýzy šifer
Proudové šifry, blokové šifry a symetrické šifrování
Hašovací funkce
Základní principy asymetrického šifrování a výměny klíčů
Digitální podpisy
Kryptografie eliptických křivek

***Předmět:* A7B32KBE**

Základy kryptografie

Kryptografie neboli šifrování je nauka o metodách utajování smyslu zpráv převodem do podoby, která je čitelná jen se speciální znalostí (tou jsou dnes prakticky myšleny dvě věci: znalost použitého šifrovacího algoritmu a použitého klíče). Slovo kryptografie pochází z řečtiny – kryptós je skrytý a gráphein znamená psát - tedy "nauka o skrytém písmu". Oproti tomu *kryptologie* zahrnuje kryptografii a kryptoanalýzu, neboli luštění zašifrovaných zpráv.

Kryptografie se po staletí vyvíjela k větší složitosti zároveň s lidskou civilizací. První doložení o zašifrování zprávy pochází z roku 480 pr. n. l. za období Řecko-Perských válek v bitvě u Salamíny. Do historie kryptografie se zapsal i významný římský vojevůdce a politik Julius Caesar, a to vynalezením šifry, která byla pojmenována jako Caesarova šifra. To bylo období klasické kryptografie, kdy k šifrování stačila pouze tužka a papír, případně jiné jednoduché pomůcky.
Během 1. poloviny 20. století s nástupem složitější mechaniky a později elektroniky a počítačů začaly vznikat různé sofistikované přístroje, které umožnovaly složitější postupy při šifrování. Tím přibližně začala druhá část, kterou nazýváme moderní kryptografie. Zpočátku byly užívány speciální mechanické a elektromechanické šifrovací stroje (např. Enigma); v dnešní době se již zpravidla nepoužívájí zvlášt vytvářené přístroje, ale klasické počítace se softwarovou aplikací nějakého šifrovacího algoritmu.

Na bezpečném přenosu informací a na schopnostech protivníka šifru rozbít dnes závisí spousta moderních aplikací - utajení uživatelských hesel, bezpečný přenos emailů, souborů či komunikace s bankou pomocí elektronického bankovnictví.

Slovem *šifra* nebo *šifrování* označujeme kryptografický algoritmus, který převádí čitelnou zprávu neboli prostý text na její nečitelnou podobu neboli šifrovaný text. Klíč je tajná informace, bez níž nelze šifrovaný text prečíst.

Otevřeným textem se v kryptografii rozumí text nezašifrovaný, čitelný. Jedná se tedy o vstup algoritmu šifrovacího a výstup algoritmu dešifrovacího. V původním historickém pojetí se jednalo skutečně o text v jazyce komunikujících, dnes pojem zobecněl a často se tím míní všeobecně jakákoliv nešifrovaná data, tedy například obrázky, videa, ale i třeba i data posílaná mezi bankomatem a bankou.

Z hlediska bezpečnosti šifrování je otevřený text tím, co se nesmí nikdy dostat do rukou útočníka, tedy musí být bezpečně uschováno mimo jeho dosah. Od šifer samotných se pak navíc obvykle požaduje, aby byly odolné vůči útoku se známým otevřeným textem a útoku s výběrem otevřeného textu. Tedy znalost nějaké zprávy v obou jejích podobách - otevřené i zašifrované by neměla nikomu dát do ruky

pomůcku pro rozluštění / prolomení šifry.

Kód versus šifra: Velice často dochází k záměně slov kód (*kódování*) a šifra (*šifrování*). Nejedná se však o totéž. Kódování je převod informace z jedné podoby (reprezentace) do jiné, avšak ne za účelem jejího utajení! Přestože neznalost použitého kódu může stížit rozlišení a přečtení zakódované informace, není to primární účel. Zde je třeba zmínit, že v nějakém kódu se nachází ve své podstatě vždy všechny informace. Kódem je například i písmenová abeceda, zápis čísel arabskými číslicemi či český jazyk. Při kódování potom dochází k převodu z jednoho kódu na jiný za účelem přizpůsobení dat pro jejich následné další zpracování, například přenos určitým přenosovým médiem. Například při přenosu dat digitálním přenosovou cestou jsou na daném vedení vysílány pouze dvě úrovně signálu - proto je třeba přizpůsobit i přenášená data jejich převodem na binární reprezentaci (převod do dvojkového / binárního kódu). Datový proud zprávy se pak skládá pouze z řady jedniček a nul, což odpovídá právě potřebným dvou úrovním signálu.

Steganografie: Starší sestrou kryptografie je steganografie neboli ukrývání zprávy jako takové. Sem patří různé neviditelné inkousty, vyrývání zprávy do dřevěné tabulky, která se zalije voskem apod. V moderní době lze tajné texty ukrývat například do souboru s hudbou či obrázky namísto náhodného šumu. Lze provést i skrytí skutečného zašifrovaného textu mezi nic neznamenajícími náhodnými daty. Ta můžou být vytvořena algoritmy generujícími pseudonáhodná čísla nebo hashovacími funkcemi (o hašovacích funkcích se dočtete dále). Dešifrování je potom stíženo tím, že je třeba nejprve rozlišit, která z dat jsou vlastně užitečné zašifrované informace a co je pouhá neužitečná výplň.

Teorie složitosti, teorie informace a pravděpodobnost

Teorie složitosti je odvětvím teorie počítání v informatice a matematice, které se zaměřuje na klasifikaci výpočetních problémů dle jejich vlastní složitosti. V tomto kontextu je problém chápán jako úkol, který lze zpracovat na počítači. Problém tedy spočívá v zadání a jeho řešení. Základním příkladem je situace, kdy chceme zjistit zda-li je nějaké přirozené číslo n prvočíslem nebo ne. Zadáním tohoto jsou tedy přirozená čísla a výsledkem je odpověď ANO/NE případně 0 nebo 1.

S tímto problémem souvisí také analýza algoritmů a teorie vyčíslitelnosti. Hlavním rozdílem mezi analýzou algoritmů a teorií vyčíslitelnosti je že teorie vyčíslitelnosti se zabývá spíše výší finančních prostředků, které mimochodem potřebují samostatný algoritmus, zatímco analýza algoritmů se zabývá možnostmi všech možných použitelných algoritmů. Snaží se tedy klasifikovat problém podle omezenosti dostupných zdrojů. Podle pořadí a omezenosti zdrojů dává tedy odpověď na otázku zda úkol lze řešit algoritmicky či ne.

Teorie informace se zabývá přenosem, kódováním a měřením informace. Za tvůrce vzniku teorie informace se pokládá C. E. Shannon. K nejdůležitějším odvětvím teorie informace patří teorie přenosu informace.

Pravděpodobnost náhodného jevu je číslo, které je mírou očekávatelnosti výskytu jevu. Náhodným jevem rozumíme opakovatelnou činnost prováděnou za stejných (nebo přibližně stejných) podmínek, jejíž výsledek je nejistý a závisí na náhodě. Příklady mohou být například házení kostkou nebo losování loterie. Pravděpodobnost události se obecně označuje reálným číslem od 0 do 1. Událost, která nemůže nastat, má pravděpodobnost 0, a naopak jistá událost má pravděpodobnost 1. Někdy se kvůli názornosti pravděpodobnost uvádí v procentech, tedy setinách klasického vyjádření. Jinou používanou mírou pravděpodobnosti je šance (anglicky odds), která je definována jako poměr pravděpodobnosti definované běžným způsobem ku pravděpodobnosti, že nastane opačná událost: šance = p / (1 – p). Šance se často v praxi uvádí jako celočíselný zlomek, například „mám poloviční (1/2) šanci, že stihnu vlak" znamená totéž jako „je pravděpodobnost 0,5, že stihnu vlak" (a to samé, jako „pravděpodobnost

50%, že stihnu vlak"). Občas se pravděpodobnost udává též poměrově, srovnáním pravděpodobnosti výskytu dvou náhodných jevů: "mám šanci jedna ku jedné (1:1), že stihnu vlak". Například šance 2:3 je tedy rovna 2/(2+3).

Teorie pravěpodobnosti je pro kryptografii důležitá, protože kryptografické algoritmy často pracují s náhodnými čísly. Při šifrování jsou potom jako součást algoritmu generováná náhodná čísla. Aby bylo šifrování kvalitní, je třeba aby i vygenerovaná náhodná čísla vyla "kvalitní". Toho je bohužel na počíčíc dosahováno horko-těžko, protože počítačem generovaná náhodná čísla nejsou skutečně náhod á, nýbrž psedonáhodná. To znamená, že v generované posloupnosti lze nalézt jistou zákonitost - například po určitém počtu prvků se posloupnost opakuje. To může mít za následek zneužitelnou slabinu v šifrovacím algoritmu. Obtížnost nalezení této zákonitosti potom určuje zmíněnou kvalitu náhodnosti. Tu nazýváme mírou entropie. Pro zvýšení entropie se dá v algorimu využít například systémového šumu (stav operačního systému a jeho subsystémů v určitém okamžiku), či ještě lépe náhodných fyzikálních jevů (prostřednictvím externě připojeného elektronického zařízení).

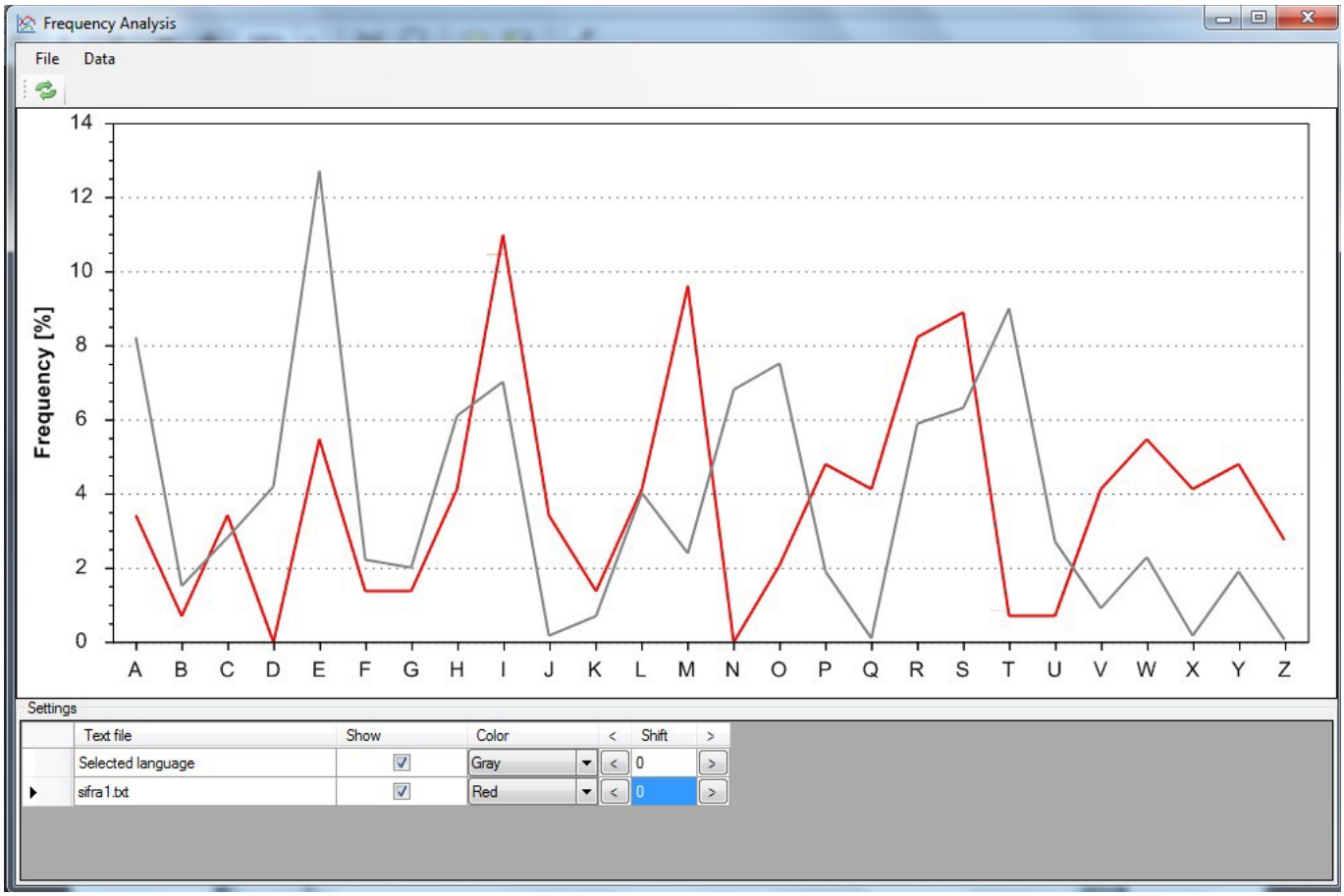
Základy kryptoanalýzy šifer

Kryptoanalýza moderních šifer je velice rozáhlé téma, ke kterému je třeba hluboká znalost související matematiky a kryptologie, jakožto i potřeba extensivních výpočetních prostředků. Zde si proto demonstrujeme pouze základní principy kryptoanalýzy příkladech některých jednoduchých šifer.

Kryptoanalýza monoalfabetické substituční šifry
Nejjednodušší šifrou je prostý posun. Každé z písmen v abecedě je posunuto o určitý počet míst. Každé z písmen abecedy je tedy zaměněno za písmeno jiné - a to vždy stejné. Protože každé z písmen je posunuto o stejný počet pozic, není třeba ani sestavovat žádnou složitější šifrovací tabulku a postačí nalézt, o kolik míst je posnunuto jedno z písmen - a tím získame posun pro všechna písmena. To můžeme provádět buď „hrubou silou", kdy budeme zkoušet postupně všechny posuny, až výsledný dešifrovaný text začne dávat smysl. To je však dosti pracné a neexaktní. Lepším způsobem, jak správný posun nalézt je pomocí frekvenční analýzy. Ta spočívá ve spočítání frekvence výskytů jenotlivých písmen v šifrovaném i otevřeném textu. Například anglickém i českém textu se nejčastěji vyskytují písmena E a A. Nejčastěji se vyskytující písmeno v šifrovaném textu bude s největší pravěpodobností odpovídat nejčastěji se vyskytujícímu písmenu v daném jazuce. Proto můžeme zkusit přiřadit písmeno E nejčasteji se vyskytovanému znaku v šifrovaném textu, spočítat posun v abecedě a o tento posun posunout všechny znaky v šifrovaném textu. S nejvyšší pravděpodobností vyjde otevřený text, který dává smysl. Nemusí tomu tak být, ale je to velmi pravděpodobné a okruh pravděpodobných posunů se nám tím dosti zmenší.

Kryptoanalýza substituce s klíčem
Tahle šifra už je složitější. Každému z písmen abecedy je přiřazeno písmeno jiné. Klíčem je potom anglická abeceda s písmeny v jiném pořadí. Stačí tedy nalézt, které písmeno odpovídá kterému, což nemusí být zas tak úplně jednoduché. Postup je velice podobný předešlému případu s prostým posunem, ale nestačí "odhalit" pouhé jedno písmeno, ale nějakým způsobem všechny.
Opět je doporučováno provedení frekvenční analýzy zašifrovaného textu - ručně nebo pomocí nějakého automatizovaného nástroje. Ten potom umožňuje získanou statistiku zobrazit do grafu – a také zobrazovat více grafů pro různé vstupy zároveň. Opticky tak můžeme porovnat špičky výskytů některých písmen našeho grafu s grafem statistiky dané (třeba anglické) abecedy a posunout náš graf tak, aby se oba grafy v průběhu víceméně shodovaly. Princip je ten, že některá písmena (hlavně samohlásky) se vyskytují o hodně častěji než jiná. Je dobré se orientovat především podle písmen A, E, I a T, jelikož ta se v Angličtitě vyskytují nejčastěji. Dále lze využít znalosti kolokace (spoluvýskytu) některých dvojic písmen, rytmu opakování některých častých písmen (samohlásek) v daném jazyce apod. Postupně budujeme dešifrovací tabulku, kterou aplikujeme na šifrovaný text a hledáme v

dešifrovaném textu smysl. S využitím vhodných nástrojů a zapojením mozku není problém s troškou úsilí šifrovaný text po jistém čase úspěšně dešifrovat.



Graf frekvenční analýzy

Kryptoanalýza afinní šifry

Afinní je opět substituční šifra, která ovšem eliminuje základní nevýhodu Casesarovy šifry (tedy prostého posunu) – a to málo možností transformace. Nalezení posunu hrubou silou je tedy poněkud náročnější. Jednotlivým písmenům jsou přiřazena písmena jiná dle vzorce:

$$Ci = a \cdot Ti + b \pmod m$$

Ci - zašifrované písmeno
Ti - původní písmeo
a, b - koeficienty tvořící klíč šifry
m - počet písmen dané abecedy

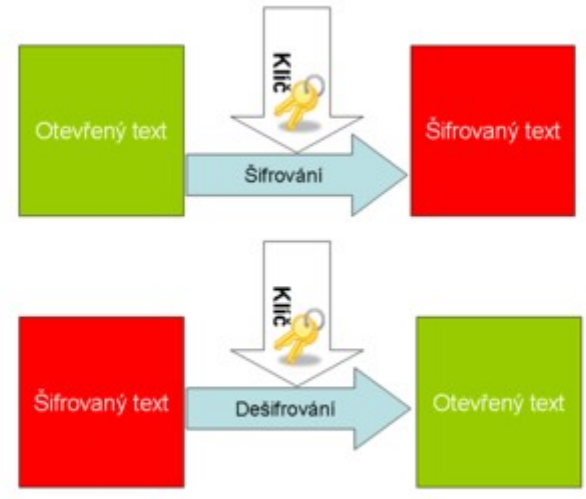
Při dešifrování je tedy potřeba nalézt koeficinty a a b – multiplikátor a přičítanou konstantu. Můžeme hledat hrubou silou, ale to je časově dosti náročné. Proto lze opět vyjít z předpokladu znaků daného jazyka (nejčasteji se vyskytovaných písmen, rytmus samohlásek) a frekvenční analýzy. Za předpokladu, že máme k dispozici dosti dlouhý šifrovaný text, který poskytne dostatečná statistická data lze opět vytipovat některá písmena a sestavit pravděpodobné rovnice dle výše uvedeného vzorce. Potom již jen stačí vyřešit soustavu rovnic a najít koeficienty. Možných kombinací rovnic je třeba většinou vytipovat vícero a tím získat více možných koeficientových dvojic. Jejich aplikací šifrovaný text poté můžeme ověřit, zda davají smysluplné výsledky.

Symetrické šifrování, proudové šifry a blokové šifry

Symetrické šifrování pracuje na velmi jednoduchém principu. Pro šifrování i dešifrování datového řetězce (zprávy) je použit algoritmus, který pro obě činnosti používá jediný *tajný klíč*. Ten se také nazývá symetrický tajný klíč, kvůli jeho povaze (použití pro šifrování i dešifrování, tedy oběma směry) nebo sdílený tajný klíč, protože je sdílen všemi zainteresovanými subjekty - odesílatelem zprávy i přijemci. Tento klíč si subjekty musí nejříve vyměnit buď osobně nebo prostřednictvím zabezpečeného kanálu. Je velmi důležité, aby tento klíč nezískal nikdo jiný. Symetrickému šifrování se též říká šifrování s tajným klíčem.

Výhodou symetrických šifrovacích algoritmů je jejich nízká výpočetní náročnost a z toho plynoucí vysoká rychlost šifrování.

Nevýhodou je již zmíněná nutnost se předem dohodnout na tajném klíči, což může být obtížné, protože k tomu je třeba užít nějakého bezpečného kanál - a ten se právě snažíme vytvořit.



Symetrické šifry se dělí na dva druhy. *Proudové šifry* zpracovávají otevřený text po jednotlivých bitech. *Blokové šifry* rozdělí otevřený text na bloky stejné velikosti a doplní vhodným způsobem poslední blok na stejnou velikost. U většiny šifer se používá blok o 64 bitech, AES používá 128 bitů.

Proudové šifry provádějí šifrování datového řetězce po jednotlivých bitech, každý z nich zpracovávají zvlášť. Vstupní datový tok je kombinován (typicky pomocí funkce XOR) s pseudonáhodným proudem bitů (anglicky keystream) vytvořeným z šifrovacího klíče a daného šifrovacího algoritmu. Výsledkem je zašifrovaný datový tok (proud). Proudové šifry jsou typicky rychlejší než blokové šifry a pro implementaci potřebují jednodušší hardware. Naopak jsou na rozdíl od blokových šifer náchylnější ke kryptoanalytickým útokům, pokud jsou nevhodně implementovány.

Blokové šifry jsou symetrické šifry, které data šifrují/dešifrují po blocích pevně stanovené délky, zpravidla dlouhých 8 B. Při zpracování se datový proud rozdělí na více bloků, přičemž pokud jsou data vstupující do posledního bloku kratší, je do zbylého místa přidána výplň (anglicky padding). Při šifrování je každý blok transformován pomocí šifrovacího algoritmu řízeného utajeným šifrovacím klíčem. Dešifrování probíhá stejným postupem – zašifrované bloky stejné délky jsou postupně rozšifrovány stejným šifrovacím algoritmem pomocí stejného utajeného šifrovacího klíče. Z matematického hlediska je použita při dešifrování funkce inverzní k dané šifrovací. Slabinou blokových šifer je, že i když potenciální útočník nemůže vidět do šifrovaného textu, může hypoteticky zaútočit tak, že zpřehází pořadí jednotlivých bloků. Tomu se blokové šifry brání vázáním po sobě následujících bloků. Způsob

tohoto vázání určuje tzv. mód šifry. Těch bylo vyvinuto povícero a liší se náročností provedení a stupněm bezpečnosti. Často používanými módy jsou např. módy CBC, ECB...

Některé populární šifry

RC4 - jedná se o proudovou šifru, navrhl ji Ron Rivest z RSA Security v roce 1987. Je používána například pro šifrovaný přenos webových stránek (HTTPS) nebo pro zabezpečení přenosu v bezdrátových sítích Wi-Fi (šifrování WEP). Šifra RC4 je jednoduchá, rychlá, ale má i slabé stránky, které omezují její použitelnost.

FISH - velmi rychlá proudová šifra navržená společností Siemens v roce 1993. Bylo zjištěno, že není příliš bezpečná, protože k jejímu prolomení stačí znát několik tisícovek bitů otevřeného textu.

DES (Data Encryption Standard) - je zřejmě nejznámějším představitelem symetrických blokových šifer, vznikla v roce 1976 a představuje dřívější průmyslový standard pro šifrování, dnes již šifra není považována za bezpečnou (byla prolomena v roce 1997) a byla nahrazena modernějším standardem AES.

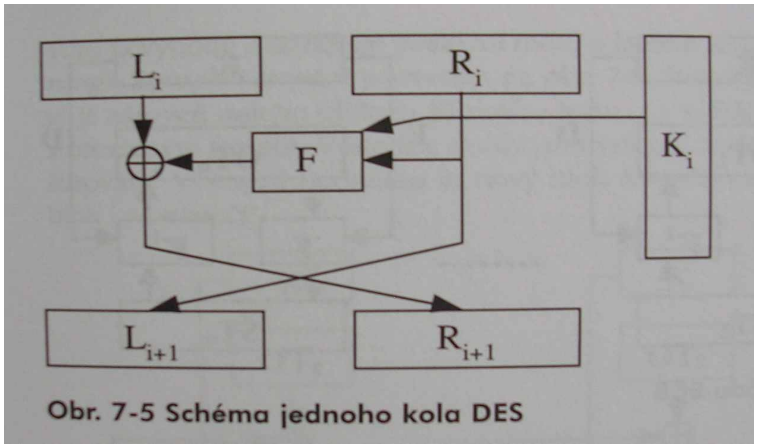
3DES - Triple DES je bloková šifra založená na šifrování Data Encryption Standard (DES), které aplikuje třikrát a tak zvyšuje její odolnost proti prolomení. Je trojnásobnou aplikací šifry DES a pracuje s klíčem o celkové délce 168 bitů. Jde sice o poměrně dost silnou a bezpečnou šifru, nicméně poněkud výpočetně náročnou. Byla nahrazena AES.

AES - bloková šifra, vyvinuta Joanem Daemenem a Vincentem Rijmenem pod původním názvem Rijndael. Poté byla vybrána ve veřejné soutěži NIST jako nový šifrovací standard, který je nazýván AES. Délka klíče může být 128, 192 nebo 256 bitů. Metoda šifruje data postupně v blocích s pevnou délkou 128 bitů. Šifra se vyznačuje vysokou rychlostí šifrování. V současné době není veřejně znám žádný případ jejího plného prolomení.

Twofish je symetrická bloková šifra se 128-bitovou délkou bloku a až 256-bitovou délkou klíče, vyvinutá Brucem Schneierem. Jde o nepatentovanou otevřenou šifru pro volné použití. Šifra Twofish byla jedním z pětice finalistů soutěže standardu AES.

Blowfish - předchůdce Twofish, publikována v roce 1993 Brucem Schneierem.

IDEA - International Data Encryption Algorithm - je symetrická bloková šifra, kterou navrhli Xuejia Lai a James L. Massey ze Švýcarského národního technologického institutu (ETHZ) v Zürichu v roce 1991. Algoritmus měl nahradit Data Encryption Standard. IDEA je drobným přepracováním dřívější šifry PES (Proposed Encryption Standard), původně se nazývala IPES (Improved PES). Je na ní založen šifrovací algoritmus PGP.



Obr. 7-5 Schéma jednoho kola DES

Hašovací funkce

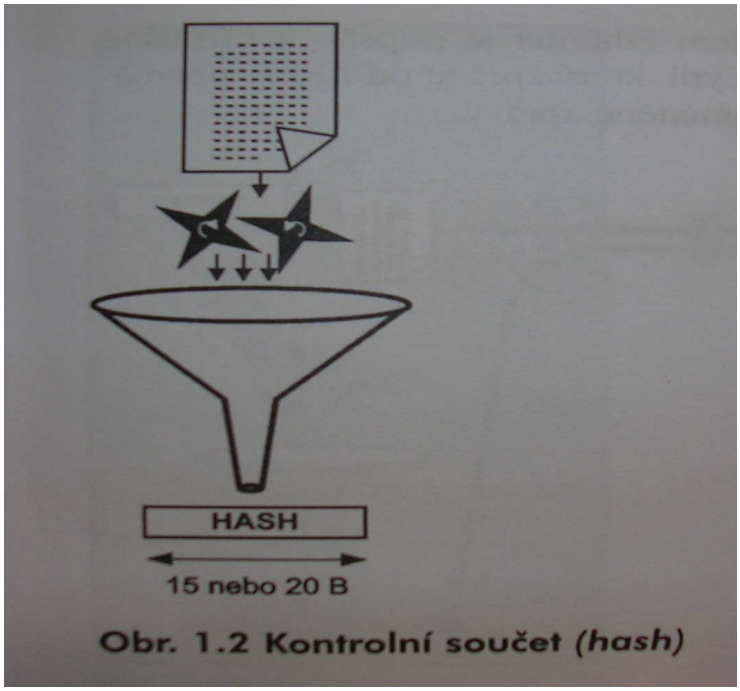
Hash - do češtiny lze překládat jako kontrolní součet, miniatura, či otisk (fingerpring).

Hashovací funkce je je jednocestná funkce, která provede výpočet nad libovolně dlouhým řetězcem dat (např. textovým řetězcem či jakoukoli jinou posloupností bytů) a vrátí řetězec konstantní délky. Výsledný řetězec se nazývá hash.

Jednocestnou funkcí je matematická funkce, která je tvořena algoritmem, jenž umožňuje technicky snadno spočíst požadovaný výsledek, avšak najít k danému výsledku původní řetězec je výpočetně extrémě náročné, až nemožné. Obecně tedy neexistuje inverzní funkce, která by ke spočtenému kontrolnímu součtu jednoznačně nalezla původní vstupní posloupnost. Ve skutečnosti lze ke každému kontrolnímu součtu nalézt teoreticky nekonečně mnoho původních textů, což plyne ze samé postaty funkce - možných variací vstupních dat je nekonečně mnoho, zatímco možných výsledků je omezeně mnoho - funkce není prostá. I kdybychom tedy nějaké nazlezli, nevíme, který z nalezených řetězců je ten původní.

Kvalitní funkce pro výpočty kontrolních součtů by měly dávat výrazně rozdílný výsledek i při drobné změně původních dat. Jsou konstruovány na výpočetních operacích nízké úrovně, jako jsou bitové operace a posuny, proto jsou (pro určený směr) velmi efektivní a rychlé.

Algoritmy pro výpočet kontrolního součtu nejsou v žádném případě šifrovacími algoritmy, i když spousta lidí si to plete. Příkladem je ukládání hesel do souboru v "šifrované" podobě (např v operačních systémech unixového typu), aby nebylo přečtením souboru možno zjistit skutečná hesla. Ve skutečnosti nejde o zašifrovaná hesla, ale pouze o jejich hashe.



Obr. 1.2 Kontrolní součet *(hash)*

Praktické využití

Bezpečné ukládání hesel – tady je přínos zřejmý - pokud jsou hesla uložena na počítačovém systému přímo ve své skutečné podobě, je možné je po získání přístupu na tento stroj jednoduše přečíst a získat tak hesla všech uživatelů. Pokud jsou zde ale uloženy pouze jejich otisky (hashe), nelze skutečná hesla takto snadno získat.

Bezpečný přenos hesel - uživatel při přihlašování se ke vzálenému systému zadává pro účely authentikace své osoby svoje heslo. To, pokud je přenášeno ve své čisté formě, může být na cestě sítí zachyceno (odposlechnuto) a následně zneužito. Pokud je ale přenášen pouze hash tohoto hesla, jeho prozrazení logicky nehrozí. Zase tak jednoduché to ale není, protože pokud vzdálený systém očekává od uživatele hash jeho hesla, útočník sice nemůže zjistit heslo samotné, ale to mu nevadí, protože mu postačí znalost hashe - po jeho odposlechnutí může zasílat přímo tento hash a bude authentikován, aniž by vůbec potřeboval původní podobu hesla znát. Proto se používá ještě o něco sofistikovanější algoritmus, tzv. bezpečná authentizace.

Bezpečná authentizace funguje na principu, který sestává z následujících kroků:

- uživatel se hodlá přihlásit ze svého počítače na vzdálený systém
- z jeho vlastního lokálního počítače je na ten vzdálený vyslán požadavek na authentizaci
- vzdálený systém vygeneruje náhodné číslo (tzv. chalenge) a to zašle ho zpět uživateli
- zároveň ho spojí s lokálně uloženým heslem uživatele a z zohoto celého řetezce vytvoří hash (ten nikam neposílá, nechá si ho pouze pro vlastní potřebu)
- systém na straně uživatele převezme chalenge a také ho spojí, tentokráte s heslem, které uživatel zadá z klávesnice a také z tohoto spojení vztvoří hash - ten obratem zašle vzdálenému počítači
- vzdálený počítač převezme hash vzprodukovaný uživatelem a porovná ho s tím, který vztvořil sám
- pokud se oba hashe shodují, uživatelel je autentizován, pokud se liší, authentizace není úspěšná

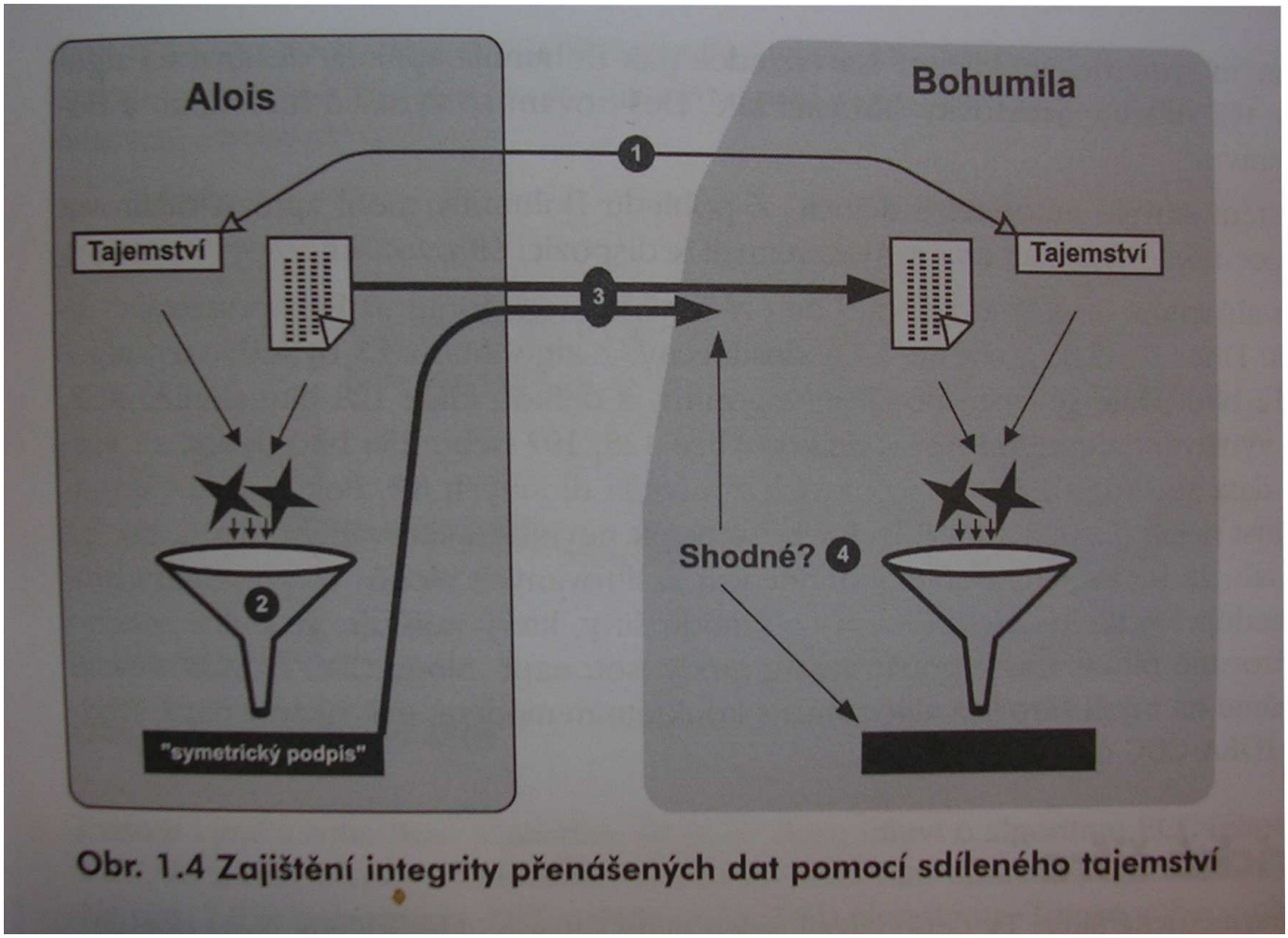
Protože chalenge se generuje pro každé jednotlivé přihlášení znovu, je pro každou danou session unikátní a proto případné zachycení přenášeného hashe není útočníkoví k ničemu.

Ověřování integrity uložených dat – některé weby, které nabízejí ke stažení velké soubory, zároveň poskytují jejich hashe; pokud si uživatel takový soubor stáhne, může si pomocí něj ověřit, že stáhl soubor celý a korektně, tedy bez chyb jednoduše tím, že si sám na své straně vytvoří jeho hash a oba hashe porovná – pokud se shodují, je soubor v pořádku

Ověřování integrity přenášených zpráv – například, pokud je napříč počítačovou sítí přenášen datový paket, může dojít cestou k jeho poškození; proto je na konec každého paketu do zápatí přidáván jeho kontrolní součet, tedy hash; příjemce si potom spočítá kontrolní součet z přijatého paketu sám a porovná ho s tím přijatým – pokud se shodují, je paket bez chyb a je přijat, pokud se neshodují, došlo k jeho poškození a je zahozen.

Symetrický podpis - Hash lze také použít k realizaci symetrické varianty ekektronického podpisu. Pomocí něj lze zajistit, že zaslaná zpráva nebyla po cestě nikým změněna. Funguje to stejně jako prosté ověřování integrity přenášených zpráv, s tím že navíc oba subjekty (odesílatel a příjemce) sdílejí společné tajemství, které se před vytvořením hashe z přenášené zprávy s touto zprávou žřetězí. Přenášena je potom samozřejmě pouze samotná zpráva a tento hash. Za předpokladu, že sdílené tajemství nikdo jiný nezná, není ani možné, aby byl schopen z dané zprávy příslušný hash vytvořit. Nevýhodou opět je, že sdílené tajemství si musí oba subjekty předem vyměnit pomocí nějakého

bezpečného kanálu, například při osobní schůzce.



Obr. 1.4 Zajištění integrity přenášených dat pomocí sdíleného tajemství

Některé známé a používané hashovací funkce:

CRC32 - pro svou jednoduchost a dobré matematické vlastnosti jde o velmi rozšířený způsob realizace kontrolního součtu. Vytváří kontrolní součet délky 32 bitů.

MD5 - Message-Digest algorithm - široce používaná hashovací funkce z roku 1991, v roce 1996 byla nalezena první kolize (nalezení stejného výsledku pro dvě různé množiny dat) a tím prokázáno, že by funkce neměla být považována za zcela bezpečnou. Vytváří otisk dlouhý 128 bit, což již dnes není postačující.

SHA-1 - navržena organizací NSA (National Security Agency), od roku 2002 defacto standard. Používá se u některých zabezpečených přenosových protokolů a aplikací, včetně kontroly integrity souborů nebo ukládání hesel. Je považována za nástupce hašovací funkce MD5. Produkuje hash dlouhý 20 B, tedy 160 bitů. Bezpečnost SHA-1 byla již též zpochybněna kryptografickými odborníky, ačkoli nebyly oznámeny žádné úspěšné útoky.

SHA-2 - označení pro množinu hned několika variant hashovacích funkcí, které jsou jsou algoritmicky shodné s SHA-1. Zahrnuje čtyři hashovací funkce SHA, které jsou pojmenovány podle své délky v bitech: SHA-224, SHA-256, SHA-384 a SHA-512. Je doporučováno je používat namísto SHA-1.

Základní principy asymetrického šifrování a výměny klíčů

Asymetrická kryptografie (kryptografie s veřejným klíčem) je skupina kryptografických metod, ve kterých se pro šifrování a dešifrování používají odlišné klíče. To je základní rozdíl oproti symetrické kryptografii, která používá k šifrování i dešifrování jediný klíč.

Kromě očividné možnosti použití pro utajení komunikace se asymetrická kryptografie používá také pro elektronický podpis, tzn. možnost u dat prokázat jejich autora.

Asymetrické šifrování tedy nepoužívá pouze jeden tajný klíč sdílený mezi odesilatelem a příjemcem, ale vždy klíče dva - jeden pro šifrování a druhý pro dešifrování.
Protože u asymetrických šifer jsou operace šifrování a dešifrování obecně zaměnitelné (co zašifrujeme prvním klíčem, to lze dešifrovat druhým a co zašifrujeme druhým, to lze dešifrovat prvním), nemluvíme obyčejně o klíči šifrovacím a dešifrovacím, ale o klíči soukromém a klíči veřejném. Klíč veřejný je potom používán pro šifrování a soukromý pro dešifrování. Asymetrickému šifrování se též říká šifrování s veřejným klíčem.
Princip je takový, že uživatel, který chce přijímat zašifrované zprávy si vygeneruje takovýto pár klíčů - soukromý si pečlivě uloží na bezpečné místo a veřejný veřejně vystaví (například na svých webových stránkách). Odesílatel, který chce tomuto uživateli zaslat šifrovanou zprávu si jednoduše stáhne jeho veřejný klíč a použije ho k zašifrování odesílané zprávy; poté zprávu odešle. Dešifrovat ji je možné pouze příslušným soukromým klíčem, který nemá k dispozici nikdo jiný než jeho právoplatný majitel - znalost veřejného klíče k dešifrování nestačí.

Výhodou oproti symetrickému šifrování je, že odpadá potřeba výměny tajného klíče bezpečným kanálem - veřejný klíč může mít k dispozici opravdu kdokoli, aniž by hrozilo nežádoucí rozšifrování přenášené zprávy třetí stranou.

Nevýhodou asymetrických šifer je jejich značná výpočetní náročnost, jsou řádově asi 100-krát pomalejší než šifry symetrické. Důvodem je kromě algoritmu samého také potřeba značně delšího klíče. U algoritmu RSA se považuje za bezpečnou délka klíče minimálně 1024 bitů.

Pokud spolu chtějí dva subjekty obousměrně komunikovat, je třeba, aby každý z nich disponoval párem asymetrických klíčů. Svůj soukromý si každý z nich uchová v tajnosti a veřejný poskytne protější straně, což lze učinit jeho prostým veřejným vystavením, není třeba ho chránit před třetími stranami.
Při odesílání potom každý z odesílatelů šifruje pomocí veřejného klíče protější strany a přijaté zprávy dešifruje svým vlastním soukromým klíčem. Výměna klíčů je tedy v případě asymetického šifrování jednoduchá a bezproblémová.

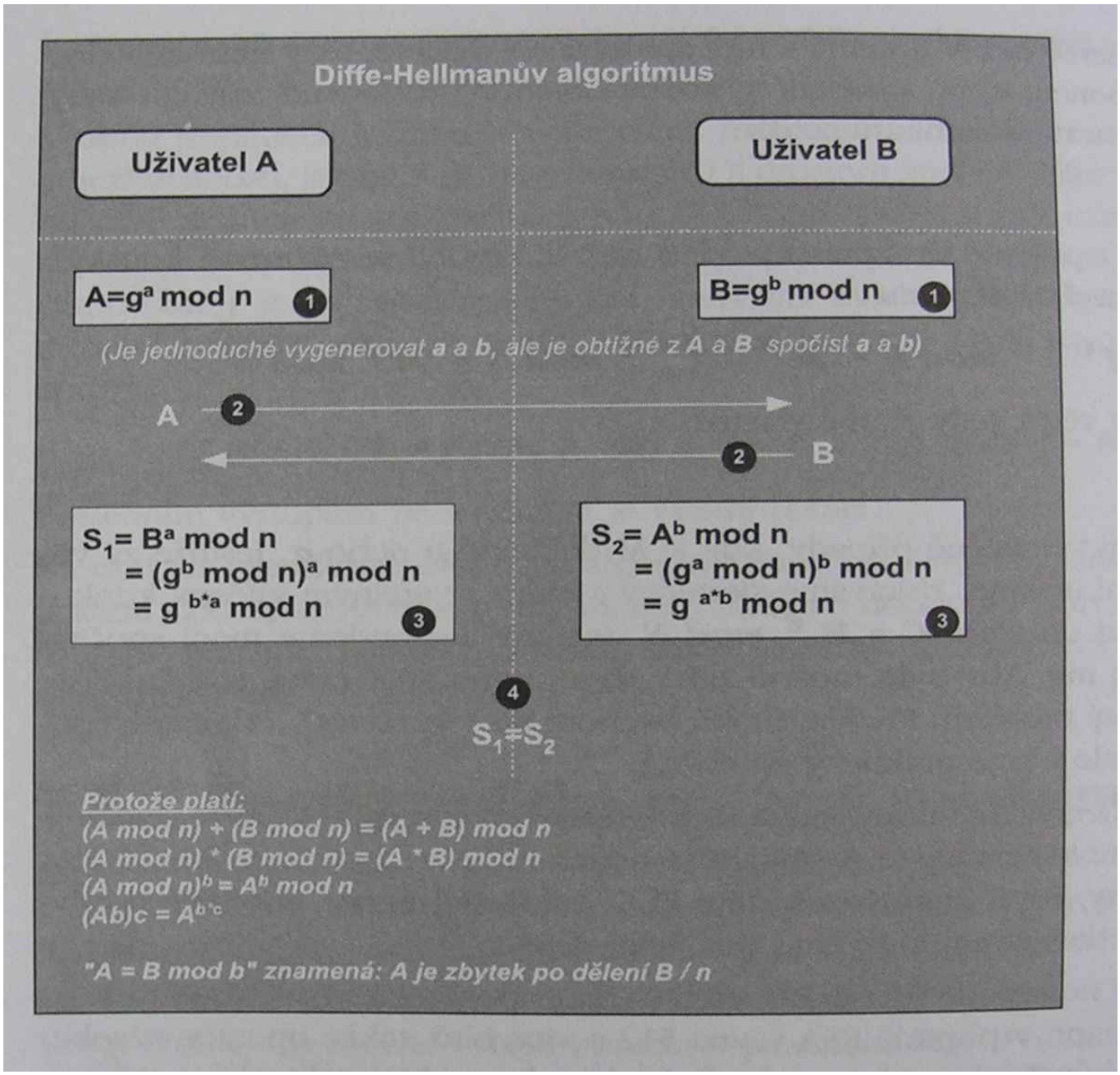
Typickým představitelem šifry s veřejným klíčem je *RSA* (iniciály autorů Rivest, Shamir, Adleman). Jedná se o první algoritmus, který je vhodný jak pro šifrování, tak podepisování. Dnes se stále používá, přičemž při dostatečné délce klíče je považován za bezpečný.

Princip bezpečnosti RSA je postaven na předpokladu, že rozložit velké číslo na součin prvočísel (faktorizace) je velmi obtížná úloha. Z čísla n = pq je tedy v rozumném čase prakticky nemožné zjistit činitele p a q, neboť není znám žádný algoritmus faktorizace, který by pracoval v polynomiálním čase vůči velikosti binárního zápisu čísla n. Naproti tomu násobení dvou velkých čísel je elementární úloha. Z toho plyne, že šifrování je relativně rychlé, zatímco dešifrování je pro dostatečně velká čísla p,q i s použitím výpočetní techniky extrémě časově náročné, prakticky nemožné.

Protože asymetické šifry jsou příliš pomalé a pro šifrování velkého množství dat při přenosu v reálném čase příliš těžkopádné a náročné na výpočetní prostředky, je jich užíváno ve spojení se symetrickými

šiframi pro výměnu klíčů. Na začátku komunikace si obě strany pomocí asymetrického šifrování dohodnou symetrický klíč, ten oproti samotným přenášeným datům není vůbec velký a proto to není z časového hlediska žádný problém - a následná komunikace probíhá již pomocí symetrického šifrování, které je řádově rychlejší. Symetrické klíče můžou být vyměněny a dohodnuty i dva - záleží na použitém algoritmu - obě strany si mohou data navzájem šifrovat jedním klíčem nebo každá z nich může pro svůj směr komunikace používat jiný.

Pro výměnu klíčů lze užít též *Diffie-Hellmanova algoritmu*.
Diffie-Hellman (D-H) výměna klíčů je kryptografický protokol, který umožňuje přes nezabezpečený kanál vytvořit mezi komunikujícími stranami šifrované spojení, bez předchozího dohodnutí šifrovacího klíče. Výsledkem tohoto protokolu je vytvoření symetrického šifrovacího klíče, který může být následně použit pro šifrovaní zbytku komunikace. Výhodou je, že případný útočník odposlouchávající komunikaci tento klíč nezachytí. Klíč je zkonstruován všemi účastníky komunikace a nikdy není poslán v otevřené formě. Nevýhodou tohoto protokolu je bezbrannost proti útoku Man in the middle, protože neumožňuje autentizaci účastníků. Tento protokol bez kombinace s jinými metodami je tedy vhodný pouze tam, kde útočník nemůže aktivně zasahovat do komunikace. V příkladu je ukázáno vytvoření zabezpečeného spojení mezi dvěma účastníky. Počet účastníků však není omezen.



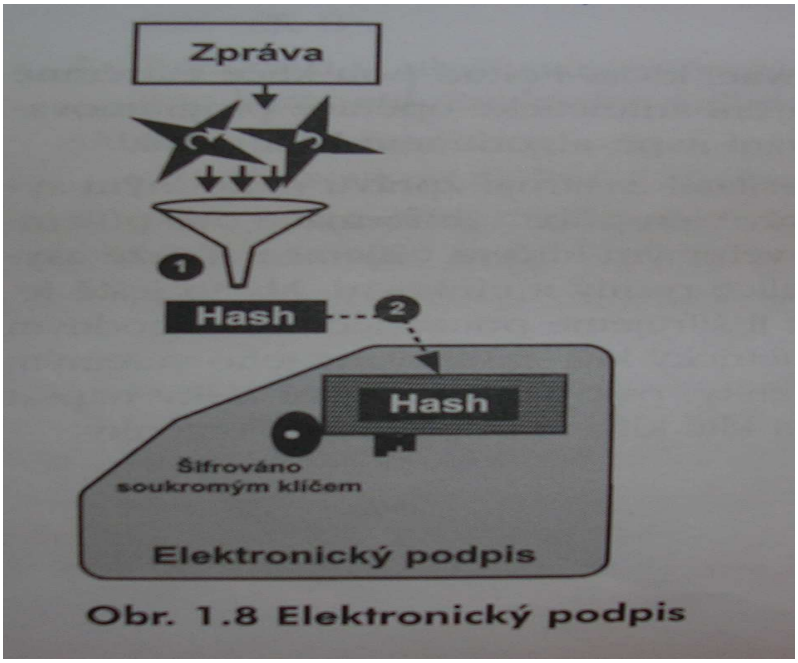
Digitální podpisy

Kromě symetrického podpisu, který byl již zmíněn v části o symetrickém šifrování se též používají podpisy založené na šifrování asymetrickém - a nutno poznamenato, že mají mnohem větší význam. Elektronické podpisy se používají pro možnost ověření pravosti a původce zasílané zprávy.

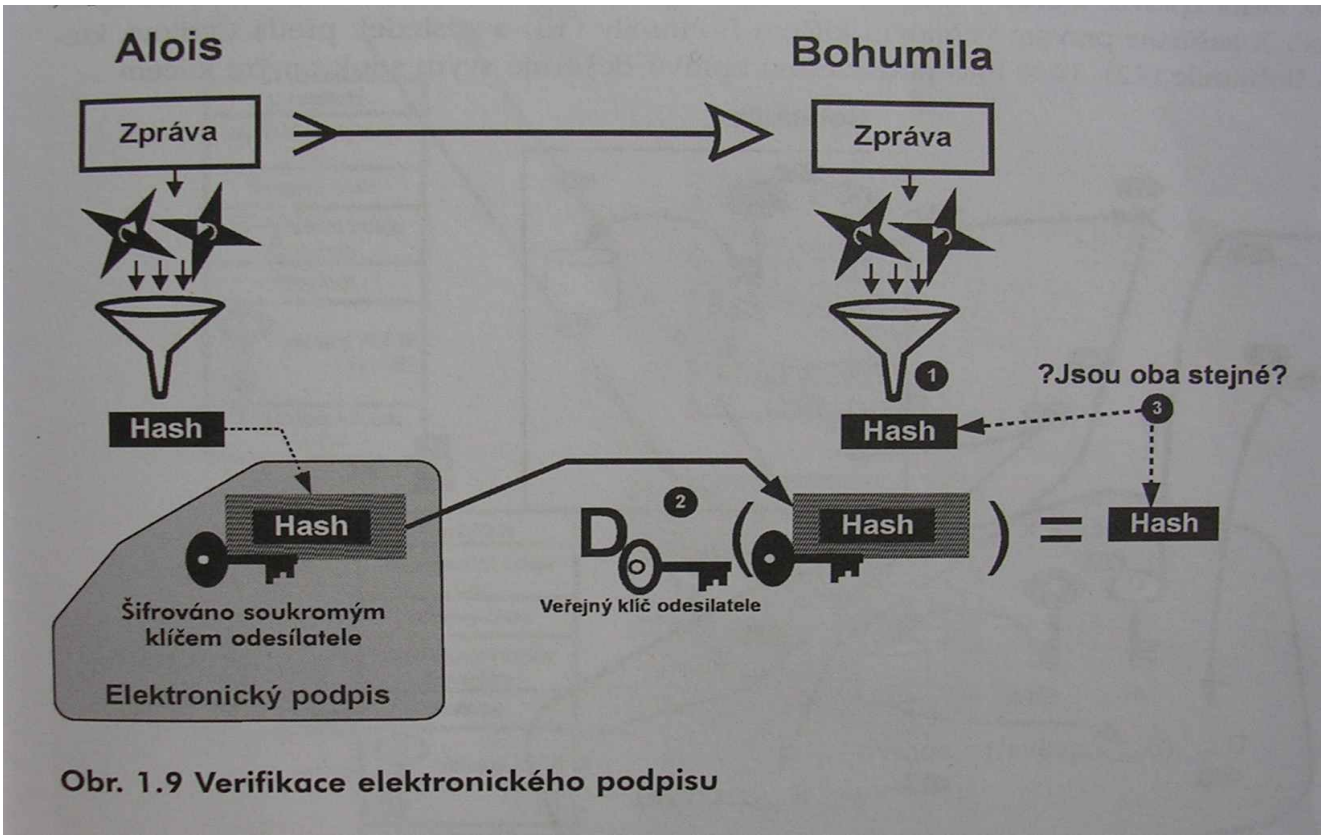
Asymetrický elektronický podpis funguje z principu tak, že odesítalel dokumentu, který chce podepsat, ho zašifruje ne veřejným klíčem příjemce jako při šifrování, nýbrž svým vlastním soukromým klíčem. Tento dokument potom může naopak kdokoli rozšifrovat pomocí příslušného majitelova klíče veřejného. Takový postup sice nezajišťuje utajení dokumentu, ale zajišťuje ověření, že byl skutečně odeslán deklarovaným odesílatelem, protože nikdo jiný nevlatní párový soukromý klíč ke klíči veřejnému, kterým dokument dešifrujeme.
V praxi se takovýto prostý princip ovšem nepoužívá, protože zasílaná zpráva může být hodně velká a šifrování a dešifrování zabere nějaký čas. Pokud by k tomu zpráva navíc byla ještě zašifrována, docházelo by k dvojímu šifrování a dešifrování - jednou by byla zašifrována veřejným klíčem příjemce kvůli utajení, podruhé soukromým klíčem odesílatele kvůli možnosti ověření pravosti - a poté opět dvakrát dešifrována odpovídajícími párovými klíči. Proto se používá způsob trošku jiný, který zajišťuje stejný princip, ale trošku efektivněji. Postup podepsání a následného ověření elektronické zprávy či dokumentu tedy je následující:

- na straně uživatele je z inkriminované zprávy spočten hash (tedy její otisk)
- hash je zašifrován odesílatelovým soukromým klíčem, tím vznikne elektronický podpis dané zprávy
- vzniknuvší podpis je přidán na konec zprávy a ta může být nyní předána dále (např. odeslána)
- příjemce přijme zprávu s podpisem; ze samotné zprávy spočítá její otisk
- příjemce veřejným klíčem odesílatele (který má k dispozici každý) rozšifruje podpisovou část a tím získá původní hash spočtený na straně odesílatele
- oba hashe porovná a pokud jsou shodné, zpráva zaručeně pochází od zamýšleného odesílatele a nebyla po cestě změněna nikým jiným

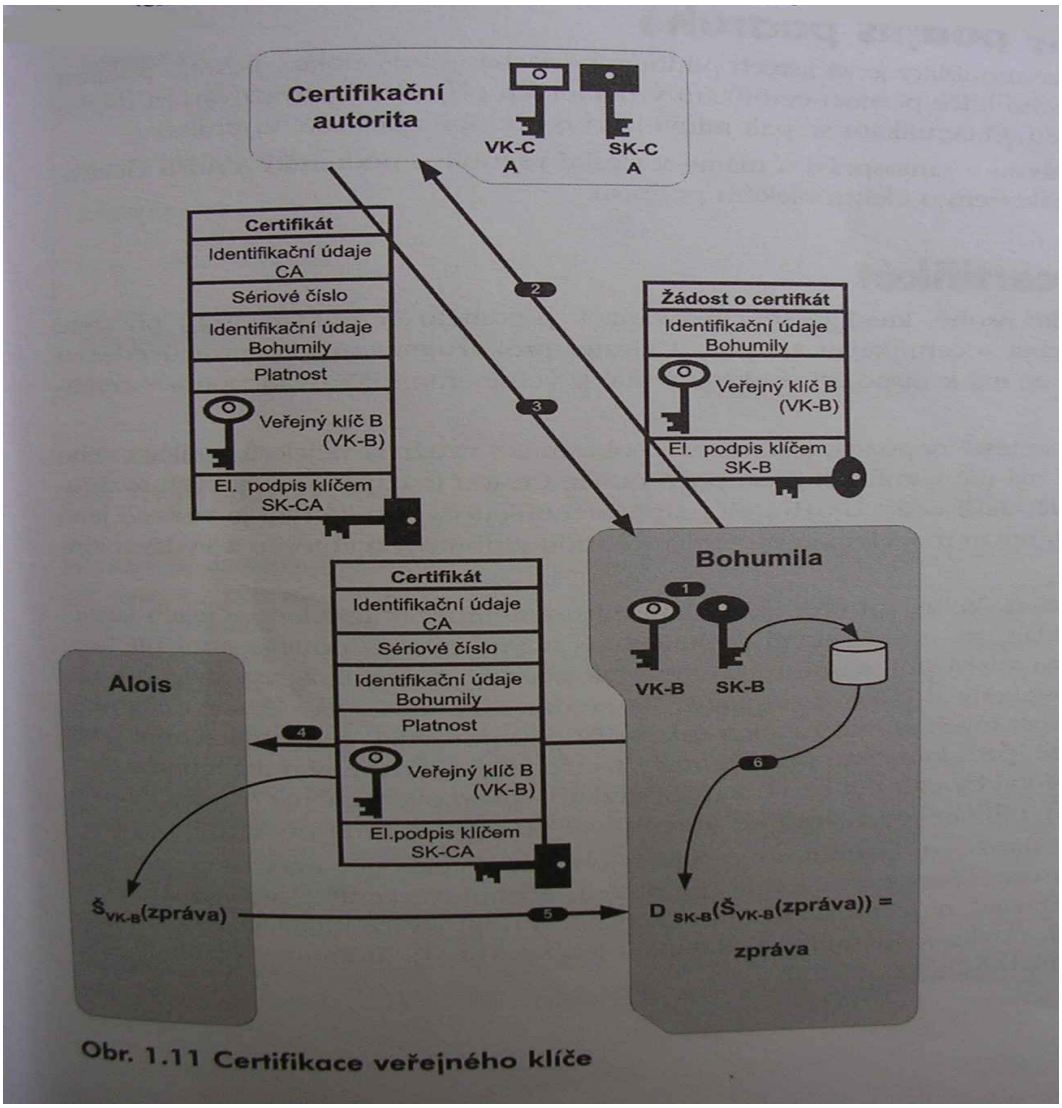
Asymetrické podpisy poskytují totiž narozdíl od těch symetrických princip nepopiratelnosti - neboli neodmítnutelné odpovědnosti. Soukromý klíč má totiž k dispozici pouze sám majitel a nikdo jiný (pokud samozřejmě nedojde k jeho kompromitaci), a proto nikdo jiný kromě něho nemůže vytvořit podpis jeho jménem.



Obr. 1.8 Elektronický podpis



Obr. 1.9 Verifikace elektronického podpisu



Obr. 1.11 Certifikace veřejného klíče

Kryptografie eliptických křivek

Jako alternativa algoritmu RSA se dá pro asymetrické šifrování využít také *algoritmus eliptických křivek* (ECC). Objevena N. Koblitzem roku 1985. Ten opět využívá rozkladu na prvočísla.

Kryptografie na bázi eliptických křivek je systém asymetrické kryptografie (tedy kryptografie s veřejným klíčem), který je založený na algebraické struktuře eliptických křivek nad konečnými poli. Použití eliptických křivek v kryptografii navrhli nezávisle Neal Koblitz a Victor S. Miller v roce 1985.

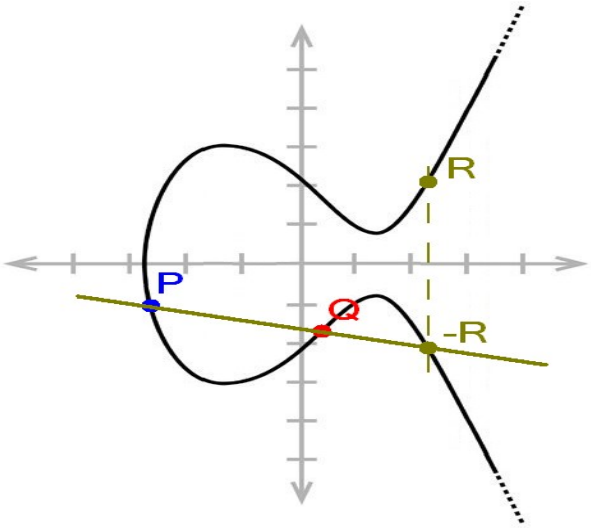
Systém je založený na velmi těžké řešitelnosti některých matematických problémů. U protokolů založených na eliptických křivkách sa předpokladá, že je nemožné nalezení diskrétního logaritmu prvku eliptické křivky. Velikost eliptických křivek potom určuje obtížnost problému. Předpokladá se, že na získaní rovnocenné úrovně zabezpečení jako u RSA systému je možno použít menší grupu. Použití malých grup redukuje požadavky na ukladání a přenos.

Obecně se míní, že z zhlediska bezpečnosti délka 160 bitů klíče algoritmu ECC odpovídá 1024 bitů dlouhému klíči algoritmu RSA.

Algebraická křivka: Uvažme množinu bodů v rovině splňující rovnost y2=ax3+bx2+cx+d a na této křivce uvažme bod O. Nyní budeme definovat sčítání bodů na této křivce.

Eliptická křivka: Eliptickou křivkou budeme rozumět již zmíněnou množinu bodů spolu s bodem O a operací +. Každou eliptickou křivku můžeme převést na tvar y^2=x^3+ax+b (Weierstrassova forma), kde bodem O bude nevlastní bod přímky x=0.

Princip: Máme-li bod P na nějaké eliptické křivce. Zvolíme libovolné (velké) přirozené číslo r a spočítáme rP, což je nějaký bod Q eliptické křivky. Nyní pokud bychom chtěli z bodů P a Q určit číslo r, je to velmi obtížné. Tomuto úkolu se říká Problém diskrétního logaritmu.
Obě strany komunikace si dohodnou bod P na eliptické křivce. Každá strana si zvolí přirozené číslo k (privátní klíč) a vypočítají Q=kP, což bude veřejný klíč. Pokud tedy Ondra má privátní klíč a, veřejný klíč A=aP, Zdenda má privátní klíč b, veřejný klíč B=bP. Potom tedy pokud dostanu od Zdendy veřejný klíč B a spočítám Z=aB=abP, dostanu stejný bod, který spočítá Zdenda, když od Ondry dostane bod A, tedy bA=baP=Z.



Vypracoval: Petr Manoušek pro účely studia v uzavřeném kruhu studentů ČVUT-FEL
Použité zdroje: Velký průvodce protokoly TCP/IP - Bezpečnost, Wikipedia, zpravodaj ÚVT MU
Poslední revize: 2011-11-27