Univerzita Karlova v Praze

Filozofická fakulta

Ústav informačních studií a knihovnictví

Studijní program: Informační studia a knihovnictví

Studijní obor: Studia nových médií

Bc. Tomáš Tománek

Aplikovaná kryptologie v internetové komunikaci

Diplomová práce

Praha 2008

Vedoucí diplomové práce: Ing. Martin Souček

Oponent diplomové práce:

Datum obhajoby:

Hodnocení:

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje.

V Praze, 14. dubna 2008

Tománek

……..

podpis diplomanta

TOMÁNEK, Tomáš. Aplikovaná kryptologie v internetové komunikaci [Applied cryptology during internet communication]. Praha, 2008. 112 s. Diplomová práce. Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví 2008. Vedoucí diplomové práce Ing. Martin Souček.

Abstrakt

Tématem diplomové práce je Aplikovaná kryptologie v internetové komunikaci. Práce si klade za cíl přinést přehled možných rizik internetu a způsobů jak se jim bránit. Těžištěm je přehled v současnosti používaných kryptografických algoritmů a jejich konkrétních implementací v síti internet. Zastoupeny jsou algoritmy symetrické (IDEA, AES) i nesymetrické (Diffie-Hellman, RSA). Zvláštní pozornost je věnována hašovacím algoritmům z rodiny SHA a MD5. Z implementací šifrování práce zmiňuje zabezpečené internetové protokoly SSH, SSL, S-HTTP, IPSec a S/MIME. Z programů jsou pak blíže popsány aplikace PGP Desktop a TrueCrypt. Velký zřetel je kladen na praktické použití elektronického podpisu. Důležitou kapitolou práce je závěrečná případová studie zabezpečení komunikace v mobilní síti GSM.

Klíčová slova

internet, šifrování, kryptologie, kryptografie, kryptoanalýza, algoritmus, zabezpečení, bezpečnost, protokol, PGP, TrueCrypt, Wi-Fi, GSM, GPRS, komunikace, mobilní, certifikační autorita, elektronický podpis, digitální podpis, hacking, riziko, útok

Předmluva

Tématem mojí diplomové práce je *Aplikovaná kryptologie v internetové komunikaci*. Je to téma stále velice aktuální. S rostoucím počtem uživatelů internetu roste i množství přenášených dat, která považujeme za důvěrná. Tato data by se za žádných okolností neměla dostat do rukou nepovolaným osobám. Přesto se tak bohužel v mnoha případech děje. I když existuje celá řada možností jak nežádoucímu úniku dat zabránit, stále jich využívá jen poměrně malá skupina uživatelů. Jednou z možných příčin je malá informovanost o dané problematice.

Práce si klade za cíl přinést přehled možných rizik internetu a způsobů jak se jim bránit. Těžištěm je přehled v současnosti používaných kryptografických algoritmů a jejich konkrétních implementací v síti internet. Nechybí zabezpečení stále populárnějších sítí Wi-Fi a také GSM komunikace, stejně jako příklady vhodného softwaru pro zabezpečení dat. Uživatel tak získá informace o možných rizicích internetové komunikace a způsobech jejich prevence.

Vzhledem k aktuálnosti tématu bylo snadné nalézt relevantní literaturu. Čerpal jsem ze zdrojů v češtině a angličtině. V procesu informační přípravy byla využita databáze INSPEC, katalogy Státní technické knihovny a Městské knihovny Praha. Také byly použity internetové vyhledávače Google a Google Scholar. Cenným zdrojem informací byl elektronicky vydávaný magazín Crypto-World.

Práce je členěna do šesti kapitol. V úvodní z nich se čtenář stručně seznámí s tématem a obsahem diplomové práce. Druhá kapitola přináší přehled rizik využívání počítačové sítě internet. Třetí kapitola se již týká samotné kryptografie. Podává výklad základní terminologie, jsou v ní popsány kryptografické algoritmy a možnosti jejich ohrožení útočníky. Kapitola čtvrtá se zaměřuje na konkrétní implementace kryptografických algoritmů v praxi. Je v ní pojednáno o zabezpečených verzích komunikačních protokolů, elektronickém podpisu, certifikačních autoritách a softwaru podporujícím šifrování dat. Na závěr kapitoly jsou uvedeny praktické ukázky práce s certifikáty, šifrováním a elektronickým podpisem v programech MS Outlook a PGP Desktop, společně s šifrováním dat na disku v programu

TrueCrypt. Pátá kapitola obsahuje případovou studii zabezpečení v GSM komunikaci. Popisuje architekturu mobilní sítě GSM, použité šifrovací algoritmy a zabývá se jejich bezpečností. Závěrečná kapitola přináší shrnutí celé práce.

Celkový rozsah diplomové práce je 113 stran. Citace jsou uvedeny v souladu s normami ISO 690 a ISO 690-2.

Na závěr předmluvy bych rád poděkoval vedoucímu bakalářské práce, panu ing. Martinu Součkovi, za konzultace a rady, jakož i za poskytnuté informační prameny. Dále chci vyjádřit velký dík panu Mgr. Pavlu Vondruškovi z Odboru bezpečnosti společnosti Telefónica O2 Czech Republic, který mi poskytl cenné materiály k vytvoření případové studie.

1 Úvod

Internet je již zcela běžnou součástí našeho života. Mnozí jej používají jako nejčastější způsob komunikace. Jeho prostřednictvím můžeme nejen vyřizovat soukromou korespondenci, být v kontaktu s přáteli díky službám instant messagingu nebo IP telefonie, ale stále častěji v síťovém prostředí také uzavíráme obchody, nakupujeme, platíme, odesíláme faktury nebo daňová přiznání, přistupujeme ke svým bankovním kontům, archivujeme důležitá data atd. Diskutují se i volby přes internet a další možnosti, které bychom ještě donedávna považovali za nereálné. Jak je vidět, s rostoucí oblibou internetu roste počet jeho uživatelů a tím i množství přenesených dat, z nichž mnohá jsou z kategorie důvěrných.

Právě důvěrnost je pro některá data nejdůležitějším aspektem. Lékařské nebo pojišťovací záznamy, platební příkazy, interní firemní materiály, to vše jsou data, která by se za žádných okolností neměla dostat do nepovolaných rukou. Banky, nemocnice, pojišťovny i nejrůznější soukromé firmy by měly vyvíjet snahy o naprosté utajení takových dat.

Dalším důležitým prvkem internetové bezpečnosti je zajištění integrity. Žádný útočník by neměl mít možnost data svévolně měnit či do nich jakýmkoli způsobem zasahovat. Jedná se například o možnost změnit číslo účtu bankovního převodu tak, aby částka odešla na účet nepovolané osoby, místo na původně zamýšlené bankovní konto.

Třetím kritickým elementem síťové bezpečnosti je zachování dostupnosti dat. Ta může být narušena např. jejich smazáním, vyřazením serveru, na kterém se data nacházejí, z provozu atd.

Porušení některého (nebo v horším případě všech) těchto prvků má negativní následky. Od prosté ztráty cenného času, nutného k navrácení systému do původního stavu, po pokles produktivity firmy, ztrátu peněz, pověsti či dokonce lidských životů (např. při narušení systému letecké navigace nebo důležitých medicínských systémů).

Útočníci mohou být různí. Od nudících se studentů, přes zkušené kryptoanalytiky až po vládami či konkurenčními korporacemi najaté agenty, kteří se pokoušejí o průmyslovou

špionáž. Může jít i o propuštěné zaměstnance, jejichž jedinou motivací je pomsta bývalému zaměstnavateli. Útočníci bývají zpravidla technicky nadprůměrně zdatní. Často studují slabiny v systémech, zdrojových kódech aplikací apod. Své poznatky rádi zveřejňují na specializovaných internetových stránkách, diskusních fórech nebo mailing listech. Snahou některých je pouze se něčemu přiučit či získat popularitu ve své komunitě. Jiní mají však čistě kriminální motivy. Používají viry, trojské koně nebo metody sociálního inženýrství, aby získali přístup k cenným datům.

Proti útokům se můžeme bránit celou řadou prostředků. Existují antivirové programy a firewally jako základní ochrana každého počítače připojeného k internetu. Důvěrná data by měla být chráněna prostředky autentizace a autorizace.

Autentizace znamená zjištění totožnosti uživatele služeb internetu. Oprávněný uživatel by tak měl mít ke službám přístup, neoprávněný naopak nesmí do systému proniknout.

Autorizace je ověřením přístupových práv uživatele. V podstatě jde o zjištění, zda má dotyčný uživatel právo vykonat danou činnost (např. administrátoři v systémech zpravidla mohou mazat data, běžný uživatel nikoli).

V neposlední řadě je zde možnost použít kryptologickou ochranu. To je zvláště užitečné u citlivých informací, kam patří přihlašovací údaje do systémů, čísla kreditních karet, medicínské a jiné důvěrné informace atd. Kvalitní šifrování zajišťuje důvěrnost, integritu i autentizaci. Existuje celá řada nástrojů implementujících šifrovací algoritmy. Ať už se jedná o zabezpečené verze internetových protokolů či aplikace dodávající externí možnost šifrování komunikace. Právě tomuto způsobu ochrany se věnuje tato diplomová práce.

Práce je členěna do šesti kapitol. První kapitola obsahuje úvod do problematiky a vymezuje zaměření celé práce.

Ve druhé kapitole jsou stručně popsána rizika, která hrozí každému uživateli sítě internet. Je v ní pojednáno o útočnících i typech útoků. Hrozby nesouvisející s kryptologií jsou jen letmo zmíněny, neboť není možné se jimi v omezeném rozsahu této práce zabývat detailněji.

Třetí kapitola už nabízí pohled na samotnou kryptologickou ochranu. V jejím úvodu je předestřena základní terminologie. Práce se následně zabývá rozdílem mezi kryptografií a kryptoanalýzou, proudovými a blokovými šiframi apod. Dále je uvedena typologie útoků na kryptografické algoritmy. Hlavním obsahem kapitoly je pak důkladný popis těchto algoritmů.

Čtvrtá kapitola se věnuje implementacím kryptologické ochrany. Pojednává o zabezpečených verzích internetových protokolů, způsobech ochrany elektronické pošty, digitálním podpisu a také o programech PGP a TrueCrypt. V závěru kapitoly se nachází popis průběhu registrace u certifikační autority CA Czechia a následné využití získaného certifikátu při šifrování a digitálním podepisování elektronické pošty.

Pátá kapitola obsahuje případovou studii, která se zaměřuje na široce rozšířenou a populární síť GSM (respektive GPRS). Mobilní komunikace je dnes využívána nejen k přenosu hlasových či textových a obrazových zpráv, ale stále častěji i k přístupu na internet a s ním i k elektronickému bankovnictví a dalším službám, vyžadujícím důvěrnost. Případová studie podává stručnou historii systému GSM, popisuje architekturu mobilní sítě, její funkce a provoz, zabezpečení a implementované šifrovací algoritmy. Na závěr se práce věnuje velice aktuálnímu tématu prolomení šifrovacích algoritmů A5/1 a A5/2.

Poslední, šestá, kapitola přináší shrnutí a závěr celé práce.

2 Rizika Internetu

2.1 Programové ohrožení

Prostředí sítě internet bylo původně navrženo jako nezabezpečené. Síť využívali hlavně akademici ke vzájemné komunikaci vědeckých poznatků. V okamžiku, kdy se internet stal veřejnou sítí, začal rapidně stoupat počet uživatelů. Ne každý má však dobré úmysly. Tak vznikly různé druhy ohrožení bezpečnosti dat v počítačových systémech.

K nejznámějším patří [6]:

1) Viry
2) Červi
3) Trójské koně
4) Logické bomby
5) Bakterie
6) Spyware

„Viry jsou programy, které modifikují jiné programy tím, že do nich vkládají své kopie[1]“. Mohou se projevovat neškodnými hláškami či obrázky vypisovanými na monitor, ale v horším případě také mazáním dat na disku atd.

Červi jsou krátké programy, které se samovolně šíří Internetem. Například v elektronické poště.

Trójské koně pracují skrytě na pozadí jiných aplikací. Slouží k vytvoření tzv. *zadních vrátek* do systému, která může případný útočník využít k jeho napadení.

Logické bomby bývají neškodně uloženy v počítači až do chvíle, kdy nastane situace, na kterou mají reagovat. Tou může být třeba přihlášení určitého uživatele, vykonání jisté akce

[1] [6] s. 328

apod. Poté bomba spustí a provede akci, ke které je naprogramována. Může například smazat všechna data na uživatelově disku nebo odeslat důvěrná data nepovolané osobě, která bombu vytvořila.

Bakterie jsou zvláštním druhem škodlivých programů. Podobně jako jejich vzor v organickém světě se dokáží samovolně reprodukovat, čímž způsobí zahlcení systému, ve kterém se nacházejí.

Spyware se snaží být nenápadný. Pracuje skrytě a na pozadí běhu jiných aplikací odesílá citlivá data o uživatelově systému, jeho návycích při prohlížení internetu apod. Bývá používán například pro cílené zasílání nevyžádané reklamy, tzv. spamu.

Povědomí o těchto hrozbách je v současnosti již docela vysoké. Existuje proti nim spolehlivá ochrana ve formě antivirových programů, firewallů a dodržování bezpečnostních zásad pohybu po internetu. Není záměrem této práce se výše zmíněnými hrozbami hlouběji zaobírat. Místo toho se práce dále zaměří na ochranu dat před kompromitací nežádoucí osobou.

2.2 Útočníci

V literatuře [5] bývají obvykle osoby, které napadají systémy uživatelů za účelem manipulace s jejich důvěrnými daty, děleny na dvě základní skupiny:

Hacker

Slovo hacker původně označovalo člověka, který se sám pokoušel upravit programy tak, aby splňovaly jeho představy o funkčnosti [5]. Pokud program něco neuměl, hacker si chybějící funkci jednoduše naprogramoval. Později se termín posunul. Dnes literatura nejčastěji uvádí hackera jako člověka s výbornými znalostmi systémů a programování, který se „nabourává“ do počítačových systémů. Nečiní tak ale ze ziskuchtivosti či touhy škodit, ale většinou jde o touhu dokázat si svoje schopnosti, odhalit slabiny nějakého systému nebo se předvést před ostatními hackery.

Cracker

Oproti hackerovi se jedná o člověka s kriminálními sklony [19]. Snaží se nelegálně obohatit nebo škodit společnostem například mazáním dat, zásahy do dokumentů atd.

Jiný výklad tohoto termínu podává crackera jako člověka, který se snaží obcházet ochrany počítačových programů tak, aby mohly být dále pirátsky šířeny. Cracker z toho nutně nemusí mít prospěch a může tak činit pouze z touhy něčemu se přiučit nebo se ukázat ve své komunitě jako úspěšný [19].

Jiné dělení útočníků uvádí [6]:

- Zaměstnanci (Propuštění, poškození, uražení, kteří znají slabiny systému a chtějí se pomstít.)
- Zloději (Jejich cílem je zisk nebo snaha maskovat jiné aktivity.)
- Špióni (Průmyslová nebo politická špionáž, sabotáže. Snaha o získání tajných informací, vyřazení konkurence.)
- Vyděrači (Hledají bezpečnostní slabiny v systémech, aby je pak za úplatu oznámili správci. Nejsou-li jejich podmínky splněny – nedostanou-li zaplaceno, zveřejní odhalené slabiny na internetu, čímž vystaví systém riziku útoků a firmu nebezpečí ztráty cenných dat a prestiže.)
- Experimentátoři (Do systému pronikají ze zvědavosti.)
- Lovci publicity (Jde jim o ukojení vlastního ega, slávu nebo zisk.)

- Političtí aktivisté (Prezentace názorů, v horším případě rozklad systému vládní či velké komerční organizace. Čím větší je napadený systém, tím větší má akce politický dopad.)

Pro tuto práci však není důležité, za jakým účelem se nepovolaná osoba pokouší uživatelova důvěrná data získat. Ať již jde o motiv čistě ze zvědavosti nebo naopak dopředu naplánovaný kriminální čin, vždy je třeba ochránit data uživatelů před padnutím do nepovolaných rukou. Nikomu není lhostejné, pokud někdo cizí může jeho data číst nebo dokonce modifikovat, i když by šlo jen o „neškodnou“ zábavu. Z tohoto důvodu nebude v této práci rozlišováno zda se v případě napadení systému jedná o hackera či crackera, ale vždy se bude jednat pouze o „útočníka“.

2.3 Formy útoků

Existuje několik hlavních způsobů, kterými se útočník může pokusit získat citlivá data, jako jsou např. přihlašovací jména a hesla uživatelů, vyřadit z provozu server nebo jinak škodit. Patří sem [4]:

Denial of Service / Distributed Denial of Service (DoS / DDoS)

Tzv. útok odepřením služby. Útočník odešle na cílový server velký objem dat, který příjemce zahltí a server tím dočasně vyřadí z provozu. Protože je prakticky nemožné provést takovýto úspěšný útok z jednoho jediného počítače, používá se většinou jeho vylepšená varianta DDoS – distribuovaný útok odepřením služby. Při něm útok probíhá z velkého množství počítačů naráz. Obvykle je do nich napřed pomocí nějakého červa potají nahrána logická bomba, která pak v předem určený čas spustí a všechny napadené počítače podniknou útok DoS na cílový server. Často se takto útočí na servery známých mezinárodních organizací, sdělovacích prostředků nebo velkých softwarových firem jako je např. Microsoft.

Útok opakováním

Útočník odposlechne část komunikace mezi dvěma stranami a získané informace použije při útoku na systém.

Útok ze středu (zrcadlový útok)

Útočník „sedí“ mezi dvěma komunikujícími stranami a kontroluje komunikační kanál. Pro stranu A přitom vystupuje, jako by byl stranou B a opačně (viz. obr. 1).

obr. 1 – Schéma útoku ze středu (Zdroj: [4])

Útok na hesla

Snaha o průnik do systému pomocí tzv. *útoku hrubou silou*. Útočník použije program, který do systému zkouší zadávat stovky hesel, dokud nenalezne to správné. Lepší variantou je tzv. *slovníkový útok*, kdy program obsahuje seznam slov, která se nejčastěji jako hesla používají (mohou to být slova přirozeného jazyka – např. angličtiny, češtiny atd.). Druhý jmenovaný útok počítá s častou chybou uživatelů. Ti si ve velkém počtu případů volí jako heslo nějaké běžné slovo nebo jméno.

Útok na integritu zprávy

Při tomto útoku je zpráva zachycena během přesunu po síti a změněna ve prospěch útočníka (např. falešný příkaz k úhradě peněz) nebo tak, aby ji příjemce nemohl přečíst.

Odposlech

Útočník pouze zachycuje a analyzuje data během jejich přesunu mezi dvěma komunikujícími stranami. Snaží se zachytit citlivé informace, které by mohl zneužít ve svůj prospěch.

3 Kryptologická ochrana

Již od starověku lidé utajovali důvěrné zprávy pomocí šifer. Kryptografie byla poprvé použita před 4000 lety Egypťany [21]. Zatímco dříve šifrování používaly hlavně vlády, armáda a diplomaté, dnes, s nástupem počítačů, je dostupné i soukromým osobám. Například v podobě programu PGP pro zabezpečení elektronické pošty. V úvodu této kapitoly se práce zaměří na úvod do problematiky kryptologie, vysvětlení základních pojmů a rozdílů mezi proudovými a blokovými šiframi, symetrickou a asymetrickou kryptografií. Dále budou popsány jednotlivé v současnosti používané kryptografické a hašovací algoritmy. Zmíněny budou i formy útoků na ně.

3.1 Terminologie

Pokud se chceme detailněji seznámit s fungováním kryptografických algoritmů, musíme nejprve uvést definice základních pojmů, které se v této oblasti používají.

Kryptologie

„Kryptologie je vědní obor zabývající se tvorbou, používáním a luštěním čili prolamováním šifer. Zahrnuje dvě odvětví – kryptografii a kryptoanalýzu.“[2]

Jiná definice uvádí:

„Kryptologie je matematický vědní obor, který se zabývá šifrovacími a kódovacími algoritmy. Rozpadá se na dvě velké skupiny – kryptografii, která se zabývá návrhem šifrovacích algoritmů, a kryptoanalýzu, která se naopak snaží šifrovací algoritmy prolomit.“[3]

[2] [19] s. 191
[3] [4] s. 23

Kryptografie

„Kryptografie je věda o tvorbě šifer, kdy informace mají abecedně číslicový charakter. Kryptografie studuje šifrovací algoritmy, kryptografické nástroje, hardwarové implementace šifrovacích algoritmů, kryptografické protokoly apod.“[4]

K cílům kryptografie patří zajištění důvěrnosti dat, datové integrity, autentizace a nepopiratelnosti (neodmítnutelnosti) [21].

- Důvěrnost je ochrana dat před neautorizovanými osobami. Jde o ochranu skrze utajení použitím kryptografických algoritmů.
- Integrita dat je znemožnění jakýchkoli zásahů do struktury dat nepovolanou osobou. Jedná se o zákaz data odstraňovat, vkládat nebo jakkoli zaměňovat.
- Autentizace se týká jak účastníků komunikace, tak samotné zprávy mezi nimi přenášené. Jak komunikující strany, tak zpráva by měly být jednoznačně identifikovatelné, aby se za ně nemohl vydávat někdo jiný.
- Nepopiratelnost zabraňuje komunikujícím, aby popřeli své dřívější akce. Jedna strana např. může autorizovat obchod s druhou stranou a později tuto operaci popřít. Nepopiratelnost je tedy ochranou proti podvodům s identifikací [21].

Kryptoanalýza

„Kryptoanalýza je věda o luštění šifer. Jiná definice kryptoanalýzy uvádí, že je to umění, věda či útok na zašifrovaný text a jeho převedení do původního textu bez znalosti klíčů.“[5]

Šifrovací algoritmus

[4] [19] s. 191
[5] [19] s. 192

„Šifrovací algoritmus je takový algoritmus, který se snaží utajit data jejich zašifrováním tak, že je pak nelze přečíst a jsou nesrozumitelná.“[6]

„Šifrovací algoritmus je algoritmus, který se snaží utajit data jejich zašifrováním. K této činnosti používá nějaké tajemství, většinou ho nazýváme šifrovacím klíčem. Více lidí tak může používat jeden šifrovací algoritmus, každý ovšem musí mít k šifrování a dešifrování dat stejný klíč.“[7]

Kódovací algoritmus

Kódovací algoritmus se také snaží utajit data tím, že je učiní nesrozumitelnými. Nepoužívá ovšem žádný šifrovací klíč, jako je tomu u šifrovacího algoritmu. Zakódovaná data mohou číst jen lidé vlastnící stejný algoritmus. Typickým příkladem je překlad textu do jiného jazyka nebo vytvoření vlastních kódových slovníků [4].

Šifrování

„Šifrování je proces, při němž konkrétní kryptografická metoda přemění, transformuje otevřený text pomocí kryptografického algoritmu a šifrovacího klíče do šifrovaného textu, který brání a znemožňuje každému, kdo nemá k dispozici jistý utajovaný údaj, tyto informace získat.“[8]

Prolomení algoritmu

„Algoritmus je prolomen, pokud je možné číst chráněná data bez znalosti šifrovacího klíče nebo kódovacího algoritmu.“[9]

Otevřený text (OT)

[6] [19] s. 191
[7] [4] s. 23
[8] [19] s. 192
[9] [4] s. 24

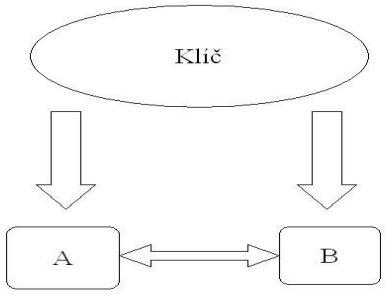
Text v nezašifrované podobě. Vyskytuje se tak před zašifrováním nebo po opětovném dešifrování [4].

Šifrovaný text (ŠT)

Zašifrovaný nebo zakódovaný text, který nelze přečíst bez znalosti klíče nebo šifrovacího algoritmu [4].

Symetrická kryptografie

Odesílatel i příjemce musejí vlastnit stejný šifrovací klíč. Odesílatel nejprve zašifruje zprávu za pomoci klíče a posléze ji odešle. Příjemce zprávu dešifruje stejným klíčem, tak získá opět původní text (viz. obr. 2). Problémem je bezpečné doručení klíče druhé straně, aby se během přenosu nedostal do rukou nepovolané osoby.



obr. 2 – Schéma symetrické kryptografie

Největší nevýhodou je však správa klíčů. Pro dvě komunikující strany postačí jeden klíč. Se vzrůstajícím počtem osob však roste neúměrně i počet klíčů.

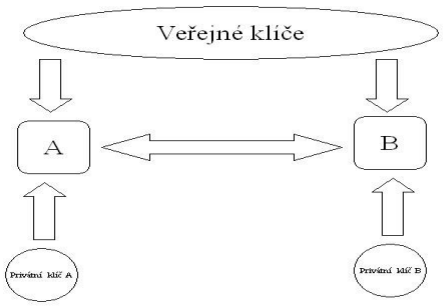
„Pro N klíčů je třeba: N = n (n-1) / 2, kde n je počet osob.“[10]

[10] [19] s. 202

Naopak výhodou je vysoká rychlost a nepříliš velká náročnost na výpočetní výkon.[19]

Asymetrická kryptografie

Též označována termínem *kryptografie veřejného klíče* (viz. obr. 3). Při tomto způsobu ochrany dat se používá tzv. klíčový pár, tvořený privátním (soukromým) klíčem (PK) a klíčem veřejným (VK). Odesílatel nejprve zašifruje text pomocí veřejného klíče příjemce. Ten obdrženou zprávu dešifruje pomocí svého soukromého klíče, čímž získá původní otevřený text. Veřejný klíč má k dispozici každý, kdo chce s příjemcem komunikovat, ovšem dešifrovat text může jen vlastník příslušného privátního klíče, tedy oprávněný příjemce. Tím je vyřešen problém správy klíčů, který je nevýhodou symetrické kryptografie. Nejen, že je pro komunikaci libovolného počtu osob třeba menší počet klíčů (každá osoba má jen jeden pár klíčů), ale odpadá i problém bezpečného doručení klíče druhé straně. [19]



obr. 3 – Princip asymetrické kryptografie

Nevýhodou je až 1000x nižší rychlost algoritmu oproti symetrické kryptografii. Další nevýhodou je nutnost ověření pravosti klíče, tj. stoprocentní identifikace majitele VK. O to se starají tzv. *certifikační autority* (viz. kapitola 4 diplomové práce), které vlastní databáze ověřených osob a jejich VK.[11]

[11] [19] s. 204

Proudové šifry

Otevřený text je šifrován po jednotlivých znacích. Celá šifra sestává z generátoru náhodných čísel, který vytváří výstup na základě hodnot symetrického klíče. Výstup je pak funkcí XOR kombinován s otevřeným textem. Dešifrování probíhá obdobně.[12]

Blokové šifry

Text je šifrován ve větších celcích, tzv. blocích. Velikost bloku vstupujícího do procesu šifrování je stejná jako velikost výstupního bloku a musí být volena tak, aby se z pouhé analýzy vstupu a jemu odpovídajícího výstupu nedala šifra rozluštit.

Dochází k překrývání obou pojmů, neboť znakem nemusí být nutně jeden bit, ale častěji celé skupiny bitů (např. 64b aj.). Délka bloku naopak může dosáhnout pouze jediného znaku (bitu) [4].

Módy blokových šifer

Literatura [3, 4] uvádí čtyři hlavní způsoby zapojení blokových šifer:

Electronic Code Book (ECB)

Vstupnímu bloku OT je jednoduše přiřazen stejně dlouhý blok ŠT. Stejnému vstupu je pokaždé přiřazen stejný výstup. Z toho plyne velké nebezpečí prolomení algoritmu.

Cipher Block Chaining (CBC)

Spočívá v zavedení zpětné vazby do systému. První vstupní blok je po zašifrování přiveden zpět na vstup, kde je logickou funkcí XOR zkombinován s druhým vstupním blokem. Teprve takto XORovaný blok vstupuje do šifrovacího algoritmu. Nevýhodou je šifrování prvního bloku v režimu ECB, čímž může dojít k ohrožení bezpečnosti, protože

[12] [19] s. 200

začátky zpráv bývají zpravidla stejné, čehož může útočník využít. Proto se i první blok XORuje. K tomu se využívá tzv. *inicializační vektor* (např. časové razítko). Vektor se často připojuje ke zprávě, neboť jej musí znát i příjemce.

Cipher Feedback Mode (CFB)

Využívá generátoru náhodných čísel, který se řídí šifrovaným textem z výstupu. Náhodná čísla jsou v šifrovacím zařízení aplikována na OT.

Output Feedback Mode (OFB)

Zde je zpětná vazba zapojena z výstupu generátoru náhodných čísel a znovu přivedena do generátoru.

Encryption-Decryption-Encryption (EDE)

Tento mód je specifický zapojením třech šifrovacích bloků v sérii. První blok zašifruje OT klíčem č. 1, druhý blok jej dešifruje druhým klíčem a třetí blok opět zašifruje klíčem č. 1. EDE se používá k prodloužení klíče.

3.2 Útoky na kryptografické algoritmy

Pojmem útok rozumíme pokus o vyluštění šifry. Cílem je získání OT nebo šifrovacího klíče. Protože všechny v současnosti používané algoritmy jsou veřejně známé, je jasné, že útočníci mají dostatek potřebných informací o algoritmu, který se pokoušejí rozbít, plně k dispozici [4]. Nyní si tedy ukážeme metody možných útoků.

Útok hrubou silou (Brute Force Attack)

Pokus o vyluštění šifrovaného textu vyzkoušením všech možných hesel. K takovémuto útoku je třeba velice výkonný výpočetní systém, který postupně projde celý prostor klíčů, až najde ten správný. Ověření probíhá formou kontrolního součtu nebo srovnáním vyluštěného

slova se slovníkem nejčastějších slov, která se vyskytují ve zprávách. Jistou variantou tohoto způsobu luštění je tzv. *slovníkový útok*. Při něm útočník využívá rozsáhlý slovník slov v různých jazycích, která se nejčastěji používají jako hesla. Tento typ útoku se často zaměřuje na uživatelské a správcovské účty v systémech firem, bank, státních organizací atd.

Luštění se znalostí šifrovaného textu (Ciphertext-only Attack)

V případě, že útočník odposlechne větší množství textů šifrovaných stejným klíčem, může jejich analýzou najít klíč, kterým pak dešifruje zprávy a získá původní otevřený text.

Luštění se znalostí vybraných otevřených textů (Chosen-plaintext Attack)

V tomto případě má útočník možnost zvolit si otevřené texty, které zašifruje. Tím má velkou možnost snadno analyzovat vstupní a k nim odpovídající výstupní texty a získat tak klíč.

Luštění se znalostí vybraných šifrových textů (Chosen-ciphertext Attack)

Kryptoanalytik volí, které texty chce dešifrovat. Tím získá OT. Zkoumáním vztahů mezi ŠT a OT může zjistit klíč. Používá se např. v případě, že se podaří zajistit dešifrovací stroj, ale není k němu k dispozici stroj šifrovací.

Pendreková metoda (Rubberhose Cryptoanalysis; též korupční metoda)

Tato metoda je cílená přímo na konkrétní osoby. Aby útočník získal klíč, snaží se přesvědčit některého z uživatelů. Může přitom použít úplatku i hrubého násilí, vydírání apod.

Problém útoků vedených v počítačových sítích je velice obsáhlý a není možné jej v omezeném rozsahu této práce rozebírat do hloubky. Předcházející část tedy přinesla pouze stručný přehled toho nejdůležitějšího[13]. Další konkrétní útoky (např. na hašovací algoritmy) budou popsány v příslušných kapitolách.

[13] Popis útoků vytvořen podle [4]

3.3. Kryptografické algoritmy

Následující podkapitola přináší přehled v praxi nejpoužívanějších algoritmů symetrických i asymetrických.

3.3.1 Symetrické algoritmy

Hodnotné informace o symetrických algoritmech přináší [20]:

DES

Zkratka tohoto algoritmu znamená Data Encryption Standard. Americký normalizační institut ANSI pro něj používá též označení DEA (Data Encryption Algorithm). Jedná se o symetrickou blokovou šifru tvořenou 64 bitovými bloky. Využívá 56 bitový klíč. DES byl do konce 90. let 20. století americkým kryptografickým standardem. Dnes tomu tak není, ale je dobré algoritmus popsat a zdůvodnit, kde se nacházejí jeho slabiny.

Základem DESu jsou dvě po sobě jdoucí operace – substituce a permutace. Ty jsou společně s klíčem použity k zašifrování otevřeného textu. Jeden takový průběh algoritmu se nazývá runda. Během šifrování vykoná DES plných 16 rund.

„V každé rundě se bity klíče posunou a pak z 56 bitů klíče se vybere 48 bitů. Pravá polovina dat se rozšíří expanzní permutací na 48 bitů, zkombinuje s 48 bity posunutého a permutovaného 48 bitového klíče ve sčítačce mod-2, zpracuje 8 S-boxy na 32 nových bitů a znovu permutuje. Tyto čtyři operace představují funkci *f*. Výstup funkce *f* se potom v další sčítačce mod-2 zkombinuje s levou polovinou. Výsledek těchto operací se stává novou pravou polovinou; stará pravá polovina se stává novou levou polovinou. Tyto operace se opakují 16krát a vytvářejí 16 rund DESu.

Bude-li B_i výsledkem i-té iterace, L_i a *Ri* levou a pravou polovinou B_i, K_i 48-bitovým klíčem pro i-tou rundu a funkcí provádějící veškeré substituce, permutace a operace mod-2 s účastí klíče, pak runda bude vyjádřena takto:

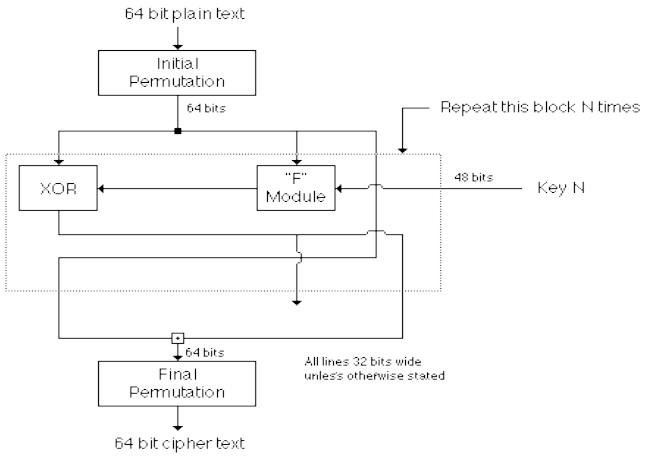
$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$ “[14]

Na vstupu máme 64 bitové bloky otevřeného textu. Algoritmus zahájí jejich počáteční permutací. Po ní jsou bloky rozděleny na pravou a levou polovinu. S těmi se provede 16 rund substituce a permutace za použití klíče. Tím dosáhneme promíchání dat s klíčem. Po dokončení poslední rundy se levé a pravé poloviny opět spojí. Na závěr dojde na konečnou permutaci. Tím je šifrování u konce.

K dešifrování se používá naprosto stejná funkce. Jediným rozdílem je, že šifrovací klíče je nutné aplikovat v opačném pořadí, abychom získali zpět otevřený text.

DES existuje i v bezpečnější rozšířené variantě - tzv. trojnásobném DESu (Triple DES, 3DES). Algoritmus je na šifrovaný text použit celkem třikrát za použití až trojnásobně dlouhého klíče. Při prvním a třetím běhu se text zašifruje prvním klíčem, prostředním krokem se dešifruje použitím druhého klíče [20].



obr. 4 – schéma algoritmu DES (zdroj: [24])

[14] [20] str. 124

Kritika a prolomení DESu

DES byl uznán za americký národní šifrovací standard roku 1976 přes to, že vůči němu byly vznášeny námitky. Týkaly se především délky jeho klíče (56 bitů). NSA argumentoval tím, že stroj natolik rychlý, aby otestoval všechny klíče v rozumném čase, by stál nerozumně velkou sumu peněz.[15] O dvacet let později se však námitky ukázaly jako oprávněné.

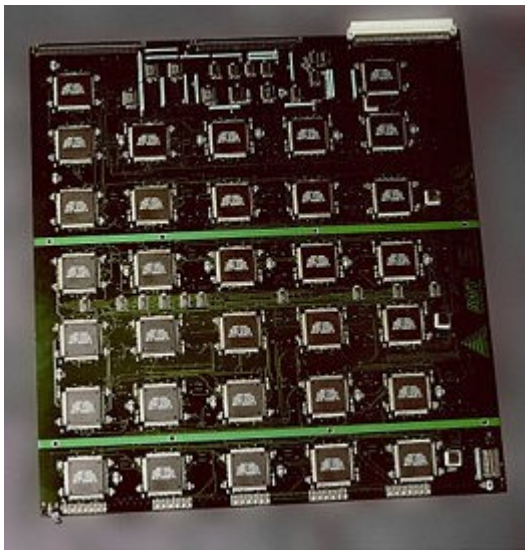
Od počátku schválení DESu za standard se v akademickém prostředí objevovaly návrhy strojů na jeho prolomení. Už rok od jeho uvedení se Diffie s Hellmanem pokusili navrhnout takový stroj, který by dokázal rozlomit šifru za jediný den. Podle jejich odhadu by se však výsledná cena vyšplhala na 20 milionů dolarů.

V roce 1993 Wiener navrhl jiný stroj, který by údajně dokázal DES prolomit již za 7 hodin. Náklady již byly o poznání nižší – 1 milion dolarů. Ani jeden z těchto návrhů však nebyl realizován (nebo to aspoň není veřejně známo).

V roce 1998 byla společností RSA Security vyhlášena soutěž na prolomení šifry DES. Nadace Electronic Frontier Foundation u její příležitosti postavila stroj Deep Crack, který stál necelých 250 000 dolarů.

Na konstrukci Deep Cracku se kromě EFF podílely společnosti Cryptography Research Inc. a Advanced Wireless Technologies. Hlavním designérem byl prezident Cryptography Research – pan Paul Kocher. Deep Crack sestával z 1856 čipů na 29 deskách po 64 čipech na každé desce (viz. obr. 5). Desky byly umístěny v šesti skříních, přičemž koordinaci celého procesu hledání klíče řídil pouze jeden počítač, který přiděloval rozsahy klíčů k prohledávání jednotlivým čipům. Tak byl stroj schopen otestovat více než 90 miliard klíčů za sekundu. Otestovat všechny klíče z rozsahu DESu by tímto tempem zabralo asi 9 dní. Průměrná doba nalezení správného klíče je ovšem poloviční.

[15] [27]



obr. 5 - několik čipů na jedné z desek stroje Deep Crack (zdroj: [28])

S pomocí Deep Cracku se podařilo rozluštit soutěžní šifru za 56 hodin. EFF tím dokázala, že pro bohaté korporace nebo vlády v současnosti není problém si opatřit stroj, který DES v relativně krátké době dokáže prolomit.

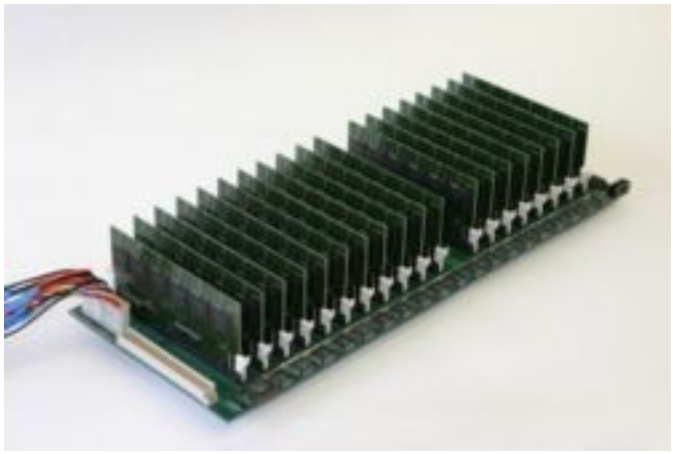
Za půl roku po této události byla RSA Security vyhlášena další soutěž se stejným cílem. EFF se rozhodlo ke spolupráci se sítí distributed.net. Složený výpočetní výkon Deep Cracku a této sítě znamenal rozluštění další zprávy za dobu kratší, než jeden den (přesně 22 hodin a 15 minut[16]). Ještě téhož roku (1999) byl DES opět potvrzen jako americký národní standard. Tentokrát ovšem ve vylepšené verzi 3DES[17].

Nebezpečně malá velikost klíče DESu společně s relativně vysokou výpočetní náročností 3DESu vyústily v jeho nahrazení novou šifrou AES. Ta vstoupila v platnost jako americký standard 26. května 2002.

Jediným dalším potvrzeným strojem na prolomení DESu byla COPACOBANA (zkratka z cost-optimized parallel code breaker), postavená týmy z německých univerzit v Bochumi a Kielu. Tento stroj nebyl, na rozdíl od Deep Cracku společnosti EFF, sestaven z žádných speciálních součástek, ale pouze z komerčně dostupných rekonfigurovatelných integrovaných obvodů (viz. obr. 6). Právě rekonfigurovatelnost umožňuje nasazení stroje i na lámání jiných

[16] [27]
[17] [27]

šifer, než je DES. Nejzajímavějším aspektem je velice nízká cena. Jeden stroj bylo možno postavit přibližně za 10 000 dolarů[18].



obr. 6 – COPACOBANA (zdroj: [28])

AES

Advanced Encryption Standard (AES) je bloková šifra, která byla přijata po náročném výběrovém řízení na nový americký kryptografický standard. Otevřenost celého procesu napomohla zvýšení důvěry v bezpečnost nového algoritmu a získala si příznivé ohlasy napříč celou kryptografickou komunitou[19].

V roce 1997 NIST oznámil, že si přeje vybrat nástupce algoritmu DES, který bude nazván American Encryption Standard. Všechny algoritmy přihlášené do soutěže, měly být blokovými šiframi, podporujícími velikost bloku 128 bitů a velikosti klíče 128, 192 a 256 bitů. Tehdy byly podobné šifry relativní vzácností. Nejznámější byl patrně Square[20].

[18] [28]
[19] [29]
[20] [29]

Postupně bylo do soutěže přihlášeno 15 algoritmů:

CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, a Twofish.

Kandidátské šifry byly podrobeny mnoha testům z nejrůznějších hledisek. Hodnocena byla samozřejmě bezpečnost, ale také například výkon v podmínkách různých implementací (na osobních počítačích s rozdílnými architekturami, ve smart kartách, hardwarové implementace atd.) nebo aplikovatelnost v limitovaných prostředích (smart karty s velmi limitovanou pamětí).

V srpnu 1999 NIST oznámil, že do užšího finále postoupila třetina uchazečů: MARS, RC6, Rijndael, Serpent a Twofish. Všech pět algoritmů, obecně známých jako *finalisté AES*, bylo navrženo dobře známými a respektovanými kryptografy[21].

Po dalších zevrubných analýzách byl v dubnu roku 2000 NISTem vybrán vítěz. Stal se jím algoritmus Rijndael, belgických kryptografů Joana Daemena a Vincenta Rijmena.

Popis šifry[22]

Jako vítěz soutěže přijal Rijndael nový název AES. Americkým standardem se stal 26. května 2002. V současnosti je jedním z nejpopulárnějších algoritmů symetrické kryptografie.[23]

Pokud bychom měli hovořit exaktně, není AES úplně to samé, jako Rijndael. Rijndael totiž podporuje větší rozsah bloku a velikosti klíče (může využít jakoukoli velikost bloku a klíče, která je násobkem čísla 32; minimum 128 a maximum 256 bitů). AES má proti němu fixní blok o velikosti 128 bitů a velikost klíče 128, 192 a 256 bitů. AES je rychlý v softwarovém i hardwarovém provedení, je relativně snadný na implementaci a paměťově nenáročný[24].

[21] [29]
[22] podle [30]
[23] [30]
[24] [30]

[30] popisuje funkci algoritmu následovně:

Vzhledem k fixní velikosti bloku - 128 bitů operuje AES na poli bytů o rozměrech 4x4, nazývaném *stav* (verze Rijndaelu s větší velikostí bloku mají ve stavu dodatečné sloupce navíc). Většina výpočtů v AES je prováděna ve speciálním konečném poli.

Vysokoúrovňový šifrovací algoritmus:

- rozšíření klíče za použití tabulky klíčů Rijndaelu
- iniciační runda
  - algoritmus AddRoundKey (přidání klíče rundy)
- rundy

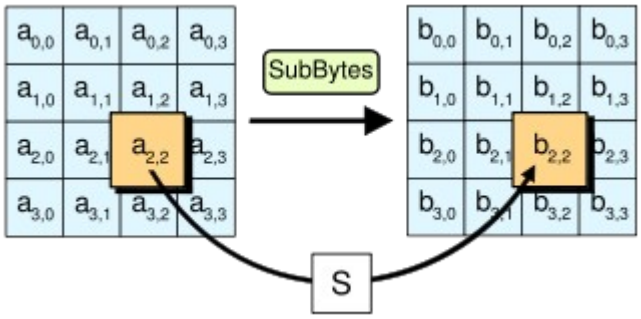
1. sub byty (SubBytes) – nelineární substituce, kde každý byte je nahrazen jiným podle tzv. vyhledávací tabulky
2. prohození řádků (ShiftRows) – transpoziční krok, ve kterém je každá řádka stavu cyklicky posunuta o daný počet kroků
3. promíchání sloupců (MixColumns) – operace, která pracuje se sloupci stavu, kombinuje 4 byty v každém sloupci
4. přidání klíče rundy (AddRoundKey) – každý byte stavu je kombinován s klíčem rundy, každý klíč rundy je získán z šifrovacího klíče za použití tabulky klíčů

- závěrečná runda (bez míchání sloupců)

1. sub byty
2. prohození řádků
3. přidání klíče rundy

Algoritmus SubBytes:

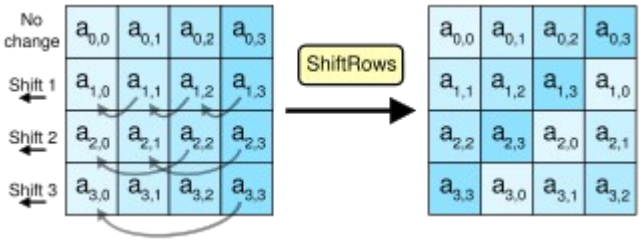
Každý byte v poli je aktualizován za použití osmibitového substitučního boxu – S-boxu Rijndaelu. Tato operace přidává do šifry potřebnou nelinearitu. Použitý S-box je odvozen z převrácené hodnoty nad konečným polem (2^8), známé pro své dobré nelineární vlastnosti. Aby se zamezilo útokům založeným na čistě algebraických vlastnostech, je S-box vytvořen kombinováním inverzní funkce s invertovatelnou afinitou. S-box je také navržen tak, aby zamezil jakýmkoli fixním bodům (a proto je dismutací) a také jakýmkoli protilehlým fixním bodům (viz. obr. 7).



obr. 7 – schéma algoritmu SubBytes (zdroj: [30])

Algoritmus ShiftRows:

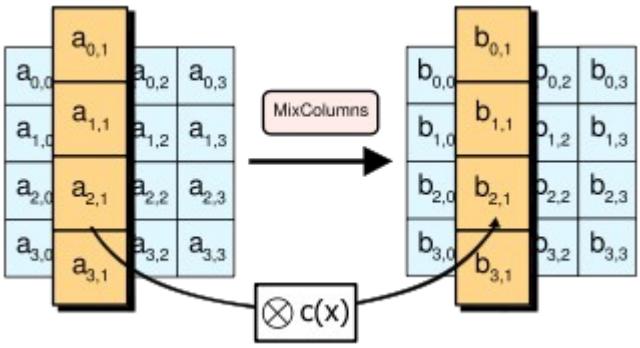
Operuje s řádky stavu. Cyklicky prohazuje byty v každém řádku jistým offsetem. U AES je první řádek ponechán nezměněn. Každý byte z druhé řady je posunut o jednu pozici doleva. Podobně jsou posunuty byty ve třetím a čtvrtém řádku o tři, respektive čtyři pozice doleva. Pro velikosti bloku 128 a 192 bitů je vzor, podle kterého se byty prohazují, stejný. Každý sloupec výstupního stavu algoritmu ShiftRows je tak složen z bytů z každého sloupce vstupního stavu (viz. obr. 8).



obr. 8 – schéma algoritmu ShiftRows (zdroj: [30])

Algoritmus MixColumns:

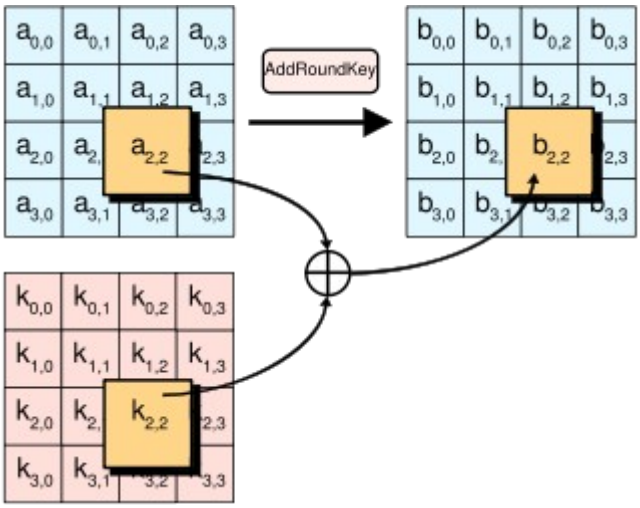
Čtyři byty z každého sloupce stavu jsou kombinovány za použití invertovatelné lineární transformace (viz obr. 9). Funkce MixColumns má na vstupu čtyři byty a na výstupu čtyři byty, kde každý vstupní byte ovlivňuje všechny čtyři byty výstupní. MixColumns a ShiftRows přinášejí do šifry potřebnou difúzi. S každým sloupcem se zachází jako s mnohočlenem nad konečným polem a je následně násoben modulo $x^4 + 1$ s fixním mnohočlenem $c(x) = 3x^3 + x^2 + x + 2$.



obr. 9 – schéma algoritmu MixColumns (zdroj: [30])

Algoritmus AddRoundKey:

Subklíč je kombinován se stavem. Pro každou rundu je odvozen nový subklíč podle tabulky klíčů Rijndaelu. Každý subklíč má stejnou velikost jako stav. Subklíč je přidán kombinováním každého bytu stavu s odpovídajícím bytem subklíče použitím bitové funkce XOR (viz obr. 10).



obr. 10 – schéma algoritmu AddRoundKey (zdroj: [31])

Bezpečnost[25]

Národní bezpečnostní agentura Spojených států amerických (NSA) prověřila všechny finalisty soutěže o nový americký kryptografický standard a prohlásila, že všechny algoritmy jsou dostatečně bezpečné, aby je americká vláda mohla používat pro neutajovaná data. V červnu 2003 vláda USA oznámila, že by měl AES být používán pro utajované informace. Pro materiály označené jako *tajné* je možné používat všechny délky AESu. *Přísně tajné* dokumenty by měly být chráněny klíčem o délce 192 nebo 256 bitů. Zajímavostí je, že poprvé v historii má veřejnost přístup k šifře, uznané agenturou NSA pro šifrování přísně tajných informací.

Objevily se námitky, že poměrně jednoduchá matematická struktura algoritmu představuje bezpečnostní riziko. V roce 2002 oznámili Courtois a Pieprzyk teoretický útok, který nazvali „XSL Attack“. Ten ukázal možnou slabinu AESu. Jeho tvůrci prohlašovali, že je schopný prolomit AES rychleji, než běžný útok hrubou silou. XSL Attack používá speciální algoritmus eXtended Sparse Linearization k získání klíče. Tato metoda vyžaduje pouze relativně malé množství známých otevřených textů oproti jiným metodám, které jich potřebují nerealisticky vysoký počet[26].

[25] podle [30]
[26] [31] (překlad autor diplomové práce)

V roce 2004 Diem dokázal, že v popisu algoritmu jsou chyby. Navíc má tato metoda vysoký pracovní činitel, takže neredukuje dobu potřebnou k prolomení AESu v porovnání s útokem hrubou silou.

„I když XSL pracuje proti některým moderním algoritmům, útok v současnosti znamená malé nebezpečí, co se týče praktické bezpečnosti. Stejně jako mnoho moderních kryptoanalytických výsledků by mohl být takzvanou *certifikační slabinou*: zatímco je rychlejší než útok hrubou silou, potřebné zdroje jsou stále příliš vysoké a je velice nepravděpodobné, že by jím mohly být skutečné systémy nějak ohroženy. Ovšem budoucí úpravy mohou zvýšit praktičnost tohoto útoku. Jelikož je takový typ útoku nový a neočekávaný, někteří kryptografové vyjádřili znepokojení nad algebraickou jednoduchostí šifer typu Rijndael.“[27]

Nebezpečí tak představují zejména tzv. *útoky postranním kanálem*, které se zaměřují nikoli na samotnou šifru, ale na její implementace v konkrétních systémech. V roce 2005 Bernstein oznámil *Cache Timing Attack* na zákaznický server šifrovaný AESem prostřednictvím OpenSSL. Útok vyžadoval přes 200 milionů vybraných otevřených textů získaných z údajů o časování serveru. Někteří odborníci namítají, že takovýto útok není v prostředí internetu praktický na vzdálenost větší než jeden hop.

Ve stejném roce jako Bernstein zveřejnili Osvik, Shamir a Tromer výsledky svých *cache timing* útoků proti AESu. Jeden z útoků umožnil získat celý klíč po vykonání pouhých 800 operací během 65 milisekund. Tento útok však vyžaduje, aby byl útočník schopen spouštět programy na stejném systému, na němž běží AES.

[27] [31]

IDEA

IDEA, celým názvem International Data Encryption Algorithm, je symetrická šifra, vyvinutá Dr. Xueija Lai a prof. Jamesem Masseyem ze Švýcarského federálního institutu pro technologii ve spolupráci s firmou Ascom. Používá se od roku 1992. Nejznámějším se stal po implementaci do šifrovacího programu PGP Phila Zimmermanna.

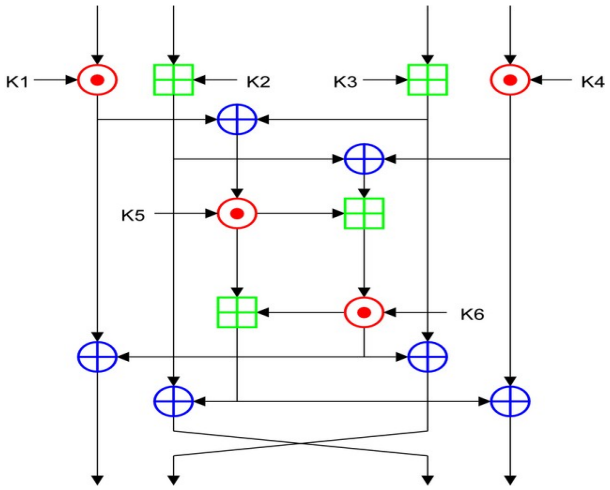
IDEA je bezpečnější a přibližně dvakrát rychlejší než DES. Je mnohem odolnější oproti útoku hrubou silou než zmíněný algoritmus.

Využívá klíč o délce 128 bitů, ze kterého se pro výpočet generuje celkem 52 podklíčů. Během šifrování otevřeného textu vykoná algoritmus osm kol stejných matematických operací. V každém kole použije šest podklíčů. Poslední čtyři klíče se používají k výstupní transformaci [20].

Dešifrování probíhá podle stejného algoritmu jako šifrování.
Jedno kolo šifrování vypadá následovně[28]:

- bitová nonekvivalence (na obrázku znázorněno modrým křížkem v kolečku),
- sčítání modulo 2^{16} (znázorněno zeleným čtvercem s křížkem),
- násobení modulo $2^{16} + 1$, kde nulová slova (0x0000) jsou interpretována jako 2^{16} (znázorněno červeným kolečkem s tečkou).

[28] Popis a obrázek přejat z [9]



obr. 11 – schéma průběhu jednoho kola šifrování algoritmem IDEA (zdroj: [9])

3.3.2 Asymetrické algoritmy

Diffie-Hellmanův algoritmus pro výměnu klíče[29]

Poprvé byl představen Whitfieldem Diffiem a Martinem Hellmanem roku 1976 v jejich slavné práci *New Directions in Cryptography* (Nové směry v kryptografii). Algoritmus závisí na problému řešení diskrétních logaritmů. Je založen na neproveditelném výpočtu sdíleného tajného klíče, daného dvěma veřejnými hodnotami, při dostatečně zvolené velikosti základu pro odvození klíče. Umožňuje výměnu tajného klíče mezi dvěma uživateli přes nezabezpečený kanál (viz obr. 12).

Na počátku obsahuje systém dva veřejné parametry *p* a *g*. Parametr *p* je tzv. základ, *g* se obvykle nazývá generátor a musí být menší než *p*. Pro každé číslo *n* mezi *1* a *p-1* (včetně) musí platit, že:

$$n = g^k \bmod p$$

[29] Popis algoritmu vytvořen podle [25]

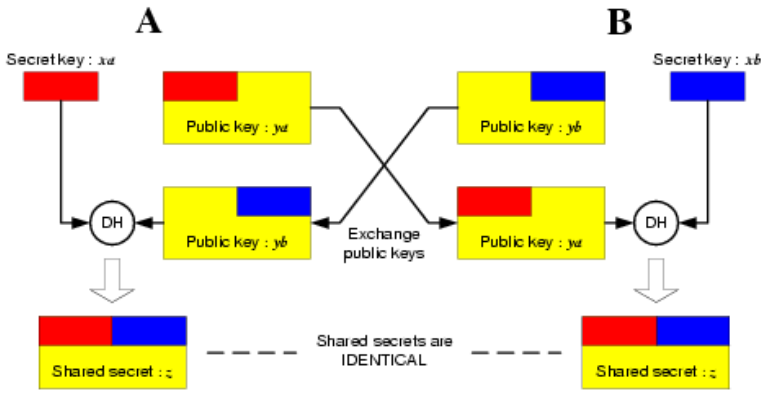
Pokud se obě strany chtějí pomocí Diffie-Hellmanova algoritmu dohodnout na tajném klíči, postup je následující:

- První strana nejprve vygeneruje náhodné privátní celé číslo *a*, strana druhá obdobně vygeneruje náhodné celé číslo *b*.

- Použitím parametrů *p* a *g* společně se svými tajnými hodnotami vygenerují své veřejné hodnoty.

  - Veřejný klíč první strany: $g^a \bmod p$
  - Veřejný klíč druhé strany: $g^b \bmod p$
  - Dojde k vzájemné výměně veřejných hodnot.
  - První strana vypočítá: $g^{ab} = (g^b)^a \bmod p$
  - Druhá strana vypočítá: $g^{ba} = (g^a)^b \bmod p$
  - Platí, že $g^{ab} = g^{ba} = k$
  - Z toho plyne, že obě strany od teď sdílejí stejný tajný klíč *k*. Mohou tedy komunikovat zabezpečeně.

Nespornou výhodou algoritmu je utajená výměna tajného klíče v nezabezpečeném komunikačním kanále. Původní návrh ovšem nepočítal se vzájemnou autentizací komunikujících stran. Je tedy napadnutelný útokem ze středu. Tento problém řeší další modifikace vycházející z tohoto algoritmu. Patří mezi ně například protokol STS (Station-To-Station), který vyvinuli Diffie, van Oorschot a Wiener roku 1992.



obr. 12 – princip algoritmu Diffie-Hellman (Zdroj: [22])

RSA[30]

RSA byl vyvinut roku 1978 [11] trojicí matematiků – Rivestem, Shamirem a Adlemanem. Jedná se o nejjednodušeji realizovatelný asymetrický algoritmus. Bezpečnost RSA se opírá o obtížnou faktorizaci velmi vysokých prvočísel. Nevýhodou je pomalost oproti symetrickým šifrám typu DES. Při srovnání softwarových realizací vychází RSA asi stokrát pomalejší.

Celý algoritmus vypadá následovně:

Na počátku je nutné zvolit dvě velmi vysoká prvočísla (řádově stovky míst). Tato čísla se označují jako *p* a *q*. Z nich se vypočítá číslo *n*:

$$n = p\,q$$

V dalším kroku se opět náhodně zvolí šifrovací klíč *e*. Musí platit, že *e* a *(p-1)(q-1)* nemají žádného společného dělitele (jsou nesoudělná). Vypočítáme dešifrovací klíč:

$$d = e^{-1} \bmod((p-1)(q-1))$$

Čísla *d* a *n* musejí být také nesoudělná.

[30] Podle [20]

Veřejný klíč je pak tvořen čísly *e* a *n*.

Privátním klíčem je číslo *d*.

Prvočísla *p* a *q* jsme potřebovali jen pro odvození klíčů, můžeme se jich tedy bezpečně zbavit. Nesmějí se nikdy dostat do nepovolaných rukou. Prolomit šifru by pak bylo velice jednoduché.

Šifrování zprávy označené jako *m*, se děje takto:

Otevřený text je rozdělen na bloky, které mají délku kratší, než je číslo *n*. Přibližně stejně dlouhé budou bloky zašifrovaného textu *c*, označené jako c_i. K šifrování slouží tato rovnice:

$$C_i = m_i^e \bmod n$$

K opětovnému dešifrování pak slouží vztah:

$$M_i = c_i^d \bmod n$$

Šifrování je také možné provést pomocí čísla *d* a dešifrovat klíčem *e*.

3.3.3 Hašovací funkce

Hašovací funkce jsou speciální transformační funkce, které libovolný vstupní řetězec převedou na výstupní řetězec s fixní délkou. Tento výstup se nazývá haš (digitální otisk, výtah zprávy atd.). Haš je tak vlastně zhuštěnou reprezentací vstupní zprávy nebo dokumentu, ze kterého byl vypočítán. Nepatrná změna na vstupu se přitom projeví dalekosáhlými změnami výstupu. Tato vlastnost se nazývá *lavinový efekt* (Avalanche Effect)[31]. Kryptografické hašovací funkce jsou používány ke kontrole integrity zpráv a k digitálním podpisům v nejrůznějších aplikacích informační bezpečnosti, jako je autentizace a kontrola integrity zpráv.

[31] [32]

Nejdůležitějšími vlastnostmi hašovací funkce jsou:

1) odolnost vůči *preimage*
2) odolnost vůči *second preimage* (slabá kolizní odolnost)
3) odolnost vůči kolizím (silná kolizní odolnost)

Odolnost vůči kolizím znamená: „Mělo by být těžké nalézt dvě rozdílné zprávy m_1 a m_2 takové, že haš(m_1) = haš(m_2). Vzhledem k možnému *narozeninovému útoku* to znamená, že výstup hašovací funkce musí být alespoň dvakrát větší, než je potřeba pro odolnost vůči *preimage*.“[32]

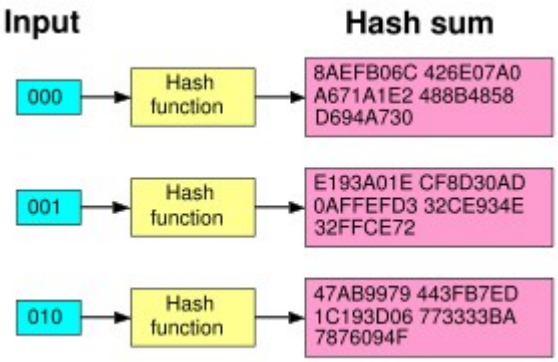
Útočník by tedy neměl být schopen najít dvě různé zprávy se stejným hašem, ani by neměl mít možnost zjistit cokoli o původní zprávě z jejího haše.

V současnosti jsou nejrozšířenějšími hašovacími algoritmy MD5 a SHA-1. V roce 2005 byly v obou objeveny bezpečnostní chyby a o dva roky později americký Národní ústav pro vědu a technologii (National Institute of Science and Technology – NIST) vyhlásil soutěž na návrh nové hašovací funkce, která dostane jméno SHA-3 a stane se novým americkým standardem.[33]

Lavinový efekt[34]

Jak bylo zmíněno výše, je lavinový efekt žádanou vlastností hašovacích algoritmů. Nepatrná změna vstupního řetězce tak způsobí rozsáhlou změnu výstupního haše (viz obr. 13).

[32] [33] (překlad autor diplomové práce)
[33] [33]
[34] podle [32]



obr. 13 – lavinový efekt hašovací funkce SHA (zdroj: [32])

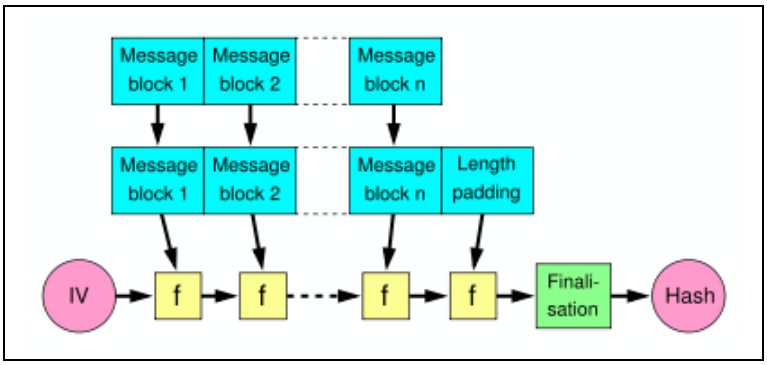
Lavinový efekt vnáší do algoritmu žádoucí prvek náhodnosti. Útočník díky němu nemůže z výsledného haše činit jakékoli predikce vstupu. V opačném případě by mohl analýzou zachyceného výstupu algoritmu odhadnout vstupní řetězec a vážně tím ohrozit bezpečnost algoritmu.

Merkle-Damgårdova hašovací funkce[35]

Nejpopulárnější hašovací funkce současnosti využívají Merkle-Damgårdovu hašovací konstrukci. Jedná se o metodu tvorby kryptografických hašovacích funkcí.

Aby mohla hašovací funkce zpracovávat rozdílně dlouhé vstupní řetězce do výstupů pevné délky, je vstup rozdělen do několika stejně dlouhých bloků, se kterými se pak operuje jednosměrnou kompresní funkcí, která zpracuje vstup pevné délky do výstupu o pevné délce. Merkle-Damgårdova hašovací funkce rozděluje vstupní řetězce do bloků, zpracuje každý blok kompresní funkcí a každý blok na vstupu při tom kombinuje s výstupem předešlé rundy.

[35] podle [34]



obr. 14 - Merkle-Damgårdova tvorba haše (zdroj: [34])

„Na obrázku (viz. obr 14 - pozn. autor diplomové práce) je jednosměrná kompresní funkce označená *f* , která transformuje dva vstupy s fixní délkou na výstup o stejné délce jakou má jeden ze vstupů. Algoritmus začíná inicializační hodnotou, tzv. *inicializačním vektorem* (IV). IV je fixní hodnota. Pro každý blok zprávy kompresní funkce *f* bere dosavadní výsledek, kombinuje ho s blokem zprávy a produkuje mezivýsledek. Poslední blok je doplněn nulami a jsou k němu připojeny bity reprezentující délku celé zprávy.“[36]

Tato metoda konstrukce byla vymyšlena Ralphem Merklem a Ivanem Damgårdem, kteří na ni přišli nezávisle na sobě. Oba dokázali, že „pokud je kompresní funkce *f* kolizně odolná, bude kolizně odolná i hašovací funkce, která ji používá.“[37]

Kolize haše[38]

Kolize haše nastává v případě, že dva různé vstupní řetězce vyprodukují identické výstupní haše. K tomuto jevu může dojít u jakékoli kryptografické hašovací funkce. Příčinou je skutečnost, že funkce musí produkovat výtahy o pevné délce z libovolného množství různě dlouhých vstupů. Čím lepší je algoritmus, tím méně často by se měly kolize objevovat.

[36] [34](překlad autor diplomové práce)
[37] [34]
[38] podle [35]

[35] definuje odolnost vůči kolizím takto:

Odolnost vůči kolizím

Proměnné:

H = vypočítaná hodnota haše
x = heslo 1
y = heslo 2

Slabá kolizní odolnost: pro dané x je těžké najít $y \neq x$ takové, že $H(x) = H(y)$

Uživatel zadá hodnotu (heslo), té budeme říkat iniciační hodnota *(x).* Pravděpodobnost, že iniciační heslo *(x)* a další nestejné heslo *(y)* vyprodukují stejný haš *(H)*, udává slabou kolizní odolnost.

Silná kolizní odolnost: je těžké najít jakákoli x a y taková, že $H(x) = H(y)$

Pravděpodobnost, že iniciační heslo *(x)* a další nestejné heslo *(y)*, která jsou libovolných hodnot, stále povedou k unikátnímu vypočítanému haši *(H)*, determinuje silnou kolizní odolnost.

Důležitou žádoucí vlastností kryptografické hašovací funkce je nemožnost vypočítat kolizi. Nalezení dvou libovolných hodnot, jejichž haše se shodují, se nazývá *kolizní útok*. Nalezení jedné libovolné hodnoty, jejíž haš koliduje s jiným (daným) hašem, se nazývá *preimage attack* (útok s využitím předobrazu). Úspěšný preimage attack je mnohem vážnější hrozbou než úspěšný kolizní útok[39].

[39] [35]

Narozeninový útok[40]

„Narozeninový útok je druh kryptografického útoku pojmenovaný podle využití matematiky v pozadí *narozeninového paradoxu*. S danou funkcí *f* je cílem útoku nalézt dva vstupy x_1 a x_2 takové, že $f(x_1) = f(x_2)$. Takový pár x_1, x_2 je nazýván kolize. Metodou používanou k nalezení kolize je prostý vypočet funkce *f* pro různé vstupní hodnoty, které mohou být vybrány náhodně nebo pseudonáhodně, dokud není stejný výsledek nalezen více než jednou. Vzhledem k narozeninovému paradoxu může být tato metoda poměrně efektivní. Specificky, pokud funkce *f(x)* poskytuje jakýkoli z *H* rozdílných výstupů se stejnou pravděpodobností a *H* je dostatečně velké číslo, pak očekáváme získání páru různých argumentů x_1 a x_2, pro které platí $f(x_1) = f(x_2)$, po vypočtení funkce pro v průměru asi 1,25 $\sqrt{H}$ rozdílných argumentů.“[41]

Narozeninový útok lze využít k falšování digitálních podpisů. Útočník může například vytvořit dvě smlouvy – jednu *„správnou“* a jednu *„podvodnou“*. Zkouší je vytvářet tak dlouho, dokud nebudou mít obě stejné haše. Pak nechá uživatele podepsat nejprve *„správnou smlouvu“* a podpis následně připojí k *„podvodné smlouvě“*. Jelikož mají obě smlouvy stejný haš, bude se uživatel jen těžko bránit, že podepsal jinou smlouvu, než tvrdí útočník.

Prevencí tohoto útoku je používat dostatečnou délku výstupního haše. Obecně se udává asi dvojnásobek bitové délky, která je potřebná na zamezení běžného útoku hrubou silou.

Preimage attack (útok s využitím předobrazu)[42]

Útočník se pokouší o *preimage attack*, pokud hledá takovou zprávu, která má specifický haš. Existují dva druhy útoků předobrazem:

1) first preimage attack – útočník má daný haš *h* a snaží se k němu najít zprávu *m* takovou, že $haš(m) = h$.
2) second preimage attack – útočník má pevnou zprávu m_1 a hledá jinou zprávu m_2 takovou, že $haš(m_2) = haš(m_1)$.

[40] podle [36]
[41] [36] (překlad autor diplomové práce)
[42] podle [37]

Narozdíl od kolizního útoku je u preimage attacku pevně daný haš nebo pevně daná zpráva, kterou útočník napadá. Náročnost takového útoku je vzhledem k narozeninovému paradoxu $2^{n/2}$ operací pro n-bitů dlouhou hašovací funkci.

SHA[43]

Skupina algoritmů SHA obsahuje celkem pět hašovacích funkcí. Patří sem SHA-1, SHA-224, SHA-256, SHA-384 a SHA-512. Poslední čtyři algoritmy jsou souhrnně nazývány SHA-2. SHA-1 produkuje haše o délce 160 bitů. Délka hašů ostatních funkcí je dána číslem v jejich názvu. Všechny funkce SHA byly navrženy americkou agenturou NSA a publikovány jako standard USA.

Hašovací funkce jsou považovány za bezpečné, pokud:

1. je nemožné najít zprávu, která koresponduje s daným výtahem
2. je nemožné najít dvě různé zprávy, které mají stejný výtah

SHA-1 je používána ve velkém množství bezpečnostních aplikací a protokolů (TLS, SSL, PGP, SSH, S/MIME, IPSec). Při svém uvedení byla považována za nástupkyni populárního staršího algoritmu MD5. Bezpečnost SHA-1 byla kompromitována vědci z oboru kryptologie. Úspěšné útoky na SHA-2 zatím nebyly hlášeny. Její algoritmus je však podobný SHA-1, proto byla vyhlášena otevřená soutěž na novou hašovací funkci, která ponese označení SHA-3 a bude publikována jako nový americký standard v roce 2012.

SHA-1

První šifrou z rodiny SHA byla tzv. SHA 0. Ta byla publikována americkým NISTem roku 1993 jako standard pro bezpečnou hašovací funkci (Secure Hash Standard). Velice krátce po publikování však byla SHA 0 stažena a v roce 1995 ji nahradila revidovaná verze – SHA-1. Nová verze opravuje bezpečnostní chybu původního algoritmu. SHA 0 i SHA-1 produkují 160 bitový haš vstupní zprávy o maximální délce 2^{64}-1 bitů. Oba algoritmy jsou

[43] podle [38]

založeny na principech obdobných těm, které jsou použity u hašovacích funkcí MD4 a MD5[44].

SHA-2

Nejnovějšími funkcemi z řady SHA jsou SHA-256 a SHA-512. Tyto funkce pracují s 32 (SHA-256), respektive 64 bitovým slovem (SHA-512). Jejich struktury jsou v podstatě identické, pouze používají rozdílné činitele posunu a aditivní konstanty. Další dvě funkce (SHA-224 a SHA-384) jsou jen zkrácené verze těchto dvou algoritmů[45].

MD5[46]

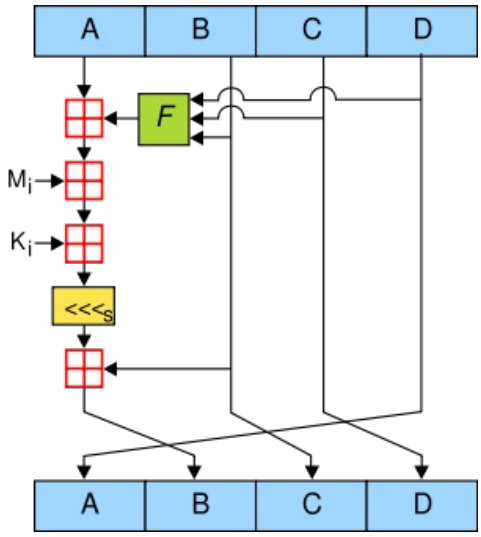
Hašovací funkci MD5 navrhl Ron Rivest již v roce 1991. Jejím cílem bylo nahrazení tehdy zastaralé funkce MD4. Zkratka MD značí Message Digest (výtah zprávy). MD5 je široce využívána. Nejen v aplikacích zajišťujících bezpečnost dat, ale také například ke kontrole integrity souborů. Haš MD5 je sekvence 32 hexadecimálních číslic.

Popis algoritmu

Vstupní zprávy s proměnnou délkou jsou funkcí MD5 zpracovány do výstupních hašů o délce 128 bitů. Vstup je rozdělen do stejně velkých bloků o délce 512 bitů. Zpráva je doplněna tak, aby její výsledná délka byla dělitelná číslem 512. To se dělá takto:

1) na konec zprávy je přidán jediný bit: 1
2) poté jsou přidány nuly, aby byla délka zprávy o 64 bitů kratší, než násobek 512
3) nakonec jsou zbývající bity doplněny 64 bitovou celočíselnou reprezentací délky původní zprávy v bitech

[44] [38]
[45] [38]
[46] podle [39]



obr. 15 – jedna ze 64 operací funkce MD5 (zdroj: [39])

Vysvětlivky ke schématu:

F – nelineární funkce použitá v každé rundě

M_i – 32 bitový blok vstupu zprávy

K_i – 32 bitová konstanta; různá v každé operaci

⋘ - bitová rotace vlevo o *s* míst; *s* je různé pro každou operaci

⊞ - sčítání modulo 2^{32}

„Hlavní algoritmus MD5 operuje se 128 bitovým stavem, rozděleným do čtyř 32 bitových slov, označených jako A, B, C a D. Ty jsou inicializovány na jisté fixní konstanty. Hlavní algoritmus potom střídavě operuje s každým 512 bitovým blokem, přičemž každý blok modifikuje stav. Zpracování bloku zprávy sestává ze čtyř podobných stupňů – nazývaných rundy. Každá runda obsahuje 16 podobných operací, založených na nelineární funkci F, modulárním sčítání a levé rotaci. Schéma (viz obr. 15) ilustruje jednu rundu. Existují 4 možné funkce F, každá z nich je použita v jedné rundě.

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

$\oplus, \wedge, \vee, \neg$ představují logické operace XOR, AND, OR a NOT“[47]

Kryptoanalýza MD5 a SHA

Nejrozšířenější hašovací funkce byly v posledních letech podrobeny důsledné kryptoanalýze, jejímž výsledkem bylo odhalení mnoha závažných chyb.[48]

V srpnu roku 2004 čínský tým soustředěný kolem profesorky Wangové na konferenci Crypto 2004 oznámil, že je schopen nalézt kolize pro několik hašovacích funkcí, včetně MD4 a MD5. Kolizi MD5 přitom dokázal nalézt během hodiny a čtvrt na počítači IBM p690[49].

V prosinci téhož roku publikoval Ondrej Mikle práci[50], v níž ukazuje, jak lze využít jediné datové kolize MD5 ke konstrukci útoku, při kterém se dva různé útočníkem vytvořené soubory uživatelům jeví jako shodné, prostřednictvím kontroly haše MD5 a digitálního podpisu.

O tři měsíce později (v březnu 2005) publikoval Vlastimil Klíma výsledek své práce[51]. Oznámil schopnost generovat kolize MD5 pro libovolnou inicializační hodnotu na běžném domácím notebooku. Klíma tak ukázal velké možnosti kryptoanalýzy algoritmu MD5. Zejména jde o nebezpečí falšování digitálních podpisů softwaru a padělání digitálních certifikátů.

[47] [39] (překlad autor diplomové práce)
[48] [40]
[49] [41]
[50] [42]
[51] [43]

Klíma k tomu dodává:

„K nalezení kolizí jsme nepoužili žádný superpočítač, pouze běžné domácí počítače. Autor prováděl své experimenty výhradně na notebooku, kde nalezl jak desetitisíce kolizí prvního bloku, tak i úplné kolize MD5 pro platnou inicializační hodnotu i volené inicializační hodnoty. ... Výsledek na běžném notebooku (Acer TravelMate 450LMi, Intel Pentium 1,6 GHz) je tento: během 8 hodin bylo nalezeno 331 kolizí prvního bloku a 1 úplná kolize MD5. Vzhledem k tomu, že nalezení jedné kolize prvního bloku trvalo čínskému týmu 1 hodinu na počítači IBM p690, nalezení 331 těchto kolizí by trvalo cca 331 hodin, což je 40 krát více. Výkony notebooku a velkého počítače lze těžko srovnávat z důvodu různých architektur, ale když uvažujeme, že uvedený počítač je 25 – 50 krát rychlejší než notebook (odhad poskytl Ondrej Mikle na základě poměru bogomips), dostáváme velmi hrubý odhad, že naše metoda hledání kolize prvního bloku je 1000 – 2000 krát rychlejší než v [případě čínského týmu – pozn. autor diplomové práce]. Naproti tomu hledání kolize druhého bloku je 2 – 80 krát pomalejší. Pokud srovnáme celkový čas hledání úplné kolize u čínského týmu (1,0 až 1,08 hodiny) s naším (8 hodin) na 25 – 50 krát pomalejším stroji, je naše metoda celkově 3 – 6 krát rychlejší. Všechna tato srovnání jsou orientační a autor si nečiní žádný nárok na jejich přesnost (přesné jsou pouze časové údaje).

Ukazuje se však, že:

- kolizi MD5 lze dnes vyhledat už i na notebooku
- naše metoda a čínská metoda se zásadně liší v rychlosti a pravděpodobně i v obsahu (v obou částech výpočtů)
- naše metoda je celkově rychlejší
- metoda pracuje pro jakoukoli zvolenou inicializační hodnotu.“[52]

O rok později přišel Klíma s další prací, ve které představil novou metodu kryptoanalýzy MD5 – tzv. tunelování. Díky této metodě se mu podařilo zkrátit dobu potřebnou k nalezení kolize na řádově deset až třicet sekund na běžném počítači[53].

[52] [43], s. 4
[53] [44]

Klíma vyjádřil možnost použití posledních objevů v kryptoanalýze také k prolomení funkcí rodiny SHA:

„Kromě toho jsou dnes již k dispozici nástroje, které ukazují, jak tvořit *diferenční cesty* jiné, než navrhla Wangová, a jinak. Když to spojíte s *metodami mnohonásobných modifikací* a *tunely*, máte k dispozici *Nástroj*e. Záleží jen na vaší invenci, co s nimi budete tvořit. Domnívám se, že je možno je použít pro hledání druhého vzoru zprávy nebo pro generování kolizí složitějších hašovacích funkcí jako jsou SHA-1 a SHA-2. Čas ukáže.“[54] (kurzíva Klíma)

„Současně s kryptoanalýzou je vyvíjena národní i mezinárodní aktivita na získání bezpečných hašovacích funkcí. Je potřeba vyvinout zcela nový koncept hašovacích funkcí, který vyloučí používání triviálních funkcí typu MD a SHA, jejichž technologie odpovídá osmdesátým létům minulého století. V současné době jsou k dispozici technologie odolné vůči diferenciální a lineární kryptoanalýze, které je potřeba pouze vhodně přesunout z oblasti blokových šifer do oblasti hašovacích funkcí a vyvinout teorii, která poskytne mnohem větší záruky bezpečnosti než ta současná. Současné hašovací funkce mají totiž, až na základní koncept, teoretické základy značně chabé. Přesun technologií blokových šifer do hašovacích funkcí podporuje i Eli Biham, spoluobjevitel diferenciální kryptoanalýzy.“[55]

Závěry k těmto kryptoanalytickým zjištěním jsou následující:

„Řada předních kryptologů se shoduje v tom, že je nutno zahájit práce na veřejné mezinárodní soutěži na nový koncept hašovacích funkcí, neboť iterativní funkce nesplňují požadované bezpečnostní vlastnosti. Uvedené odhalené vlastnosti jsou teoretického rázu, ale jednoho dne by se mohly projevit zcela prakticky. Proto je nezbytná změna konceptu.“[56]

Jako reakci na novou situaci vydal americký standardizační úřad NIST v srpnu 2004 prohlášení, ve kterém doporučil používat funkce třídy SHA-2 a do roku 2010 zcela opustit používání zastaralé funkce SHA-1.

[54] [45], s. 4
[55] [45], s. 4 - 5
[56] [46], s. 3

Třída hašovacích funkcí SHA-2 se zdá zatím poskytovat dostatečnou složitost pro případné pokusy o prolomení. Ovšem její základ je podobně jako u SHA-1 a MD5 iterativního charakteru, takže i u nich se projevují výše zmíněné teoretické nedostatky. Proto by světoví kryptologové měli přijít s novým konceptem hašovacích funkcí[57].

Silně se nedoporučuje používat funkce, u nichž byly nalezeny kolize, ve všech aplikacích vyžadujících bezkoliznost (časová razítka, elektronické podpisy apod.). V těchto případech totiž kolize umožňuje vytvoření dvou různých zpráv se stejným platným digitálním podpisem[58].

Klíma[59] doporučuje přejít na funkce SHA-2 nebo Whirlpool a funkce SHA-1 a MD5 používat jen tam, kde nevadí porušení vlastnosti bezkoliznosti (tedy ne v digitálních podpisech a obdobných aplikacích).

Zajímavým se jeví hlavně použití funkce Whirlpool, která byla vydána jako mezinárodní norma ISO a byla přijata jako kryptografická technika v rámci evropského projektu NESSIE[60].

[57] [46], s. 4
[58] [47], s. 2
[59] [46], s. 4
[60] [48], s. 13

Whirlpool

Whirlpool je hašovací algoritmus navržený Vincentem Rijmenem a Paulem Barretem. Funkce pracuje se zprávami do maximální délky 2^{256} bitů. Produkuje haš zprávy o velikosti 512 bitů.

Whirlpool používá Merkle-Damgårdovo posílení a Miyaguchi-Preneelovo hašovací schéma s dedikovanou 512 bitovou blokovou šifrou nazvanou *W*. Funguje následovně:

Řetězec bitů, který má být hašován, je doplněn jediným bitem: 1, poté sekvencí nul a nakonec původní délkou řetězce (ve formě 256 bitového celého čísla), takže výsledná délka po doplnění je násobkem 512. Výsledný řetězec je rozdělen do řady 512 bitových bloků m_1, m_2, ...m_t, které jsou potom použity ke generování sekvence mezivýsledků haše H_0, H_1, H_2, ...H_t. H_0 je samozřejmě řetězec 512 „nulových“ bitů. Aby vypočítala H_i, musí *W* zašifrovat m_i, za použití H_{i-1} jako klíče, a XORovat výsledný šifrovaný text jak s H_{i-1}, tak s m_i. Nakonec dostaneme H_t - haš zprávy[61].

Velkou výhodou Whirlpoolu proti většině současných hašovacích funkcí je jeho široká škálovatelnost. Algoritmus není primárně orientován na jednu konkrétní platformu, ale je schopen pracovat na mnoha různých platformách. Nevyžaduje příliš velkou paměťovou kapacitu, takže je vhodný k implementaci v omezených prostředích, jako jsou smart karty. Whirlpool také nepoužívá žádné neobvyklé instrukce, které by musely být vestavěny do procesoru, a zároveň má velice dlouhý haš, jenž poskytuje zvýšenou ochranu vůči narozeninovým útokům[62]

[61] podle [49]
[62] [49]

Zabezpečení bezdrátových sítí Wi-Fi

V poslední době čím dál více rozšířené bezdrátové sítě jsou specifickou oblastí, ve které se uplatní kvalitní kryptografická ochrana. K přenosu dat využívají rádiových vln, což je činí mnohem náchylnější k odposlechu než klasické „kabelové“ sítě. Jedním z útoků na Wi-Fi síť je tzv. Evil Twin Phishing.

Evil Twin Phishing[63]

Je variantou klasického *phishingového útoku*. Phishing je pokus získat citlivá data, jako jsou přihlašovací jména a hesla, čísla kreditních karet atd.

Evil Twin Phishing nachází uplatnění v bezdrátových sítích Wi-Fi. Útočník oklame uživatele bezdrátové sítě, aby připojili svůj laptop nebo mobilní telefon k falešnému *hotspotu* tím, že se maskuje jako legitimní poskytovatel připojení (provider).

Bezdrátová zařízení se připojují k internetu skrze „hotspoty“ – blízké přípojné body, na které se zaměří. Útočník s příslušným technickým vybavením dokáže zaujmout místo takového hotspotu (a vytvořit tak jeho *„zlé dvojče“* – evil twin). Uživatelé se pak pod dojmem, že se připojují k hotspotu poskytovatele, připojí k základní stanici útočníka. Ten jim podstrčí falešné přihlašovací tabulky, ze kterých dokáže vyčíst jejich přihlašovací údaje, stejně jako další důvěrné informace.

Wired Equivalent Privacy (WEP)[64]

Algoritmus WEP byl uveden v roce 1999. Měl Wi-Fi sítím zajistit stejnou úroveň ochrany, jakou mají „kabelové“ sítě.

Bohužel však v roce 2001 kryptoanalytici odhalili v algoritmu několik závažných slabin. Dnes tak může být spojení chráněné WEPem prolomeno během několika minut. Navzdory slabinám je však stále používán, i když úroveň zabezpečení, kterou poskytuje,

[63] podle [50]
[64] podle [51]

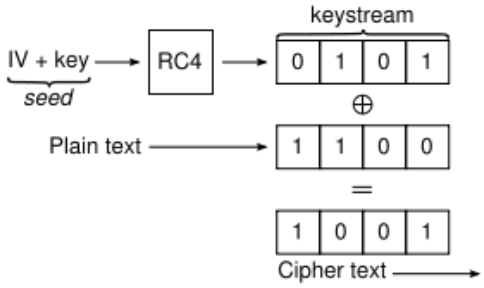
zabrání nanejvýše náhodnému a často nezáměrnému vstupu neoprávněného uživatele do Wi-Fi sítě.

Popis algoritmu

WEP využívá proudovou šifru RC4 pro zajištění důvěrnosti a kontrolní součet CRC-32 pro kontrolu integrity.

„Standardní 64 bitový WEP používá 40 bitový klíč (také známý jako WEP-40), který je zřetězený s 24 bitovým inicializačním vektorem (IV), aby společně vytvořily tzv. RC4 traffic key. V době, kdy byl původní standard WEP navržen, omezovala americká exportní omezení kryptografických technologií velikost klíče. Jakmile byla omezení zrušena, všichni významní výrobci implementovali rozšířený 128 bitový WEP protokol používající 104 bitovou velikost klíče (WEP-104).

128 bitový klíč WEP je téměř vždy zadáván uživatelem jako řetězec 26 hexadecimálních znaků (0-9 a A-F). Každý znak reprezentuje 4 bity klíče. 4 x 26 = 104 bitů; přidáním 24 bitového IV dostaneme 128 bitový WEP klíč. 256 bitový WEP systém je dostupný u některých dodavatelů a, podobně jako u výše zmíněného, je 24 bitů pro IV, což zanechává 232 bitů pro ochranu. To je typicky zadáváno jako 58 hexadecimálních znaků (58 x 4 = 232 bitů) + 24 bitů IV = 256 bitů ochrany WEP.“[65]



obr. 16 - základ šifry WEP: klíč RC4 XORovaný otevřeným textem (zdroj: [51])

[65] [51] (překlad autor diplomové práce)

Related-key Attack a WEP

„Related-key attack je jakákoli forma kryptoanalýzy, kdy útočník může vysledovat zpracování šifry pro několik různých klíčů, jejichž hodnoty jsou zpočátku neznámé, ale útočník zná některé matematické vztahy spojující klíče. Například: útočník může vědět, že posledních 80 bitů klíče je vždy stejných, i když zpočátku neví, jaké tyto bity jsou. To se na první pohled může zdát nerealistickým - jistě by se nemělo stávat, aby útočník donutil lidského kryptografa šifrovat otevřené texty pod množstvím tajných klíčů, nějakým způsobem spojených. Ovšem moderní kryptografie je implementována za použití komplexních počítačových protokolů, často neposouzených kryptografy, a proto je v mnoha případech velice snadné Related-key attack provést.“[66]

Příkladem protokolu, který selhal přičiněním tohoto typu útoku, je WEP.

Kvůli problémům s bezpečnostní WEPu se dnes doporučuje přejít na WPA2 nebo alespoň WPA. Oba algoritmy jsou mnohem bezpečnější než WEP. K přidání podpory pro WPA nebo WPA2 je potřeba nahradit některé staré přístupové body k Wi-Fi sítím, u některých stačí pouze upgradovat jejich firmware.

Wi-Fi Protected Access (WPA)[67]

„Wi-Fi Protected Access (WPA a WPA2) je třída systémů pro zabezpečení bezdrátové počítačové sítě. Byla vytvořena jako odpověď na množství závažných slabin, které vědci objevili v předešlém systému – Wired Equivalent Privacy (WEP). WPA implementuje většinu ze standardu IEEE 802.11i a byl zamýšlen jako prostředek k přechodnému nahrazení WEPu, zatímco byl připravován standard IEEE 802.11i. WPA je specificky navržen ke spolupráci se síťovými Wi-Fi kartami, vyrobenými před jeho příchodem (skrze upgradování firmwaru), ale ne nezbytně s první generací bezdrátových přístupových bodů. WPA2 implementuje plný standard, ale nefunguje na některých starších síťových kartách.“[68]

[66] [51] (překlad autor diplomové práce)
[67] [52]
[68] [52] (překlad autor diplomové práce)

Existují dva druhy WPA: pro komerční a osobní využití. Komerční verze počítá s používáním autentizačního serveru IEEE 802.1X, který distribuuje různé klíče každému uživateli. Osobní WPA používá méně škálovatelný mód *pre-shared key*, kde každý povolený počítač využívá stejnou vstupní frázi. V tomto módu bezpečnost závisí na síle a utajení vstupní fráze.

K šifrování dat používá WPA algoritmus RC4 se 128 bitovým klíčem a 48 bitovým IV. Oproti WEPu je WPA vylepšen o protokol TKIP (Temporal Key Integrity Protocol), který dynamicky mění klíče za chodu systému. Společně s mnohem větším IV dává takovéto řešení větší odolnost vůči útokům zaměřeným na získání klíče.

TKIP je „obal“, který ovinuje existující šifrování WEP. Zahrnuje stejný šifrovací engine a algoritmus RC4 definovaný pro WEP. Ovšem klíč používaný pro šifrování v TKIP je dlouhý 128 bitů. To řeší první problém WEPu – příliš krátký klíč. Důležitou vlastností TKIP je, že mění klíč používaný pro každý paket.

„Každý paket přenesený pomocí TKIP má unikátní 48 bitové sériové číslo, které je inkrementováno pokaždé, když je paket přenesen a použit jak v IV, tak jako součást klíče. Přidání čísla sekvence do klíče zaručuje, že klíč je rozdílný pro každý paket. To řeší další problém WEPu, nazývaný *kolizní útoky*, které mohou nastat, pokud je stejný klíč použit pro dva rozdílné pakety. S rozdílnými klíči nehrozí nebezpečí kolizí.

Použití sériového čísla paketu zároveň jako IV také pomáhá redukovat další problém WEPu, nazývaný *útoky opakováním*. Protože 48 bitovému číslu sekvence zabere tisíce let, než se bude opakovat, nikdo nemůže zopakovat staré pakety z bezdrátového spojení – byly by detekovány jako „mimo pořadí“, neboť čísla sekvencí nebudou v pořádku.“[69]

Dalším vylepšením je kontrola *payload integrity*. WEP využívá cyklickou redundantní kontrolu (CRC), která je nespolehlivá, neboť je možné vyměnit payload a aktualizovat CRC zprávy bez znalosti klíče. WPA oproti tomu spoléhá na MIC (Message Integrity Code) s algoritmem Michael, který zabraňuje útokům opakováním (díky zabudovanému počítadlu rámců – frame counter).

[69] [53] (překlad autor diplomové práce)

„Prodloužením délky klíčů a inicializačních vektorů, snížením počtu paketů poslaných s příslušnými klíči a přidáním systému bezpečné verifikace zprávy algoritmus WPA značně ztěžuje prolomení do bezdrátové sítě. Algoritmus Michael byl to nejsilnější, co mohli designéři WPA navrhnout, aby přitom zajistili funkci na většině starých síťových karet.“[70]

[70] [52] (překlad autor diplomové práce)

4 Implementace kryptologické ochrany

V minulosti probíhala veškerá internetová komunikace nezabezpečeně. To znamená, že veškerá přenášená data, včetně přihlašovacích údajů a hesel, byla posílána jako otevřený text. Postupný nárůst počtu internetových útoků však vedl k vývoji různých forem zabezpečení. K doposud užívaným internetovým protokolům tak přibily jejich zabezpečené verze. Dále byly vyvinuty programy pro šifrování souborů na disku i e-mailové komunikace a v neposlední řadě také systém elektronického podpisu. Všechny tyto formy zabezpečení využívají kryptografii. Budou popsány níže v této práci.

„Kryptografický protokol je sdílený algoritmus definovaný posloupností kroků, které precizují aktivity vyžadované na dvou či více entitách s cílem dosáhnout určitého bezpečnostního cíle“[71].

Účelem (cílem) kryptografických protokolů bývá autentizace účastníků protokolu, utvoření dohody o dále použitém kryptografickém klíči, výměna těchto klíčů, bezpečné předávání zašifrovaných dat, ověřování integrity dat atd.[72]

4.1 Zabezpečené protokoly

4.1.1 SSH (Secure Shell)[73]

SSH je protokol, který umožňuje zabezpečenou komunikaci mezi klientem a serverem. Zajišťuje autentizaci uživatelů, šifrování přenášených souborů i kontrolu jejich integrity. Protokol SSH nachází uplatnění jako náhrada nezabezpečených protokolů Telnet a FTP. Využití SSH umožňuje [1, 4]:

- Zabezpečené vzdálené přihlašování. Tím nahrazuje protokol Telnet. SSH odesílá uživatelovo přihlašovací jméno i heslo v zašifrované podobě. Veškerá další komunikace po přihlášení je také šifrovaná.

[71] [18] s. 6
[72] [26] s 134
[73] Zpracováno podle [1]

- Zabezpečený přenos souborů. V tomto případě jde o náhradu protokolu FTP, který veškerá data, včetně přihlašovacích údajů, posílá po síti bez jakéhokoli zabezpečení.
- Zabezpečené spouštění vzdálených příkazů.
- Řízení přístupu k datům na disku i systémovým nástrojům.
- Směrování portů. Při něm je TCP/IP připojení vedeno přes vrstvu SSH, která veškerou komunikaci šifruje.
- Správu přihlašovacích klíčů do zabezpečených systémů.

Autorem tohoto protokolu i jeho softwarové implementace je Tatu Ylönen z Helsinské technologické univerzity. SSH existuje ve dvou verzích, které nejsou navzájem kompatibilní. Program SSH 1 (využívající v současnosti protokol SSH 1,5) je volně šiřitelný, zatímco SSH 2 (využívající protokol SSH 2) je komerční produkt. Společné vlastnosti i rozdíly mezi oběma řešeními budou popsány níže.

Principy SSH

Nasazení Secure Shell poskytuje uživateli možnost využít šifrování, kontroly integrity, autentizace, autorizace a směrování.

Šifrování

SSH šifruje přenášená data použitím množství šifrovacích algoritmů. Z nejznámějších sem patří DES, 3DES a IDEA.

Integrita

Integrita je v SSH2 zaručena použitím hašovacích algoritmů MD5 a SHA-1. V SSH 1 pak slabší metodou - cyklickou redundantní kontrolou (CRC-32). Hašovací algoritmy slouží k získání digitálního otisku (výtahu, haše) elektronického dokumentu.

Autentizace

Autentizace probíhá ve dvou směrech. Nejenže server ověřuje klienta, ale i klient provádí autentizaci serveru, aby se ujistil, že komunikace nebyla podvržena. Tím se dá zabránit útoku ze středu. SSH využívá jak autentizaci heslem, tak veřejné klíče uživatelů i další metody (karty SecurID atd.). Na způsobu autentizace se na začátku komunikace dohodnou server a klient.

Autorizace

Autorizace je nastavení práv pro jednotlivé uživatele. Tento proces nastává bezprostředně po autentizaci.

Směrování

Směrování umožňuje uživateli využívat výhod zabezpečené komunikace i při práci s dalšími službami na úrovni TCP/IP. V praxi to vypadá tak, že se jiný protokol (např. Telnet) zapouzdří do SSH, čímž můžeme využívat šifrovaného přenosu dat.

Principy SSH 1

Před zahájením šifrovaného přenosu dat je třeba nejprve zajistit zabezpečené spojení mezi klientem a serverem. V první řadě klient odešle serveru požadavek na síťovou komunikaci. Následuje výměna informací o podporovaných verzích protokolu SSH na straně klienta i serveru. Pokud jsou verze vzájemně kompatibilní, komunikace může pokračovat a klient i server se přepnou na paketový protokol.

Nyní je třeba, aby se server autentizoval klientovi. Tato část komunikace probíhá nešifrovaně. Server pošle klientovi své dva klíče – klíč hostitele a serveru, dále kontrolní bajty, které musí klient použít ve své odpovědi, aby byla přijata serverem, a nakonec také informaci o podporovaných metodách autentizace, šifrování a komprese. Nyní komunikující

strany vypočítají algoritmem MD5 tzv. identifikátor relace SSH. K výpočtu použijí klíč hostitele, klíč serveru a kontrolní bajty.

Klient vygeneruje klíč relace a zašifrovaný ho předá serveru. Šifrování provede nejprve klíčem hostitele a poté ještě klíčem serveru. Od této chvíle už veškerá komunikace probíhá šifrovaně. O šifrování i dešifrování se stará právě klíč relace.

Pokud klient v této chvíli obdrží od serveru klíčem relace zašifrovanou potvrzovací zprávu, znamená to, že vše proběhlo v pořádku. Server se tímto způsobem autentizuje. Nyní je řada na klientovi, aby se autentizoval.

Nejbezpečnější metodou je autentizace veřejným klíčem.[74] Probíhá následovně:

- Klient pošle serveru požadavek na autentizaci veřejným klíčem spolu s identifikací konkrétního klíče. Jako identifikátor požadavek obsahuje modul klíče. Klíčem je implicitně zvolen algoritmus RSA.
- Server přečte autorizační soubor cílového účtu a zkusí vyhledat záznam s odpovídajícím klíčem. Pokud žádný takový záznam v souboru není, požadavek na autentizaci je zamítnut.
- Jestliže je v autorizačním souboru odpovídající záznam, server záznam načte a zjistí z něj veškerá omezení, která se daného klíče týkají.
- Server vygeneruje náhodný, 256 bitový řetězec, který slouží jako výzva. Zašifruje jej veřejným klíčem klienta a odešle jej klientovi.
- Klient obdrží výzvu a dešifruje ji pomocí odpovídajícího soukromého klíče. Pak výzvu zkombinuje s identifikátorem relace, výsledek předá hašovací funkci MD5 a hašovou hodnotu pošle serveru jako svoji odpověď na jeho výzvu.
- Server vypočítá z výzvy a identifikátoru relace tutéž hašovou hodnotu MD5; pokud se odpověď klienta shoduje, autentizace uspěje.

[74] [1] s. 59

Principy SSH2

Jak již bylo uvedeno výše, není SSH 2 kompatibilní s SSH 1. Změny v SSH 2 oproti předchozí verzi jsou následující:

- Klient a server si při vzájemné komunikaci mohou dohodnout nejen jaký šifrovací algoritmus použijí, ale také další algoritmy (hašovací, pro výměnu klíče relace...).
- Možnost použití více metod výměny klíčů. Všechny implementace musejí podporovat Diffie-Hellmanovu metodu spojenou s hašovací funkcí SHA-1.
- Možnost přiřazení certifikátů k veřejným klíčům.
- Flexibilnější autentizace.
- Lepší kontrola integrity použitím algoritmů MAC (Message Authentication Code).
- Možnost změny klíče relace v jejím průběhu.

Algoritmy použité v SSH 1

veřejný klíč: RSA

hašování: MD5, CRC-32

symetrická šifra: 3DES, IDEA, ARCFOUR, DES

Algoritmy použité v SSH 2

veřejný klíč: DSA, DH
hašování: SHA-1, MD5
symetrická šifra: 3DES, Blowfish, Twofish, CAST-128, IDEA, ARCFOUR, AES

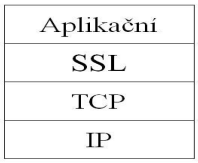
Jelikož SSH 1 obsahuje chyby, které ho činí zranitelným například útoky ze středu, je dnes obecně považován za zastaralý. Ve všech verzích SSH je velice důležité verifikovat neznámé veřejné klíče před tím, než je uživatel přijme jako platné. V opačném případě může útočník odhalit heslo, čímž se mu otevře prostor k útoku ze středu.[75]

[75] [54]

4.1.2 SSL (Secure Socket Layer)

SSL je protokol firmy Netscape. Zajišťuje bezpečnost dat mezi vrstvou transportní a aplikační. Používá se k zabezpečení protokolů http, smtp, pop3, ftp a dalších. Na internetu se jako oficiální standard používá odvozený protokol TLS (Transport Layer Security).

SSL dokáže šifrovat data, zajistit jejich integritu, autentizovat server i klienta. Odesílatele ověřuje pomocí digitálních podpisů. Všechny protokoly, které SSL pro zabezpečení využívají, mají v názvu koncovku –S (HTTPS, FTPS atd.).



obr. 17 – vrstva SSL a okolní vrstvy síťové komunikace (Zdroj: [4])

Komunikace přes SSL probíhá následovně:

Nejprve se klient a server dohodnou na algoritmech, které během přenosu použijí. Pro výměnu klíče lze použít např. DSA, RSA, Diffie-Hellman atd. Podporovanými symetrickými šiframi jsou RC4, 3DES, AES. Hašovací algoritmy používané v SSL jsou MD5, SHA.[76]

- Server zašle svůj veřejný klíč klientovi.
- Klient obdrží klíč, vygeneruje náhodné číslo (tzv. *premaster secret*), zašifruje ho získaným klíčem serveru a odešle zpět.
- Nyní se vytvoří tzv. *master secret*. K tomu se použijí předchozí náhodná čísla a *premaster secret*.
- Klient a server opět vygenerují náhodná čísla. Navzájem si je pošlou.

[76] [55]

- Komunikující strany použijí *master secret* a náhodné číslo k vytvoření základu pro tvorbu klíče relace.
- V poslední fázi se vytvoří klíč relace z bitů, vybraných z výsledku předcházejícího kroku.

SSL se mimo jiné používá k ochraně e-mailové komunikace. Naneštěstí nezaručí bezpečnost po celé cestě mezi dvěma koncovými uživateli. Může totiž zabezpečit jen cestu mezi klientem a serverem nebo dvěma servery. V počítači klienta či serveru jsou data vždy opět dešifrována. Hrozí tedy nebezpečí odposlechu, pokud data míří přes nedůvěryhodné servery[77].

4.1.3 S-HTTP

S-HTTP neboli Secure Hypertext Transport Protocol byl vyvinut roku 1994 jako zabezpečená verze protokolu HTTP. Za jeho zrodem stojí společnost Enterprise Integration Technologies. Protokol využívá šifrování zpráv, přičemž údaje o použitém algoritmu, jakož i o způsobu jeho dešifrování, jsou uloženy v hlavičce zprávy. Hlavička je sama o sobě nezašifrovaná a slouží pouze k předání těchto informací příjemci.

S-HTTP používá symetrické i asymetrické šifrovací algoritmy, autentizaci, digitální podpisy i certifikáty.

Komunikace v S-HTTP:

- Obě strany se nejprve dohodnou, jaký kryptografický algoritmus použijí.
- Poté si mezi sebou vymění svoje veřejné klíče. Tím končí nezabezpečená část komunikace.
- Klient a server si předají zašifrovaný klíč relace.

[77] Popis proveden dle [4]

- Klíč relace je od teď používán k šifrování těla zprávy symetrickou šifrou. Navenek se protokol chová stejně jako HTTP. Uživatel tedy na první pohled rozdíl nepozná. Ovšem žádná jeho data již nebudou přenášena v otevřené podobě[78].

4.1.4 IPSec

Vývoj IPSec započal roku 1992. První implementace se protokol dočkal o 3 roky později v roce 1995.

Protokol IPSec pracuje, jak již jeho název napovídá, na úrovni TCP/IP vrstvy. Výhodou tohoto principu je nezávislost na použitých aplikacích. IPSec před přenosem po síti šifruje veškerá data, která mu předají vyšší vrstvy. Protokol pracuje ve dvou možných režimech.

Prvním je režim *transportní*. Při jeho použití se každému paketu přidá bezpečnostní hlavička, která určuje, jak se mají chránit data paketu.

Tunelový režim oproti tomu dokáže ochránit celý datagram. Používá se hlavně k vzájemnému propojení dvou nezávislých sítí.

IPSec využívá pro svou činnost dalších třech protokolů. Prvním je AH (Authentication Header). Má na starost autentizaci odesílatele a zároveň zajištění integrity přenášených dat. K tomu využívá hašovacích algoritmů SHA-1 a MD5.

Druhým protokolem je ESP (Encapsulated Security Payload), který zajišťuje šifrování algoritmem DES i jinými [23].

V průběhu práce s IPSec se uživatel může rozhodnout, kterých protokolů využije a tedy zda data bude jak šifrovat, tak autentizovat, nebo se rozhodne jen pro jednu z těchto funkcí[79].

[78] Popis protokolu podle [26]
[79] [4] s. 114

Posledním z protokolů je IKE (Internet Key Exchange). Využívá se k dohodnutí šifrovacích klíčů mezi oběma stranami komunikace. K tomu slouží Diffie-Hellmanova metoda. Podobně jako v předchozích případech si klient a server následně domluví způsoby šifrování a autentizace, poté se navzájem autentizují, po čemž mohou započít se vzájemnou výměnou šifrovaných dat.

Algoritmy používané v protokolu IPSec jsou SHA-1 a MD5 pro kontrolu integrity, 3DES a AES pro zajištění bezpečnosti.[80]

4.1.5 S/MIME

S/MIME je zabezpečenou verzí protokolu rozšíření elektronické pošty MIME (Multipurpose Internet Mail Extension) vyvinutou společností RSA. Umožňuje digitální podepisování i šifrování zpráv veřejným klíčem. K zabezpečení zprávy používá protokol CMS. Poté zprávu převede do sedmibitové podoby. Data jsou digitálně podepsána, zašifrována a mohou být bezpečně odeslána příjemci[81].

S/MIME je standardem pro šifrování a podepisování zpráv. Poskytuje autentizaci, integritu zpráv, nepopiratelnost, soukromí a bezpečnost dat. Funkce S/MIME jsou vestavěny do mnoha současných poštovních klientů a spolupracují mezi nimi.[82]

„Před tím, než může používat S/MIME, musí uživatel získat a nainstalovat osobní certifikát od certifikační autority (CA) zaměstnavatele nebo od některé z veřejných CA. Nejlepší je používat oddělené klíče (a k nim příslušné certifikáty) pro podepisování a šifrování, neboť je tak zajištěno používání šifrovacího klíče bez kompromitování vlastnosti „nepopiratelnosti“. Šifrování vyžaduje držení certifikátu příjemce (což je většinou automatické; po obdržení zprávy od strany s platným certifikátem). Je sice technicky možné odeslat šifrovanou zprávu příjemci (za použití jeho certifikátu) bez toho, aby měl uživatel nainstalován svůj vlastní certifikát pro digitální podepisování, ovšem většina klientů S/MIME

[80] [56]
[81] [4] s. 112
[82] [57]

bude vyžadovat instalaci uživatelova certifikátu před tím, než mu povolí šifrovanou zprávu odeslat.“[83]

Jednou z nevýhod S/MIME je skutečnost, že po vymazání jakéhokoli certifikátu (ať již expirovaného nebo platného) ze seznamu certifikátů nepůjdou jím zašifrované zprávy dešifrovat.[84]

4.1.6 PGP

Pretty Good Privacy je šifrovací program vyvinutý Philem Zimmermannem. Umožňuje zašifrovat soubory na disku algoritmem IDEA, vytvářet, spravovat a certifikovat tajné i veřejné klíče, šifrovat e-mailovou komunikaci nebo používat digitální podpis.

PGP využívá k potvrzení důvěryhodnosti certifikátů tzv. *Web of Trust* (Síť důvěry). Princip sítě je jednoduchý. Každý uživatel má možnost podepsat certifikáty těch uživatelů, kterým on sám důvěřuje. Čím víc podpisů certifikát „nasbírá“, tím důvěryhodnějším se jeví ostatním. Díky vzájemnému propojení všech uživatelů tak vzniká jakási „síť“, v níž se jednotliví uživatelé vzájemně zaručují za certifikáty ostatních.

Web of Trust byl používán od počátku uvedení PGP do praxe. V pozdějších verzích se objevila i druhá forma ověřování, která je analogická klasickému stromu certifikačních autorit. Tím je zajištěno ověření právoplatného majitele podpisu. Majitel zároveň může podepisovat certifikáty, které se nacházejí o jednu úroveň pod jeho certifikátem.

Úroveň nula je v podstatě to samé, jako Web of Trust. První úroveň odpovídá certifikační autoritě. Uživatel na úrovni jedna může ověřit a podepsat libovolné množství certifikátů úrovně nula. Úroveň dva odpovídá seznamu důvěryhodných certifikačních autorit. Takovéto seznamy se nacházejí například ve webových prohlížečích. Na úrovni dvě může uživatel ověřovat a podepisovat certifikáty úrovně jedna.[85]

---

[83] [57]
[84] [57]
[85] [58]

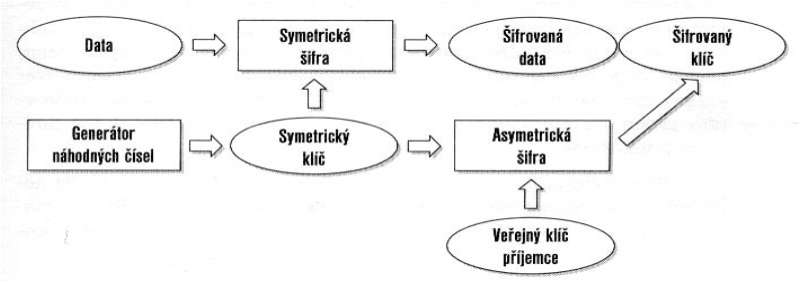
Šifrování zpráv

PGP řeší problém výpočetní náročnosti asymetrického šifrování tak, že nejprve vygeneruje náhodný klíč relace, kterým zašifruje požadovanou zprávu a tento klíč posléze zašifruje silnou asymetrickou šifrou. Zašifrovaný klíč přiloží ke zprávě a odešle. Příjemce nejprve dešifruje klíč a pomocí něj pak samotnou zprávu. Protože klíč relace je krátký, je jeho asymetrické šifrování mnohem rychlejší, než kdybychom tímto způsobem zabezpečovali celou zprávu.[86]

K šifrování klíče relace používá PGP algoritmus RSA. Samotná zpráva je pak šifrována algoritmem IDEA.

Digitální podpis

PGP umožňuje digitální podepisování zpráv. Uživatel tak může ověřit, jestli zpráva nebyla po cestě změněna útočníkem. Podpis také slouží k ověření identity odesílatele. K tvorbě digitálního podpisu používá PGP hašovací funkci MD5 a šifrovací algoritmus RSA.



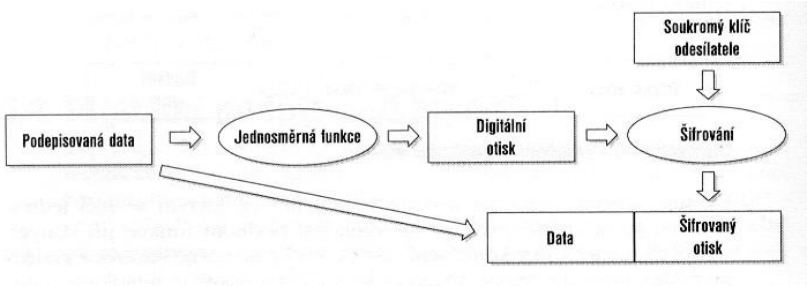
obr. 18 – schéma funkce programu PGP (zdroj: [4])

[86] [4] s. 111

4.2 Elektronický podpis

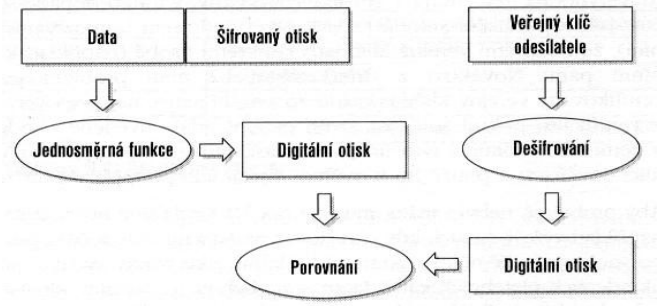
K zajištění nepopiratelnosti a integrity zpráv slouží elektronický (digitální) podpis. Využívá asymetrické kryptografie k zajištění bezpečné komunikace mezi oběma stranami.

Odesílatel vytvoří zprávu, z níž provede výtah (otisk, haš) za použití hašovacího algoritmu. Výtah zašifruje svým privátním klíčem, přiloží na konec zprávy a odešle. Zašifrovanému otisku zprávy se říká elektronický podpis.



obr. 19 – tvorba elektronického podpisu (zdroj: [4])

Příjemce nejprve z obdržené zprávy znovu vypočítá haš. Pak dešifruje odesílatelovým veřejným klíčem jeho podpis a získanou hodnotu porovná s vypočítaným výtahem. Pokud se obě shodují, znamená to, že zpráva nebyla po cestě změněna a elektronický podpis nebyl podvržen.



obr. 20 – ověření elektronického podpisu (zdroj: [4])

Může nastat situace, kdy útočník napadne databázi klíčů a přidá do ní podvržený klíčový pár. Abychom poznali, že obdrženému veřejnému klíči můžeme věřit, potřebujeme pomoc certifikační autority.

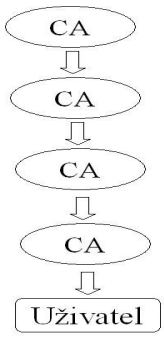
4.2.1 Certifikační autorita (CA)

Jedná se o společnost, která se stará o ověřování pravosti veřejných klíčů. Kdokoli chce ověřit pravost svého klíče, musí se v autoritě zaregistrovat (zadat své osobní údaje). CA pak svým elektronickým podpisem stvrdí pravost jeho klíče.

Příjemce klíče tedy musí nejprve ověřit pravost podpisu autority, osobní údaje registrované osoby a teprve pak může přistoupit k samotnému ověření podpisu zprávy.

Pokud dojde ke kompromitaci soukromého klíče, musí to jeho majitel nahlásit autoritě, která sestavuje tzv. *Certificate Revocation List* – seznam odvolaných certifikátů. Příjemce zprávy tedy musí vždy ještě zkontrolovat, zda se certifikát nenachází v tomto seznamu.

Samostatnou kapitolou je ověřování certifikátů certifikačních autorit. Nad autoritou totiž bývá další autorita, která dodává důvěryhodnost všem CA pod sebou. Tomuto způsobu ověřování se říká strom certifikačních autorit. Celá problematika certifikátů, autorit a veřejných klíčů se nazývá PKI – Public Key Infrastructure [4].



obr. 21 – strom certifikačních autorit

4.2.2 Certifikační autority v ČR

V České republice v současnosti působí tři akreditované certifikační autority.[13]

- 1. Certifikační autorita, a.s.
- Česká pošta, s.p.
- eIdentity, a.s.

Akreditované CA mají povolení vydávat certifikáty, které jsou nutné pro komunikaci se státní správou.

4.2.3 Elektronické časové razítko

„Kvalifikovaným časovým razítkem je datová zpráva, kterou vydal kvalifikovaný poskytovatel certifikačních služeb a která důvěryhodným způsobem spojuje data v elektronické podobě s časovým okamžikem, a zaručuje, že uvedená data v elektronické podobě existovala před daným časovým okamžikem.“[87]

Podobně jako jsou běžné tištěné smlouvy a jiné důležité dokumenty opatřeny kromě podpisu zúčastněných stran i datem podpisu, je dobré mít také elektronické dokumenty opatřeny časovým údajem.

„Dokumenty s dlouhou časovou platností někdy vyžadují ověření elektronického podpisu s časovým odstupem, v době kdy původní certifikáty obvykle ztratily svoji platnost. V okamžiku, kdy nemáme k dispozici údaj o datu vzniku dokumentu, nelze jednoduchým porovnáním s dobou platnosti certifikátu odvodit regulérnost podpisu dokumentu v době platnosti certifikátu. Východiskem je opětovné podepsání dokumentu a zároveň získat nový platný certifikát, tato cesta však předpokládá nahrazení původního dokumentu nově podepsaným. Tento krok není příliš snadný, předpokládá totiž znalost o existenci, počtu a místě uložení všech předchozích kopií a to i s návazností na vlastníky těchto kopií.

[87] [59]

S obdobnými obtížemi je možné se setkat v případě zničení, ztráty nebo ukradení soukromého klíče podepisující osoby, kdy je tedy certifikát pokládán za neplatný a certifikační autorita ho zařazuje na seznam zneplatněných certifikátů. Jelikož jako v předchozím případě není možné zjistit datum vzniku dokumentu podepsaného zneplatněným certifikátem, není tedy ani možné při ověřování elektronického podpisu vycházet z časových vymezení diskreditace soukromého klíče.“[88]

Řešením výše uvedených problémů je zaslat digitální otisk (haš) dokumentu autoritě časových razítek. Ta jej opatří razítkem, které kromě aktuálního data a času obsahuje také sériové číslo razítka a identifikaci autority, která razítko vydala. Na závěr autorita připojí svůj elektronický podpis a vše odešle zpět žadateli. Tím je zajištěno určení časového okamžiku, před kterým elektronický dokument jistě existoval. Autorita přitom nemůže nijak číst obsah dokumentu, neboť pracuje pouze s jeho digitálním otiskem.

4.3 Praktické využití šifrování a podepisování

V této části bude nejprve popsán postup registrace certifikátu u společnosti CA Czechia a jeho následné využití při šifrování a podepisování e-mailové komunikace. Poté bude popsáno zabezpečení elektronické pošty za využití programu PGP Desktop.

4.3.1 Postup žádosti o certifikát u CA Czechia[89]

Certifikační agentura Czechia nabízí 4 kategorie certifikátů k elektronickému podpisu:

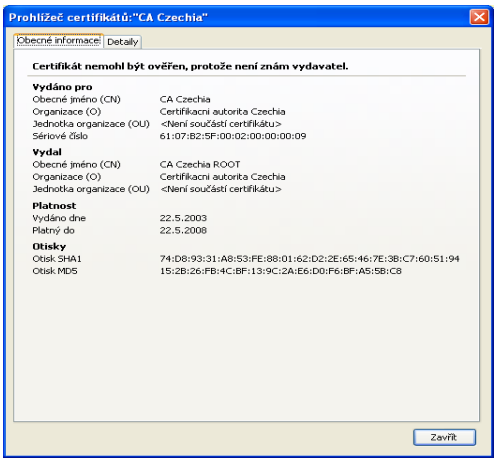
- Testovací
- Osobní
- Firemní
- Serverový

[88] [59] – v textu článku
[89] Zpracováno podle [60]

Žádost o osobní certifikát sestává z osmi kroků, které se provádějí přes Průvodce žádostí o certifikát (viz obr. 23).[11]

1. **Úvodní informace**. Zde jsou popsány podporované internetové prohlížeče a funkce, které v nich musejí být zpřístupněny a zapnuty, aby mohl proces registrace v pořádku proběhnout.

2. **Instalace certifikátů** CA Czechia (viz obr. 22). Nutný krok. Jedině s nainstalovanými certifikáty CA Czechia může uživatel používat k podepisování dokumentů svůj certifikát.



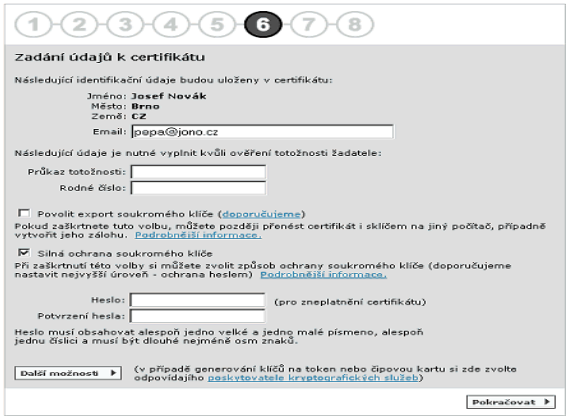
obr. 22 – certifikát CA Czechia

3. **Registrace**. Pokud už má uživatel vytvořený účet u CA Czechia, stačí mu zadat své přihlašovací údaje. V opačném případě musí přikročit k bodu 4.

4. **Zadání osobních údajů**. Uživatel musí zadat platné osobní údaje. Ty budou později ověřeny v kanceláři certifikační autority. Pokud by jakýkoli údaj vyplněný na webu nesouhlasil se skutečností, nebude certifikát vydán. Uživatel také musí zadat heslo, kterým se bude autentizovat do systému CA.

5. **Dokončení registrace**. Systém uživateli přidělí jeho identifikační číslo a vypíše jej na obrazovku společně se zvoleným heslem k přístupu do systému. Oba údaje je nutné si zapamatovat.

6. **Údaje k certifikátu**. Zde se vytvoří samotná žádost o vystavení certifikátu k elektronickému podpisu. Uživatel musí nastavit e-mailovou adresu, ke které bude certifikát používat. Ke každé adrese totiž potřebuje samostatný certifikát. Dále je třeba nastavit vlastnosti certifikátu – zda půjde exportovat na jiný počítač a zda bude pro jeho zabezpečení použita silná ochrana. Ta upozorní uživatele na každou událost, při které je manipulováno s klíčem. Uživatel dále musí vyplnit číslo identifikačního průkazu (občanského průkazu, cestovního pasu), své rodné číslo (oba údaje slouží k jeho identifikaci kanceláří CA) a nakonec heslo pro manipulaci s certifikátem (používá se např. při podání žádosti o zneplatnění certifikátu). Po zadání těchto údajů a jejich odeslání na server dojde k vytvoření žádosti o certifikát, během níž je v počítači uživatele vytvořen klíčový pár pro elektronický podpis.



obr. 23 – průvodce žádostí o certifikát CA Czechia

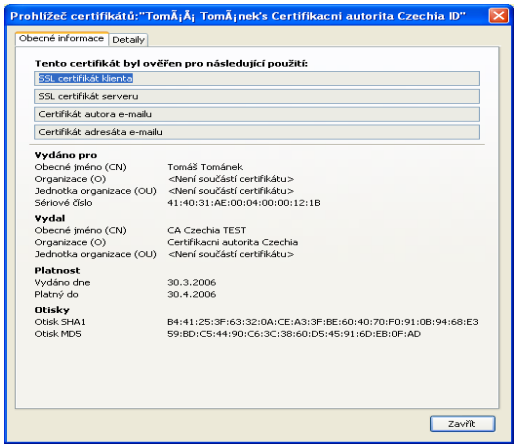
7. **Tisk smlouvy**. Uživatel vyhotoví dvě kopie smlouvy. Jednu kopii opatří úředně ověřeným podpisem a obě odešle do sídla CA Czechia.

8. **Zadání fakturačních údajů**. Uživatel musí vyplnit všechny povinné údaje (označené hvězdičkou). Po jejich odeslání mu systém elektronickou poštou pošle fakturu, kterou je nutné zaplatit. Po jejím zaplacení obdrží uživatel potvrzenou kopii smlouvy k certifikátu.

9. **Dokončení žádosti**. V tomto závěrečném kroku systém odešle na uživatelovu e-mailovou adresu zprávu o úspěšném dokončení žádosti společně s jejím identifikačním číslem.

4.3.2 Instalace certifikátu

Pokud byly všechny náležitosti registrace provedeny správně, bude uživateli vystaven certifikát k elektronickému podpisu (obr. 24). Může se k němu přihlásit přes webovou stránku CA Czechia odkazem „Správa certifikátu.“ Nyní si může certifikát nainstalovat do svého systému. Po instalaci se certifikát stane součástí internetového prohlížeče (Internet Explorer, Mozilla Firefox) a poštovního klienta (MS Outlook, Mozilla Thunderbird).

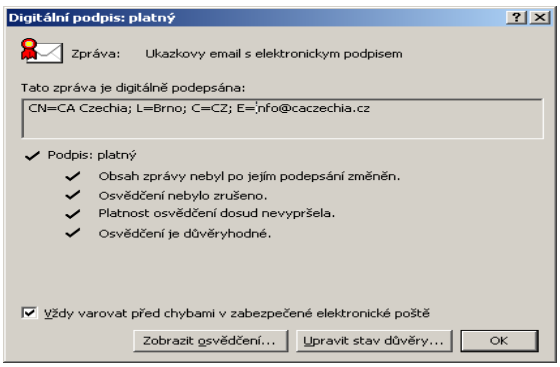


obr. 24 – zkušební certifikát autora této práce

4.3.3 Použití certifikátu v e-mailové komunikaci

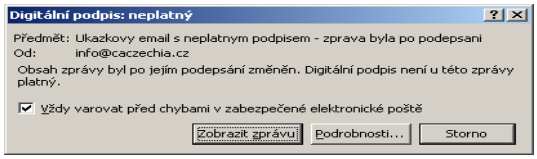
Společnost CA Czechia umožňuje vyzkoušení využití certifikátů zasláním několika různých zkušebních e-mailů na adresu uživatele. Mimo podpisu platného jsou zde i ukázky neplatných podpisů. Několik příkladů je uvedeno níže.

Po obdržení digitálně podepsaného dokumentu si uživatel může prohlédnout tabulku s informacemi o ověření digitálního podpisu (viz obr. 25).



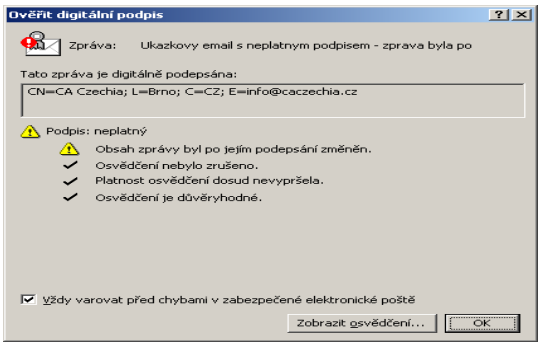
obr. 25 – tabulka platného podpisu

Pokud je podpis neplatný, poštovní klient uživatele na tuto skutečnost upozorní hlášením (viz obr 26.).



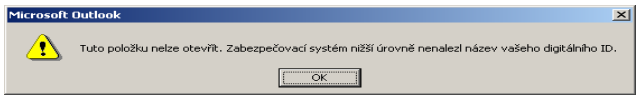
obr. 26 – upozornění na neplatnost el. podpisu

Výskyt chyby při ověřování se v tabulce projeví žlutou výstražnou značkou (viz obr. 27).



obr. 27 – tabulka neplatného podpisu: zpráva byla po podepsání změněna

Může se také stát, že některé zprávy nepůjdou vůbec otevřít. Tato situace nastane v případě, že klient nenajde v e-mailu elektronický podpis přes to, že by zpráva podle informací ve své hlavičce měla být podepsána. Bývá to způsobeno některými antivirovými programy a firewally, které na konec zprávy automaticky doplňují určité texty (viz obr. 28). [2]



obr. 28 – ukázka chybového hlášení při nemožnosti nalézt digitální podpis

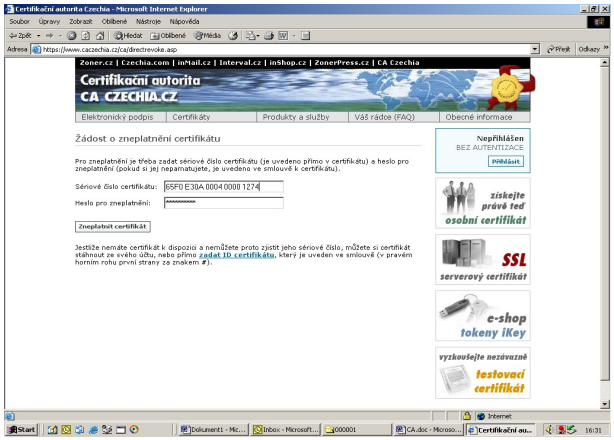
Poté, co zde byly uvedeny příklady situací, které mohou nastat při příjmu digitálně podepsaných zpráv, zbývá uvést ještě postup tvorby a odeslání podepsaného e-mailu. V podstatě je postup stejný jako u běžné e-mailové komunikace. Jen je třeba před samotným odesláním zprávy volbou ikony „Možnosti“ zobrazit tabulku, ve které pak uživatel musí zatrhnout políčko „Přidat digitální podpis do odesílané zprávy.“

Ve stejné tabulce se dá také nastavit šifrování zprávy zatrhnutím políčka „Zašifrovat obsah zprávy a přílohy.“ Pokud uživatel hodlá zabezpečovat všechny odesílané zprávy a nechce vše u každé z nich nastavovat „ručně“, stačí, když v hlavním menu „Nástroje“ vybere „Možnosti“ a následně na kartě „Zabezpečení“ zatrhne pod položkou „Zabezpečená elektronická pošta“ políčka podle toho, zda chce zprávu jen podepsat, jen zašifrovat nebo obojí najednou.

Zašifrovat zprávu může uživatel jen pro příjemce, od kterých předem získal jejich certifikát a má ho ověřený. Ověřit jej může u certifikační autority, která certifikát vydala. Je zde také možnost zvolit, aby systém certifikátu důvěřoval i bez ověření jeho platnosti. To se ale příliš nedoporučuje.

4.3.4 Zneplatnění certifikátu

Může se stát, že je certifikát kompromitován. Například dojde k ukradení počítače, na kterém je nainstalován, a hrozí, že se nepovolaná osoba bude v elektronické komunikaci vydávat za majitele certifikátu. V takovém případě je třeba certifikát zneplatnit (viz obr. 29) [2].

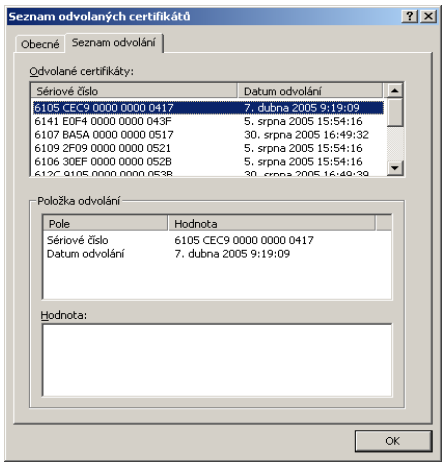


obr. 29 – formulář CA Czechia pro odvolání certifikátu

Společnost CA Czechia nabízí na svém webu formulář pro zneplatnění certifikátu. Sem je třeba zadat sériové číslo certifikátu a heslo. Autentizace heslem je nutná, aby nikdo neoprávněný nemohl zrušit uživatelův certifikát.

Platnost certifikátu je zrušena nejdéle do 24 hodin. Do této doby musí být umístěn na seznam zneplatněných certifikátů (Certificate Revocation List, CRL). Zde se nachází každý certifikát dané autority, jehož účinnost byla zrušena (viz obr. 30).

CRL je třeba mít nainstalován v systému, aby se z něj čerpaly informace potřebné k ověření platnosti certifikátů osob, s nimiž uživatel komunikuje. Nikdo nesmí mít možnost použít k digitálnímu podepsání nebo šifrování elektronické zprávy neplatný certifikát.

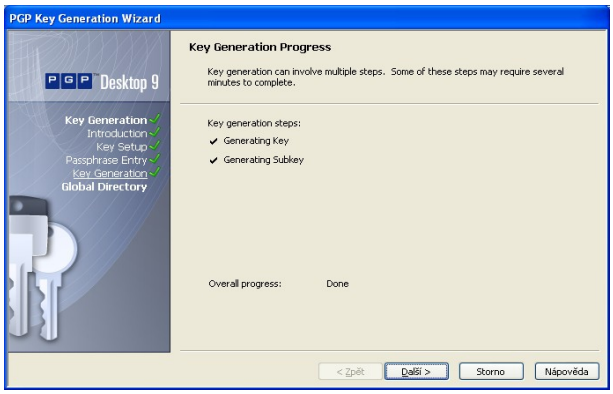


obr. 30 – seznam odvolaných certifikátů (CRL)

4.4 Šifrování a podepisování v programu PGP Desktop

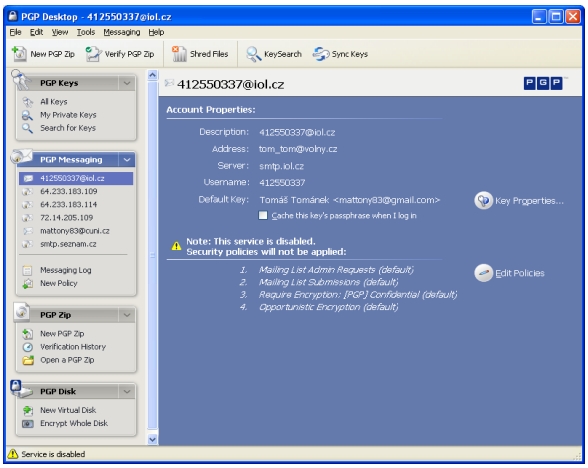
Program PGP Desktop je dalším nástrojem pro zabezpečení dat. Umožňuje tvorbu a správu klíčů. Šifrování jednotlivých souborů, celého disku, e-mailové komunikace i instant messagingu.

Tvorba nového klíče se děje přes „Průvodce tvorbou klíče PGP“, který se spouští v menu „File.“ V průvodci je třeba, aby uživatel vyplnil své celé jméno a e-mailovou adresu (adresy), pro které chce vystavit šifrovací klíč. Poté zadá svou vstupní frázi (passphrase), která se používá při manipulaci s klíči. Program nyní vygeneruje klíčový pár (viz obr. 31) a umístí jej do Globálního adresáře PGP (PGP Global Directory), který se nachází na internetu a slouží k zjišťování a ověřování veřejných klíčů uživatelů PGP. Proceduru je třeba dokončit na odkaze, který přijde do e-mailové schránky, zadané uživatelem při registraci. Po kontrole zadaných údajů a jejich potvrzení je registrace kompletní. Uživatel si ještě musí nainstalovat ověřovací klíč PGP Directory, aby mohl důvěřovat klíčům ověřeným tímto adresářem.



obr. 31 – generování klíče v PGP Desktop

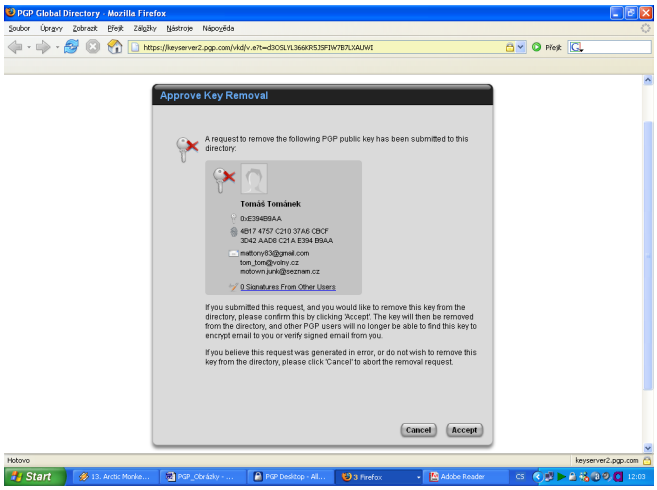
Pokud chce uživatel použít PGP Desktop k šifrování a elektronickému podepisování e-mailové komunikace, stačí, když vytvoří standardním způsobem zprávu. Po zadání příkazu k jejímu odeslání se objeví dialogové okno PGP, kde se ho program dotáže, zda si přeje tento účet zabezpečit (viz obr. 32). Po potvrzení volby a následném zadání vstupní fráze a výběru klíče, který se má pro zabezpečení použít, je už veškerá další komunikace z tohoto uživatelského účtu automaticky šifrována a podepisována (viz obr. 34).



obr. 32 – nastavení zabezpečení e-mailového účtu v PGP Desktop

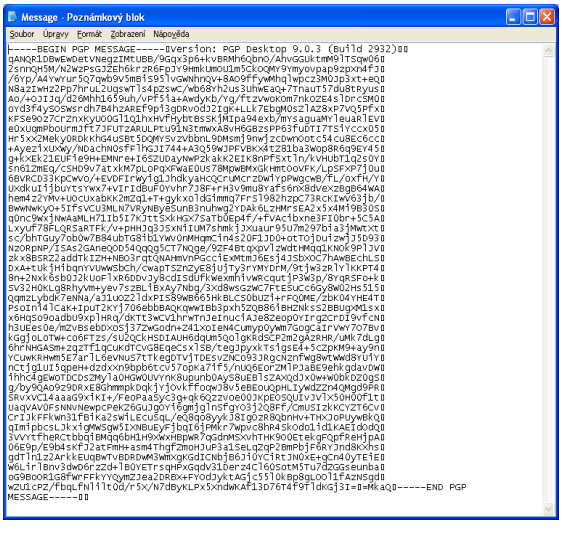
Může se stát, že uživatel chce z nějakého důvodu svůj klíč zneplatnit. Například proto, že je zde reálná hrozba kompromitace klíče. Pro tento případ má PGP Desktop funkci „Revoke“. Po zadání vstupní fráze je klíč zneplatněn a vyřazen z internetové databáze klíčů. Pokud uživatel zapomene svou vstupní frázi, je možné platnost klíče zrušit přes PGP Global

Directory (viz obr. 33). Zde je třeba zadat e-mailovou adresu, na kterou bude zaslána žádost o potvrzení volby, podobně jako tomu bylo při registraci. Při přechodu na zaslaný odkaz stačí již jen definitivně potvrdit volbu a klíč je zneplatněn.



obr. 33 – žádost o potvrzení zrušení klíče v PGP Global Directory

Používání programu PGP Desktop je díky přehlednému uživatelskému rozhraní příjemné a nevyžaduje po běžném uživateli přehnané technické znalosti.



obr. 34 – zpráva zašifrovaná v PGP (Aplikace *Poznámkový blok*)

4.5 TrueCrypt

TrueCrypt je open-source aplikace, která dokáže vytvářet šifrované diskové oddíly na platformách Windows i Linux. Umožňuje též sdílení zašifrovaných diskových oddílů mezi oběma platformami. TrueCrypt umí šifrovat celý diskový oddíl takzvaně „za pochodu“ (anglicky „on-the fly“). Souborový systém může být uložen na pevném disku nebo v jediném souboru, který se označuje jako tzv. kontejner. Při čtení ze zabezpečeného virtuálního disku dochází k průběžnému dešifrování dat. Při zápisu jsou data ihned šifrována. Celý proces probíhá automaticky, bez dalších zásahů uživatele a je tedy vhodný i pro počítačové laiky. TrueCrypt poskytuje velké množství šifrovacích a hašovacích algoritmů (včetně SHA-1 a Whirlpool).

Přední český kryptolog Vlastimil Klíma k bezpečnosti programu poznamenává: „Zkontrolovali jsme, že zdrojový kód TrueCrypt odráží to, co je pro tento případ popsáno v dokumentaci, a neobsahuje žádná zadní vrátka.“[90]

TrueCrypt umožňuje šifrovat data na disku nejznámějšími algoritmy současnosti a dokonce i jejich zřetězením (viz. tabulka):

Podporované algoritmy

Šifra (E)	**Délka Primárního klíče (K1) v bitech**
AES	256
Blowfish	448
CAST5 (CAST-128)	128
Serpent	256
Triple DES	3 x 56
Twofish	256
AES+Twofish	256+256
AES+Twofish+Serpent	256+256+256
Serpent+AES	256+256
Serpent+Twofish+AES	256+256+256
Twofish+Serpent	256+256

zdroj tabulky: [61]

[90] [61]

Jak známo, blokové šifry mohou pracovat v několika módech. Aplikace TrueCrypt používá nový mód, známý jako LRW (pojmenovaný po svých tvůrcích Liskov-Rivest-Wagner). Tento mód používá k šifrování bloků na různých pozicích otevřeného textu stejnou blokovou šifru, ovšem s různými klíči.

Samotné šifrování probíhá následovně:

„Označme E zvolenou blokovou šifru (jednoduchou nebo zřetězení dvou nebo tří) a její klíč K1. Klíč K1 nazýváme *Primární klíč* a jeho délku ukazuje tabulka. LRW používá navíc tzv. *Sekundární klíč* K2, který má 128 bitů. Právě klíč K2 se modifikuje podle pozice otevřeného textu. Otevřený text (pro blokové šifry s délkou bloku 128 bitů) si rozdělíme na 128 bitové bloky Pi, kde i je 128 bitový index bloku. Pro šifrování bloku otevřeného textu Pi se nepoužije K2, ale nějaká hodnota K2(i), která vznikne modifikací K2 indexem i. Právě hledání vhodné modifikační funkce, která by byla rychlá, vedlo k návrhu obecnější třídy blokových šifer, tzv. *tweakable block ciphers*, kde tweak – zde K2(i) – modifikuje blokovou šifru stejně kvalitně jako klíč, ale je mnohem snadněji měnitelný než klíč. Šifruje se jednoduše takto: Ci = EK1(Pi ^ K2(i)) ^ K2(i), kde Ci je vzniklý blok šifrového textu.

Jinými slovy, na blok otevřeného textu se přičte modifikovaný *Sekundární klíč*, výsledek se zašifruje blokovou šifrou s *Primárním klíčem* a na závěr se opět přičte modifikovaný *Sekundární klíč*. Dodejme, že modifikační funkce je násobení dvou 128 bitových hodnot (jakožto polynomů) v jistém Galoisově tělese GF(2128), stručně řečeno je to (jednoduchá) lineární funkce, závislá na klíči.

Jakmile vytváříme nový disk a zvolíme si blokovou šifru E z nabídky, jsou klíče K1 a K2 vytvořeny generátorem náhodných znaků (RNG) TrueCryptu a uloženy do hlavičky disku. A hlavička je zašifrována pomocí náhodné soli a passwordu uživatele.“[91]

Nejdůležitějším prvkem celého procesu šifrování je volba dostatečně silného hesla. Nejlépe náhodné směsi čísel, písmen a jiných znaků, která se nedá odhalit běžným slovníkovým útokem. TrueCrypt umožňuje zvolit libovolný soubor uložený na disku, jako tzv. *klíčový soubor*, jehož obsahem se ještě dále modifikuje heslo. TrueCrypt však „tento

[91] [61]

soubor zpracovává jednoduchoučkou funkcí, která je ve srovnání s tisícinásobným hašováním velmi legrační. Pokud máte kvalitní password, nemusíte tento klíčový soubor používat.“[92]

Závěrem se dá napsat, že TrueCrypt je velice spolehlivým nástrojem pro ochranu citlivých dat na disku. Zvlášť, pokud je uživatel nepřetržitě připojen k internetu.

[92] [61]

5 Zabezpečení mobilní sítě GSM – případová studie

Následující případová studie si klade za cíl kompletně popsat mobilní systém GSM s jeho historií, funkcí, architekturou, ale hlavně autentizací a zabezpečením. Mobilní telefony se staly zcela běžnými nástroji každodenní komunikace. Jejich počet neustále stoupá a s ním se zvyšuje i množství jimi nabízených funkcí. Samozřejmostí je dnes přístup k síti internet přes GPRS nebo EDGE. Zabezpečení GSM sítí přitom není zdaleka tak dobré, jak by se slušelo. O tom více pojednává sekce věnující se šifrovacímu algoritmu A5 a možnostem jeho prolomení. Rád bych na tomto místě poděkoval panu Mgr. Pavlu Vondruškovi z Odboru bezpečnosti společnosti Telefónica O2 Czech Republic, který mi poskytl mnoho cenných materiálů k vytvoření této studie.

5.1 GSM [93]

„GSM (Globální Systém pro Mobilní komunikaci) je nejoblíbenější standard pro mobilní telefony na světě. GSM telefony používá přes miliardu lidí z více než 200 zemí.“[94] K hlasové komunikaci nejstarších GSM telefonů byla poměrně záhy přidána datová komunikace (komunikační kanály jsou digitální). Paketové přenosy jsou označeny zkratkou GPRS. V současnosti se čím dál více rozšiřují mobilní sítě třetí generace (3G), které podporují vyšší rychlosti přenosu dat ve standardech označených EDGE a UMTS.

Historie mobilních telefonů sahá do počátku 80. let 20. století. V roce 1982 byla založena skupina GSM (Groupe Spécial Mobile). Ta zahájila práci na systému mobilní komunikace, jehož technické základy byly položeny o 5 let později. Od roku 1998 existuje projekt pro mobilní sítě třetí generace (3GPP).

„GSM je buňková síť, což znamená že mobilní telefony se připojují do sítě prostřednictvím nejbližší buňky. GSM síť funguje na několika radiových frekvencích.

[93] podle [62]
[94] [62]

Existují čtyři různé velikosti buněk - makro, mikro, piko a *deštníkové buňky*. Oblast pokrytí každé buňky se liší podle prostředí. Za makro buňky jsou považovány ty, kde je umístěna anténa základové stanice na stožáru nebo na budově nad úrovní střech. Mikro buňky mají anténu umístěnou pod úrovní střech; typické je použití v zastavěných oblastech. Pikobuňky jsou malé buňky s průměrem pár desítek metrů; používají se hlavně uvnitř budov. Na druhou stranu *deštníkové buňky* se používají pro pokrytí oblastí ve stínech a na vyplnění mezer mezi buňkami.

Velikost pokrytí záleží na výšce antény, výkonu antény a na podmínkách šíření a pohybuje se od několika stovek metrů až do desítky kilometrů. Největší vzdálenost, které se podle specifikace GSM prakticky používá, je 35 km. Existuje však koncept rozšířené buňky, kde může být oblast dvojnásobná i větší.“[95]

Zabezpečení GSM obstarávají šifrovací algoritmy. Pro vzdušný přenos dat se využívá algoritmů A5/1 a A5/2. První z šifer je silnější, druhá slabší. V obou byly nalezeny vážné slabiny. Proto se doporučuje oba algoritmy nahradit silnějšími. Za tímto účelem byl schválen algoritmus A5/3, známý také pod názvem KASUMI.

Architektura GSM[96]

GSM se skládá ze dvou hlavních částí: pevné infrastruktury (sítě) a mobilních stanic. Mobilní stanice sestává z mobilního přístroje a modulu SIM.

Síť

Fixní infrastrukturu GSM lze rozdělit do třech podsystémů:

- subsystém základových stanic (BSS – Base Station Subsystem)
- subsystém přepínání a řízení (SMSS – Switching and Management Subsystem)
- subsystém chodu a údržby (OMSS – Operation and Maintenance Subsystem)

[95] [62]
[96] podle [63]

BSS
Subsystém základových stanic se dělí na:

1) BTS (Base Transceiver Station) – poskytuje rádiové pokrytí buňky; rádiové kanály pro vysílání a uživatelský datový provoz v buňce
2) BSC (Base Station Controller) – řídí jednotlivé BTS

SMSS

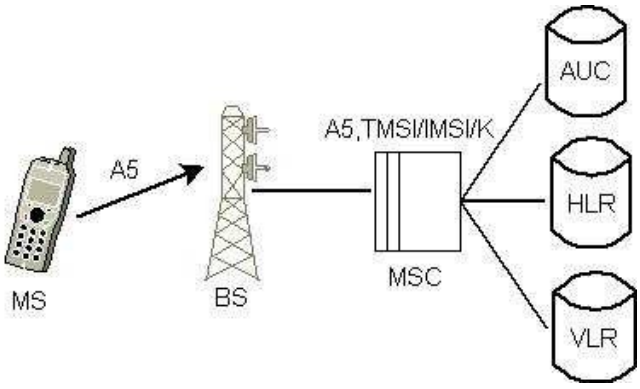
Subsystém přepínání a řízení tvoří:

- MSC (Mobile Switching Centers) – mobilní přepínací centra; fungují jako digitální ústředna
- HLR (Home Location Register) – databáze uživatelských dat všech stálých uživatelů sítě; jsou zde uložena autentifikační data a čísla IMSI (International Mobile Subscriber Identity)
- VLR (Visitor Location Register) – v této databázi jsou uložena data všech mobilních stanic, které se v současnosti nacházejí v oblasti pokryté příslušným MSC

OMSS

Subsystém chodu a údržby:

- OMC (Operation and Maintenance Center) – centrum chodu a údržby
- AUC (Authentication Center) – centrum autentizace; databáze důvěrných dat a klíčů, která se stará o zabezpečení sítě GSM
- EIR (Equipment Identity Register) – registr identity zařízení; databáze sériových čísel mobilních zařízení (IMEI), slouží pro identifikaci přístroje v síti, společně s AUC slouží k zabezpečení sítě GSM



obr. 35 - struktura sítě GSM (zdroj: [64])

SIM

„Osobní čipová karta SIM může být pevně instalovaný čip (tzv. plug-in SIM) nebo vyměnitelný SIM modul. SIM je bezpečné mikroprocesorové prostředí implementované na platformě velikosti kreditní karty se zabudovanou stálou pamětí. V GSM jsou používány dva typy karet – ID1 a plug-in karty.“[97]

Na SIM kartě jsou implementovány tři druhy paměti:

1) ROM – pouze pro čtení; obsahuje operační systém, aplikace a bezpečnostní algoritmy A3 a A8
2) RAM – používá se k bufferování přenosových dat a jejich následnému spouštění
3) EEPROM – ukládá data sloužící k identifikaci uživatelů (IMSI, PIN), čísla volání, klíče Ki, síťové informace (TMSI, LAI) a identifikátor mobilního zařízení (IMEI)

SIM v sobě implementuje zabezpečení GSM prostřednictvím uživatelské autentizace v síti, utajení dat během přenosu vzduchem a podmínek přístupu k souborům v mobilním

[97] [63], s. 3 (překlad autor diplomové práce)

zařízení. Jednou z podmínek je heslo PIN. Pokud je zadán třikrát po sobě špatně, SIM karta se zablokuje.

Identifikátory

K zabezpečení GSM slouží poměrně velké množství identifikátorů. Jejich rozdělení je následující:

Identifikace uživatele – IMSI, MSISDN, TMSI, MSRN
Identifikace zařízení – IMEI

Uložení jednotlivých identifikátorů v databázích:

Báze HLR – IMSI, MSISDN, MSRN
Báze VLR – LMSI, MSRN, IMSI, TMSI, MSISDN, LAI
Báze AUC – IMSI, RAND, SRES, Ki, Kc
Báze EIR – IMEI

„Během registrace ke službám operátora mobilních služeb obdrží každý uživatel unikátní identifikátor – IMSI (International Mobile Subscriber Identity). Toto IMSI je uloženo v SIM. Mobilní stanice může fungovat pouze v případě, že je do zařízení vložena SIM s platným IMSI, jelikož je to jediný způsob, jak správně účtovat konkrétního uživatele. IMSI sestává z několika částí: mobilního kódu země (Mobile Country Code, MCC) – číslicový, pro jednoznačnou identifikaci mobilních sítí uvnitř státu; Mobile Subscriber Identification Number (MSIN) - maximálně 10 desítkových číslic, identifikační číslo uživatele v jeho domovské mobilní síti.“[98]

MSISDN

Mobile Subscriber ISDN je skutečným telefonním číslem mobilní stanice.

[98] [63], s. 4 (překlad autor diplomové práce)

TMSI

Temporal Mobile Subscriber Identity je identifikační číslo s lokální platností přidělované VLR spravující danou oblast. TMSI se používá namísto IMSI k identifikaci a adresování mobilní stanice. TMSI je přiděleno a platné pouze v oblasti působnosti konkrétní VLR, při přechodu do oblasti jiné VLR je změněno i TMSI. Takto nikdo nemůže vysledovat identitu uživatele odposlechnutím rádiového kanálu. TMSI je uloženo v paměti SIM karty mobilní stanice. Na straně sítě je uloženo pouze ve VLR a není posíláno HLR. TMSI může sestávat až z 32 bitů. Vztah mezi IMSI a TMSI je zaznamenán ve VLR.[99]

MSRN

Mobile Station Roaming Number je dočasné číslo ISDN. MSRN je závislé na umístění mobilní stanice. Je přidělováno na území VLR.

IMEI

International Mobile Station Equipment Identity je jednoznačným identifikátorem mobilního zařízení ve světě. Je přidělováno výrobcem (jedná se vlastně o druh sériového čísla) a registrováno operátorem sítě. Je uloženo v EIR.

Architektura GPRS[100]

„GPRS jako nová síťová služba používá techniku paketů k přenosu vysokorychlostních i nízkorychlostních dat a k efektivnímu vysílání. GPRS optimalizuje využití sítě a rádiové zdroje. Striktní separace mezi rádiovým a síťovým subsystémem dovoluje další využití subsystému sítě jinými rádiovými technologiemi. GPRS nevyžaduje změny již nainstalovaných základen MSC.“[101]

[99] [63], s. 4
[100] podle [63], s. 4
[101] [63], s. 4 (překlad autor diplomové práce)

GPRS se skládá ze šesti prvků:

- SGSN – doručování paketů z a do mobilní stanice uvnitř zóny, kterou SGSN spravuje, komunikace s GGSN, sledování umístění jednotlivých mobilních stanic uvnitř operační oblasti SGSN, zajištění bezpečnostních funkcí a kontroly přístupu
- GGSN – zajišťuje propojení s externími sítěmi (internet, privátní sítě atd.), je propojen se SGSN páteřní sítí založenou na IP adresách, udržuje směrovací informace do SGSN, která obsluhuje danou mobilní stanici
- Páteřní síť
- HLR
- MSC/VLR
- SMS-GSMC

„Bezpečnostní funkcionalita GPRS je ekvivalentní existující bezpečnosti GSM. SGSN provádí autentizaci a nastavuje šifrování založené na stejných algoritmech, klíčích a kritériích jako v GSM. GPRS používá nový šifrovací algoritmus, optimalizovaný pro paketové přenosy dat.“[102]

Telefon využívající služeb GPRS komunikuje se základnovými stanicemi GSM. Činí tak ovšem jiným způsobem než při běžném hlasovém volání. Při GPRS přenosu jsou pakety odesílány od BSS do SGSN. Když mobilní stanice odešle datové pakety, jdou tyto pakety přes SGSN do GGSN. GGSN je následně konvertuje pro přenos přes žádanou vnější síť (internet, privátní síť). V opačném směru jsou pakety směřující z internetu do mobilní stanice přijaty GGSN, odeslány do SGSN a teprve odtud do mobilní stanice.

5.2 Zabezpečení GSM [103]

Utajení identity uživatele

„Účelem této funkce je zamezit útočníkovi v identifikaci uživatele zachycením signálních výměn během přenosu rádiovým signálem. Toho může být dosaženo ochranou

[102] [63], s. 5 (překlad autor diplomové práce)
[103] podle [63]

uživatelova IMSI a všech signálních informačních elementů. Z tohoto důvodu by měla být namísto IMSI použita zabezpečená metoda identifikace mobilního uživatele. Signální informační elementy, které nesou informace o mobilním uživateli, musí být přenášeny v šifrované podobě. A šifrovací metoda je také použita.“[104]

Identifikace uživatele probíhá následovně:

Je použit dočasný identifikátor TMSI, platný pouze v dané lokalitě. TMSI musí být používán společně s identifikátorem LAI kvůli zamezení nejednoznačnostem v identifikaci. Vztah mezi příslušným TMSI a IMSI je definován v databázi VLR. Při každé změně lokace musí být nejprve z databáze VLR odstraněno stávající TMSI. Následně je přiděleno nové, platné pro aktuální lokaci. Nové TMSI je do mobilní stanice odesláno v šifrované podobě. Stanice uloží aktuální TMSI společně s LAI do své paměti, čímž zamezí ztrátě identifikátorů při vypnutí přístroje.

Autentizace identity uživatele

Tato funkce je spuštěna sítí, pokud nastane některá z následujících situací:

- Uživatel zažádá o změnu uživatelského informačního elementu v databázi VLR nebo HLR. Tento element obsahuje aktualizaci umístění mobilní stanice (zahrnující změnu databáze VLR), registraci nebo vymazání dodatkových služeb.
- Přístupy k uživatelským službám.
- První přístup uživatele do sítě po restartu MSC/VLR.
- Neshodnost čísla sekvence šifrovacího klíče.

Při první registraci uživatele do domovské sítě je k jeho IMSI přidělen autentizační klíč (Subscriber Authentication Key) Ki, čímž je umožněna jeho autentizace. Uživatel má Ki uložen v SIM svého mobilního přístroje. Na straně sítě je Ki uložen v AUC.

[104] [63], s. 6 (překlad autor diplomové práce)

„Autentizační procedura je založena na algoritmu A3. Algoritmus A3 je implementován jak na straně sítě, tak na straně mobilní stanice. Tento algoritmus na každé straně nezávisle spočítá Signature Response (SRES) z Ki a náhodného čísla (RAND) nabídnutého sítí (t.j. SRES = A3(Ki, RAND)). Ki a IMSI jsou přiděleny při registraci. Mobilní stanice odešle svou hodnotu SRES síti, která ji porovná s vypočítanou hodnotou. Pokud souhlasí obě hodnoty, je autentizace úspěšná. Každé spuštění algoritmu A3 probíhá s novou hodnotou RAND, která nemůže být předem odhadnuta; to zamezuje, aby zaznamenání přenosu v kanálu a přehrání takového záznamu mohlo být použito k vytvoření falešné identity.“[105]

Důvěrnost v GSM

Během spojení je třeba chránit některé uživatelské signální elementy (IMEI, IMSI). Chráněny by měly být i přenosy hlasu a textových zpráv. K tomu se používá symetrická proudová šifra A5.

Mobilní stanice a síť se dohodnou na klíči Kc, který je použit při šifrování a dešifrování algoritmem A5. Nastavení klíče probíhá na nešifrovaném dedikovaném kanálu (DCCH) hned po identifikaci uživatele.

„Přenos Kc do mobilní stanice je nepřímý. Kc je generován na obou stranách za použití algoritmu na generování klíče A8 a RAND autentizačního procesu. Na straně sítě jsou hodnoty Kc spočítány v AUC/HLR simultánně s hodnotami pro SRES. Na straně mobilní stanice je Kc uložen, dokud není aktualizován během příští autentizace.“[106]

Signální a uživatelská data jsou šifrována jak v mobilní stanici, tak v BSS. Proces probíhá následovně:

- BSS vyzve mobilní stanici k zahájení šifrovacího/dešifrovacího procesu
- BSS zahájí svůj vlastní dešifrovací proces
- Mobilní stanice zahájí šifrování a dešifrování

[105] [63], s. 8 (překlad autor diplomové práce)
[106] [63], s. 10 (překlad autor diplomové práce)

- První korektně šifrovaná zpráva od MS, která dorazí do BSS, zahájí šifrovací proces na straně sítě

„Šifrovací proud na jedné straně a dešifrovací proud na straně druhé musí být synchronizovány.“[107]

5.3 Zabezpečení GPRS [108]

Zabezpečovací procedury v GPRS jsou obdobné jako v GSM. Přesto se jisté rozdíly najdou.

Utajení identity uživatele

Identifikační metoda je podobná metodě v GSM. Rozdíl je v tom, že mobilní stanice posílá do SGSN dočasný identifikátor TLLI (Temporary Logical Link Identity) a RAI (Routing Area Identity). TLLI je použit (podobně jako TMSI v GSM) k identifikaci uživatele během přenosu. Celou proceduru zajišťuje SGSN. Během aktualizace lokace mobilní stanice je aktualizována i její směrovací oblast (Routing Area, RA).

„TLLI je pouze místní číslo. Má význam jen v dané RA. TLLI musí být doplněno ještě RAI, abychom zamezili nejednoznačnostem v identifikaci. SGSN má k dispozici příslušné databáze, které definují vztah mezi TLLI a IMSI místo VLR v síti GSM. Vztah mezi TLLI a IMSI je znám pouze mobilní stanici a SGSN. Pokud TLLI a RAI nekorespondují s aktuálním SGSN, je po SGSN spravujícím danou oblast požadováno IMSI mobilní stanice. Není-li známa adresa SGSN, je IMSI požadováno přímo od mobilní stanice. Nové TLLI je přiděleno při každé změně směrovací oblasti.“[109]

[107] [63], s. 10 (překlad autor diplomové práce)
[108] podle [63]
[109] [63], s. 7 (překlad autor diplomové práce)

Autentizace identity uživatele

Autentizace probíhá podobně jako v GSM. Rozdílem je spouštění procedur v SGSM.

Důvěrnost v GPRS

Zatímco síť GSM šifruje přenos mezi mobilní stanicí a BTS, v případě GPRS probíhá šifrování mezi mobilní stanicí a SGSM. Je použit nový šifrovací algoritmus GPRS-A5. Šifrování se děje ve vrstvě Logical Link Control (LLC). Klíč GPRS-Kc je řízen SGSN nezávisle na MSC.

5.4 Algoritmus A5/1 [110]

A5/1 je proudová šifra vyvinutá v roce 1987 v USA. Slouží k zajištění důvěrnosti a bezpečnosti hovoru při přenosu v síti GSM. Struktura algoritmu byla dlouho držena v tajnosti. Postupem času však byla metodou reverzního inženýrství rekonstruována.

Existuje též oslabená verze algoritmu, která se nazývá A5/2. Ta se zpočátku používala v zemích východní Evropy a mimo Evropu. Dnes je stále častěji nahrazována verzí A5/1. Zcela nezabezpečený komunikační algoritmus nese označení A5/0. Je nasazen v některých státech Afriky a Asie.

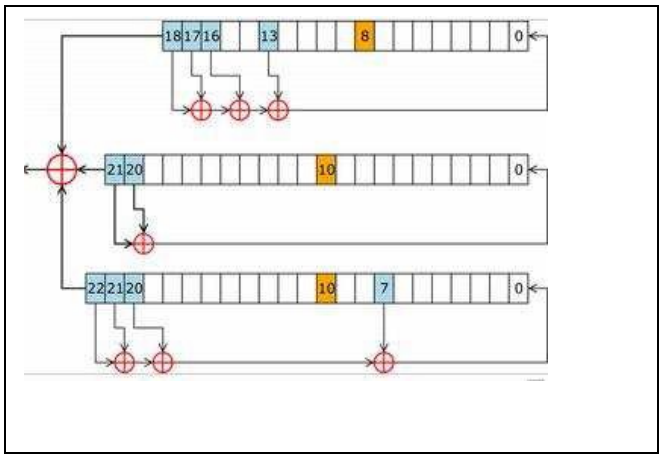
Popis algoritmu A5/1

„A5/1 je proudová šifra, která využívá zpětnou vazbu, konkrétně 3 registry (LFSR). Posuvný (“klokovací”) bit má vliv na všechny 3 registry a je označen na obrázku oranžově. Přenos dat GSM je organizován po sekvencích, tzv. *bursts (shlucích)*.

Typicky je pro přenos v jednom směru a jedním kanálem používána sekvenčně (shluk), který nese informaci 114 bitů a je odesílán každých 4.615 milisekund. Šifra musí zajistit ochranu pro komunikaci v obou směrech.

[110] podle Crypto-World 2/2008

A5/1 je používána tak, že v tomto čase vyprodukuje bity, které jsou se shlukem téže délky sečteny pomocí XOR (pro dva kanály je tedy potřeba, aby proudová šifra vyprodukovala 2*114 bitů). A5/1 je inicializována klíčem délky 64-bitů a pomocí 22-bitového čísla rámce (TDMA). V GSM byla přibližně do roku 2001 použita implementace, kde prvních 10 bitů je dáno pevně a jsou nastaveny na 0. Výsledkem bylo, že klíč A8 má efektivní délku jen 54 bitů. V současné době má klíč A8 efektivní délku celých 64-bitů (algoritmy Comp128v2 a Comp128v3). Šifra A5/1 je tvořena třemi lineárními posuvnými registry R1, R2 a R3 o délkách 19, 22 a 23 bitů se zpětnou vazbou (LSFR), jak je uvedeno na následujícím obrázku.“[111]



obr. 36 - (zdroj: [64])

„Tyto tři registry lze algebraicky popsat takto:

Bity se číslují od nejméně významného bitu registru, tento bit je označen jako 0. Pokud tedy označíme bit nejvíce vpravo indexem nula, má registr R1 zpětnovazební bity 18, 17, 16 a 13, pro R2 to jsou bity 21 a 20 a pro registr R3 bity 22, 21, 20 a 7. Prostřední bity registrů (u R1 je to bit 8, u R2 bit 10, u R3 bit 10) jsou určeny pro nelineární krokování a označíme je C1, C2 a C3. Jejich hodnoty určí, který z registrů bude stát a který se posune.“[112]

[111] [64], s. 6
[112] [64], s. 6

Krokování:

„Nejprve se vypočte majoritní (většinová) hodnota C a to takto: C se rovná nule, jsou-li alespoň dvě z hodnot C1, C2 a C3 nuly, jinak se rovná. Proto se C rovná vždy buď dvěma, nebo třem bitům z trojice (C1, C2, C3). Krokování je definováno tak, že příslušný registr Ri se posune, pokud se hodnota jeho řídicího bitu Ci rovná majoritní hodnotě C (v každém kroku se proto posunou buď právě dva, nebo právě tři registry). Pokud posun nastane, je ze stávajícího stavu vypočtena zpětná vazba Z (například u R2 je to hodnota Z2 = $R_2$21 XOR $R_2$20) a ta se plní zprava do registru. Tím se zároveň posunou všechny buňky registru o jednu doleva.“[113]

Šifrování:

Probíhá po ukončení posunu.

- Nejvyšší bity registrů XORovány => heslo
- Heslo se dalším XORem promísí s otevřeným textem => text zašifrován

Počáteční naplnění registrů:

1. Obsahy registrů se vynulují; vypnutí nelineárního řízení
2. Vytvoří se 88bitový proud složený z 64 bitů klíče Kc a 22 bitů rámce TDMA
3. 88 kroků, v kterých se zpětnou vazbou XORuje vždy jeden bit z proudu s nejnižším bitem registru
4. Proud je naplněn do všech registrů
5. Tím se vytvoří tzv. „počáteční stav“ registrů
6. Zapnutí nelineárního řízení
7. Proběhne 328 kroků produkujících heslo
8. Prvních 100 bitů hesla je ignorováno
9. Následujících 228 bitů hesla zašifruje data ve dvou kanálech o délce 2 x 114 bitů

[113] [64], s. 6 - 7

Slabiny algoritmu

A5 je již dlouho odborníky považován za slabý. Webová prezentace skupiny A5 Cracking[114] uvádí jako hlavní chyby tyto:

1. Příliš malé registry.
2. Lineární posuvné registry se zpětnou vazbou nemíchají výsledky mezi sebou
3. Implementace protokolu je chybná. – Útočník může zaznamenat veškerý šifrovaný provoz, jakmile se jednou dostane k SIM kartě uživatele. Takto může dešifrovat jakoukoli GSM konverzaci, která v minulosti proběhla.
4. Všechny trap registry jsou na jedné straně.

Pokusy o prolomení algoritmu[115]

V roce 1997 představil Golič tzv. útok pomocí předvypočítaných stavů. Složitost jeho řešení byla v té době $2^{46.16}$.

O tři roky později přišli Biryukov, Shamir a Wagner s útokem time-memory tradeoff, který umožňoval teoretické prolomení algoritmu v reálném čase. Vondruška k tomu poznamenává:

„Důležitým momentem tohoto útoku je přípravná fáze**,** během které se vytvoří tabulka obsahující 2^{35} stavů automatu A5/1, která bude během lušticího procesu používána k určení vnitřních stavů. Autorům se podařilo vyvinout metodu, díky níž jsou schopni jednotlivé stavy kódovat pomocí 40bitových řetězců. Výsledná kapacita nutná pro uložení zmíněné tabulky tedy činí zhruba 146 GB. Přípravná fáze je náročná nejen na paměť, ale i na čas, neboť pro zkonstruování uvedené tabulky je třeba 2^{38} až 2^{48} operací. Vzhledem k těmto nárokům se přípravná fáze stává vzhledem k potřebným systémovým zdrojům nejnáročnějším krokem celé metody. Velmi závažné ovšem je, že výsledek této fáze je použitelný k útoku na A5/1 opakovaně kdekoliv a kdykoliv na světě (nezávisí na síti GSM operátora, jazyku apod.). Lze

[114] [65]
[115] podle [64], s. 7 - 9

dokonce očekávat, že zmíněné tabulky naplněné potřebnými informacemi se mohou stát „obchodním artiklem“….“[116]

Ve stejném roce Biham s Dunkelmanem zveřejnili vylepšený útok se složitostí $2^{39.91}$.

Dvojice Ekdahl – Johansson v roce 2003 představila veřejnosti útok na inicializační stav algoritmu. Lze jej provést po zachycení dvou až pěti minut známé konverzace. Na rozdíl od útoku skupiny Biryukov, Shamir a Wagner nepotřebuje tento žádnou přípravu ani uložení předvypočítaných stavů. Útok byl dále vylepšován. Maximov a kolektiv dokázali metodu zdokonalit tak, že byla schopna prolomit algoritmus za jednu minutu se znalostí pouhých několika vteřin známé konverzace.

Mimo výše popsaných útoků, které se řadí mezi pasivní, existují i aktivní útoky. Jedním z nich je způsob publikovaný Barkanem v roce 2004. Jedná se o útok ze středu. Mezi BTS a mobilním telefonem (mobilní stanicí) uživatele se umístí zařízení, pomocí něhož útočník vypne šifrování algoritmem A5/1 a nahradí jej slabší verzí A5/2. Klíč používaný v obou algoritmech je stejný. Útočník pak s mobilní stanicí komunikuje přes algoritmus A5/2 a s BTS přes A5/1. Data z mobilní stanice dokáže „muž ve středu“ dešifrovat, odposlechnout, následně zašifrovat silnějším algoritmem a odeslat BTS, jako by k žádnému odposlechu nedošlo.

Hultonův a Millerův útok

Nejnověji oznámený útok na šifrovací algoritmus A5 může v blízké době znamenat konec této formy zabezpečení mobilních sítí GSM (pro upřesnění dodávám, že tato kapitola diplomové práce vzniká v březnu 2008). David Hulton a Steve Miller oznámili přípravu efektivního pasivního útoku s využitím předem vypočítaných duhových tabulek, který se vyznačuje opravdu nízkou cenou a vysokou rychlostí.

[116] [64], s. 7

V současné době probíhá generování duhových tabulek, které budou ihned po dokončení dány veřejně k dispozici. Společně s tabulkami Hulton a Miller popsali i metodu útoku na síť GSM. Snahou je poukázat na špatné zabezpečení této sítě.

Duhová tabulka

Duhová tabulka (neboli rainbow table) je vyhledávací tabulka, která nabízí tzv. time-memory trade off.

„Duhová tabulka je kompaktní reprezentací souvisejících sekvencí (neboli řetězců) nezašifrovaného hesla. Každý řetězec začíná iniciačním heslem, které prochází skrze hašovací funkci. Výsledný haš je poté poslán do redukční funkce, která vytvoří rozdílné nezašifrované heslo. Proces se pak opakuje pro pevně daný počet iterací. Iniciační heslo a poslední hašová hodnota řetězce tvoří vstup duhové tabulky.“[117]

Obsah tabulky nezávisí na vstupu algoritmu. Jakmile je jednou vytvořen, lze jej v nezměněné podobě opakovaně používat pro vyhledávání.

Time-memory trade off

Prodloužení řetězce způsobí zmenšení velikosti tabulky, ovšem zároveň zvýší množství času, potřebné k provedení iterace nad každým řetězcem. Tomu se říká time-memory trade off. V jednoduchém případě jednopoložkového řetězce je vyhledávání velice rychlé, ale tabulka velmi rozsáhlá. Jakmile se prodlouží řetězec, vyhledávání se zpomalí, ale velikost tabulky se zmenší.[118]

[117] [66] (překlad autor diplomové práce)
[118] podle [66]

Popis útoku[119]

Útok je plně pasivní. Vyžaduje 3-4 rámce konverzace, což umožňuje získat 3 rámce z výstupu šifry A5/1. To znamená možnost vypočítat 204 šedesátičtyřbitových šifrovacích hodnot z různých offsetů uvnitř šifrového proudu. Nyní nastupuje duhová tabulka ke zpětnému výpočtu interních stavů algoritmu.

204 datových bodů je puštěno „skrz“ duhovou tabulku. Tak dostaneme v průměru tři interní stavy šifry A5/1. Tyto stavy následně „nahrajeme“ do algoritmu A5/1 a zpětně vypočítáme stav před přimícháním klíče. Z důvodu majoritního klokování skončíme s několika možnými hodnotami stavu. Po zpětném výpočtu všech tří stavů se podíváme na možné hodnoty stavu. Společná hodnota bude ta, kterou hledáme. Nyní můžeme dešifrovat nebo zašifrovat jakýkoli rámec. Je možné vypočítat aktuální klíč Kc, ale v tomto bodě to není nezbytné.

Použitý hardware

Duhová tabulka je implementována do tzv Field-Programmable Gate Array (FPGA). Tím se velice sníží čas potřebný k výpočtu celé tabulky. Autoři uvádějí, že by výpočet tabulky jinak zabral jednomu PC 33 000 let nebo 33 000 PC jeden rok. S využitím clusteru 68 FPGA je to možné provést za 3 měsíce. Autoři dále uvádějí, že jsou zhruba uprostřed vývoje nového hardwaru, který útok ještě více urychlí.

V současnosti používají hardware s označením PICO E-16. Systém má diskovou kapacitu 6 TB. Finální řešení by mělo být schopno získat klíč GSM konverzace (hlas nebo sms/text) pod 30 minut s 2 TB a jedním FPGA nebo pod minutu s 2 TB flash hard disků a 32 FPGA. Rychlost je úměrná přístupové době pevného disku a počtu FPGA.[120]

Nejzajímavějším aspektem celého hardwarového řešení je jeho celková cena. Ta se pohybuje v rozmezí 1000 až 1500 amerických dolarů.

[119] podle [67]
[120] viz [67], s. 3

5.5 Závěr studie

Je vidět, že ani nejrozšířenější mobilní síť není dostatečně chráněna před útoky. Přitom mobilní telefon dnes využívá již prakticky každý člověk. Útoky se vždy pouze vylepšují, zjednodušují a zlevňují. Z toho plyne potřeba nové kvalitní ochrany mobilní komunikace. Asociace GSM a 3GPP schválily nasazení nového šifrovacího algoritmu A5/3 (též zvaného KASUMI). Ten by měl nahradit nedostačující A5/1. A5/3 slouží nejen k zabezpečení GSM přenosů, ale také k šifrování dat v GPRS a EDGE. Ovšem i v šifře A5/3 byly nalezeny chyby.[121] Proto je stále třeba analyzovat stávající algoritmy a zavčas je nahrazovat novými. Vždyť i u nás se již diskutují možnosti využití mobilních telefonů GSM k tak citlivým operacím, jako je podávání daňového přiznání nebo dokonce volby.[122]

[121] viz. např. [68]
[122] viz. [69]

6 Závěr

V předchozích kapitolách přinesla tato diplomová práce přehled množství aspektů bezpečnosti v prostředí počítačové sítě internet. Rostoucí využívání internetu vede v mnohém k urychlení a zjednodušení mezilidské komunikace. Negativem jsou ovšem četná nově se vynořující bezpečnostní rizika. V síťovém prostředí je prioritou zachování důvěrnosti citlivých dat, jejich integrity a dostupnosti. Porušení některé z těchto zásad může mít (a často také má) neblahé důsledky.

Útočníci se snaží pronikat do systémů. Nezáleží na tom, zda tyto osoby podnikají své nezákonné kroky z pouhé zvědavosti, ze zájmu nebo s kriminálním úmyslem. Vždy se jedná o útočníky (zpravidla technicky nadprůměrně zdatné), jejichž průnikům se ostatní uživatelé snaží zabránit. Snahou některých útočníků je pouze se něčemu přiučit či získat popularitu ve své komunitě. Jiní mají však čistě kriminální motivy. Používají viry, trojské koně, zadní vrátka v systémech, metody odposlechu, útoky na integritu aj., aby získali přístup k cenným datům.

Proti útokům se můžeme bránit celou řadou prostředků. Existují antivirové programy a firewally jako základní ochrana každého počítače připojeného k internetu. Pro citlivé informace je však nejdůležitější kryptologická ochrana.

Šifrovací algoritmy tvoří kryptografové. Kryptoanalytici naopak šifry luští. Nemusí jít přitom vždy o nežádoucí proces. Kryptoanalytická komunita zkoumá zevrubně každý dostupný algoritmus, aby v něm objevila případné chyby a upozornila na ně. Tím tvoří jakýsi „motor“ vývoje lepšího zabezpečení důvěrných dat.

Existují dva typy kryptografie - symetrická a asymetrická. V případě nasazení symetrické kryptografie používají všechny komunikující strany stejný klíč pro šifrování i dešifrování zpráv. Výhodou je velká rychlost algoritmů. Nevýhodou je však problém bezpečného doručení klíče příjemci zprávy. Další nevýhodou je nadměrně rostoucí počet klíčů se vzrůstajícím počtem účastníků komunikace.

Oproti tomu asymetrická kryptografie používá klíče dva, které tvoří tzv. klíčový pár. Každý účastník má tedy svůj soukromí a veřejný klíč. Veřejný klíč může dát k dispozici každému, kdo s ním chce šifrovaně komunikovat. Zprávy však bude moci číst jen vlastník příslušného soukromého klíče z tohoto páru. Výhody jsou menší počet klíčů než u asymetrické kryptografie a vyloučení problému bezpečného doručení klíče příjemci. Nevýhodou je pak značná pomalost těchto algoritmů.

Dříve byl považován za nejbezpečnější algoritmus DES. Byl však prolomen prostou metodou hrubé síly díky projektu nadace EFF. Proto jej nahradil nový algoritmus AES. To dokazuje, že žádný algoritmus, i když je zpočátku považován za neprolomitelný, nezaručuje bezpečnost na „pořád“. Kryptologie se vyvíjí.

Nejen díky programu PGP našel uplatnění algoritmus IDEA. Využívá 128 bitový klíč a z něj generované podklíče.

Diffie-Hellmanova výměna klíče patří k asymetrickým bezpečnostním technikám. Je založena na problému řešení diskrétních logaritmů. Zajišťuje bezpečnou výměnu klíče mezi dvěma uživateli po nezabezpečeném kanále. Metoda má široké uplatnění. Je použita například v SSH nebo protokole IPSec.

Dalším významným asymetrickým algoritmem je RSA, vyvinutý známou trojicí Rivestem, Shamirem a Adlemanem v roce 1978. Jeho bezpečnost je založena na obtížnosti faktorizace velmi vysokých prvočísel.

Hašovací funkce jsou speciální transformační funkce, které libovolný vstupní řetězec převedou na výstupní řetězec s fixní délkou. Tento výstup se nazývá haš. Haš je tak vlastně zhuštěnou reprezentací vstupní zprávy nebo dokumentu, ze kterého byl vypočítán. Kryptografické hašovací funkce jsou používány ke kontrole integrity zpráv a k digitálním podpisům v nejrůznějších aplikacích informační bezpečnosti, jako je autentifikace a kontrola integrity zpráv. Nezbytnou vlastností hašovacích funkcí je tzv. lavinový efekt. To znamená,. že nepatrná změna na vstupu se projeví dalekosáhlými změnami výstupu. K nejrozšířenějším hašovacím funkcím dneška patří MD5 a funkce z rodiny SHA. Obě skupiny byly bohužel

v nedávné době prolomeny. Hledá se tak náhrada, která by znovu obnovila bezpečnost systémů využívajících hašování.

Stále více rozšířené bezdrátové sítě Wi-Fi mají vlastní šifrovací protokoly. Algoritmus WEP byl již prohlášen za nebezpečný, neboť se jej podařilo prolomit. V současnosti se tedy používá šifra WPA. Uživatelé staršího hardwaru však patrně WEP stále používají, neboť WPA nefunguje se zpětnou kompatibilitou na všech zařízeních. Je to ovšem zbytečné bezpečnostní riziko, neboť se tak vystavují možnosti odposlechnutí cenných dat, jako jsou přihlašovací údaje, důvěrná bankovní data atd.

Šifrovací algoritmy našly uplatnění v celé řadě implementací. Vznikly tak zabezpečené verze internetových protokolů. K zabezpečení komunikace mezi klientem a serverem a náhradě protokolů jako jsou ftp a telnet, slouží SSH. Existuje ve dvou verzích. SSH 1 je volně dostupné. SSH 2 je komerční. Obě verze jsou vzájemně nekompatibilní. Každá má své klady a zápory a záleží na uživateli, pro kterou verzi se rozhodne. V SSH jsou implementovány nejznámější kryptografické algoritmy DES, 3DES, IDEA a RSA. SSH 2 podporuje také AES a Blowfish.

Komunikaci přes http, smtp, pop3, ftp a další protokoly lze chránit prostřednictvím SSL. To zajišťuje ochranu přenášených dat mezi transportní a aplikační vrstvou.

Dalšími protokoly s aplikovanou kryptologickou ochranou jsou S-HTTP, IPSec, který působí přímo na úrovni TCP/IP vrstvy, a S/MIME, jenž zajišťuje bezpečnost přenosu elektronické pošty.

Poštu lze zabezpečit také programem PGP. Ten kromě toho umožňuje i generování a správu klíčů, šifrování dat na disku počítače a v neposlední řadě elektronický podpis. Pro vyhledávání a kontrolu veřejných klíčů všech uživatelů používá svou vlastní databázi klíčů PGP Global Directory. Výhodou PGP je, kromě výše uvedených schopností, možnost využívat jej zcela zdarma i vcelku jednoduché a intuitivní ovládání verze Desktop PGP.

Samostatnou kapitolou je elektronický podpis. Slouží k zajištění nepopiratelnosti a integrity u zpráv, odeslaných po internetu. Představuje velkou výhodu oproti „ručnímu“

podepisování, kde při jednání s úřady jsou lidé nuceni trávit velké množství času administrativní úkony pro ověření identity. Elektronický podpis má schopnost každou proceduru znatelně urychlit a zjednodušit. Jeho princip je založen na vytvoření digitálního výtahu zprávy (pomocí hašovacího algoritmu), který je zašifrován privátním klíčem odesílatele a přidán na konec zprávy. Příjemce znovu vypočítá výtah zprávy, dešifruje ten původní veřejným klíčem odesílatele a oba porovná. Pokud hodnoty souhlasí, nebyla zpráva po cestě nikým neoprávněně změněna a navíc je jednoznačně určena identita odesílatele, neboť jeho soukromý klíč nesmí mít nikdo jiný. Pokud hrozí kompromitace klíče, je nutné odvolat jeho platnost.

Důvěru mezi uživateli zajišťují certifikační autority. Jako nadřazené organizace svým klíčem podepisují uživatelské klíče. Autority také vydávají seznamy odvolaných certifikátů. Důvěru mezi autoritami zajišťují další, nadřazené, autority. Tato struktura se nazývá PKI (Private Key Infrastructure – Infrastruktura veřejného klíče).

V České republice v současnosti fungují tři státem uznané akreditované certifikační autority. Certifikáty jimi vydané může občan použít ke komunikaci s veřejnou správou. Lze tak například podávat daňová přiznání po internetu.

Podobně jako jsou běžné tištěné smlouvy a jiné důležité dokumenty opatřeny kromě podpisu zúčastněných stran i datem podpisu, je dobré mít také elektronické dokumenty opatřeny časovým údajem. Dnes je možné odeslat digitální dotisk libovolného dokumentu v elektronické podobě autoritě časových razítek, která jej označí tzv. elektronickým časovým razítkem. To podává důkaz, že daný dokument již existoval v době vydání tohoto razítka. Lze tak např. potvrdit platnost dokumentu, který je podepsán nyní již zneplatněným certifikátem.

Ochranu citlivých dat přímo na pevném disku počítače lze zajistit některou ze softwarových implementací šifrovacích algoritmů. V této práci je jako vhodný kandidát uveden program TrueCrypt - open-source aplikace, která dokáže vytvářet šifrované diskové oddíly na platformách Windows i Linux. Při zápisu do TrueCryptem zabezpečeného diskového oddílu dochází k šifrování, při čtení jsou data naopak dešifrována. Obě operace probíhají v reálném čase, tzv. „za pochodu“.

Případová studie podává stručnou historii systému GSM, popisuje architekturu mobilní sítě, její funkce a provoz, zabezpečení a implementované šifrovací algoritmy. Mobilní komunikace je dnes využívána nejen k přenosu hlasových či textových a obrazových zpráv, ale stále častěji i k přístupu na internet a s ním i k elektronickému bankovnictvím a dalším službám, vyžadujícím důvěrnost. Jak se však nyní ukazuje, není zabezpečení sítě GSM zdaleka tak bezpečné, jak by si takto rozšířená forma komunikace zasloužila. Práce Davida Hultona a Steva Millera se zaměřuje na přípravu efektivního pasivního útoku s využitím předem vypočítaných duhových tabulek, který se vyznačuje opravdu nízkou cenou a vysokou rychlostí. Tento pasivní útok bude s největší pravděpodobností znamenat konec současné formy šifrování GSM komunikace algoritmem A5/1.

Je vidět, že tématika internetové bezpečnosti je velice rozsáhlá. I přes existenci množství bezpečnostních nástrojů stále vzrůstá počet útoků na špatně (nebo vůbec) chráněné systémy. Mnozí uživatelé nevědí, jak by se měli takovým útokům chránit. Tato diplomová práce proto přinesla přehled mnoha bezpečnostních aspektů, jejichž znalost by měla napomoci lepšímu porozumění problematiky a volbě vhodných bezpečnostních opatření.

Seznam použité literatury

[1] BARRETT, D.J, SILVERMAN, R.E. *SSH : Kompletní průvodce*. Martin Blažík. 1. vyd. Brno : Computer Press, 2003. 556 s. ISBN 80-7226-852-X.

[2] CA Czechia. *Certifikační autorita CA CZECHIA.CZ* [online]. CA Czechia, c2003-2004 [cit. 2006-04-12]. Dostupný z WWW: <www.caczechia.cz>.

[3] DES Encryption : Overview. *Tropical Software* [online]. c 1997-2004 [cit. 2006-04-10]. Dostupný z WWW: <http://www.tropsoft.com/strongenc/des.htm>.

[4] DOSEDĚL, T. *Počítačová bezpečnost a ochrana dat*. 1. vyd. Brno : Computer Press, 2004. 190 s. ISBN 80-251-0106-1.

[5] ERICKSON, Jon. *Hacking : umění exploitace*. 1. vyd. Brno : Zoner Press, 2005. 263 s. ISBN 80-86815-21-8.

[6] GARFINKEL, S, SPAFFORD, G. *Bezpečnost v UNIXu a internetu v praxi*. Jiří Veselský. 1. vyd. Praha : Computer Press, c1998. 948 s. ISBN 80-7226-082-0.

[7] GARFINKEL, S. *PGP : Pretty Good Privacy*. Jaroslav Dudr. 1. vyd. Praha : Computer Press, c1998. 371 s. ISBN 80-7226-054-5.

[8] HLAVA, E. Dočkáme se zmrtvýchvstání e-podpisu?. *Lupa : Server o českém internetu* [online]. 2005 [cit. 2006-04-10]. Dostupný z WWW: <http://www.lupa.cz/clanky/dockame-se-zmrtvychvstani-e-podpisu/>. ISSN 1213-0702.

[9] International Data Encryption Algorithm. *Wikipedie : Otevřená encyklopedie* [online]. Naposledy editován v 19:45, 2. 4. 2006 [cit. 2006-04-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/International_Data_Encryption_Algorithm>.

[10] IPSEC Protocol Overview. *Connected : An Internet Encyclopedia* [online]. 1997 [cit. 2006-04-10]. Dostupný z WWW: <http://www.freesoft.org/CIE/Topics/141.htm>.

[11] Jak získat certifikát u CA Czechia. *Certifikační autorita CA Czechia.cz* [online]. c 2003-2004 [cit. 2006-04-10]. Dostupný z WWW: <http://www.caczechia.cz/article.asp?id=8>.

[12] LITTERIO, F. The Mathematical Guts of RSA Encryption. *Francis Litterio´s Homepage* [online]. c 1999-2001 [cit. 2006-04-10]. Dostupný z WWW: <http://world.std.com/~franl/crypto/rsa-guts.html>.

[13] Ministerstvo informatiky ČR. *E-podpis* [online]. Rok vyd. neuveden [cit. 2006-04-12]. Dostupný z WWW: <http://www.micr.cz/epodpis/default.htm>.

[14] PALÁT, P. CryptoAPI: Jak na IPSEC (I). LINUXZONE. 1.1.2003. Dostupný z WWW: <http://www.linuxzone.cz/index.phtml?ids=4&idc=795>

[15] PETERKA, J. Elektronické značky nastupují. *Lupa : Server o českém internetu* [online]. 2005 [cit. 2006-04-10]. Dostupný z WWW: <http://www.lupa.cz/clanky/elektronicke-znacky-nastupuji/>. ISSN 1213-0702.

[16] PETERKA, J. Elektronický podpis : Konečně rozjezd?. *IHNed.cz* [online]. 2004 [cit. 2006-04-10]. Dostupný z WWW: <http://www.ihned.cz/1-10070660-14459790-000000_d-eb>. ISSN 1213-7693.

[17] PETERKA, J. Jak oživit elektronický podpis. *Lupa : Server o českém internetu* [online]. 2004 [cit. 2006-04-10]. Dostupný z WWW: <http://www.lupa.cz/clanky/dockame-se-zmrtvychvstani-e-podpisu/>. ISSN 1213-0702.

[18] PINKAVA, J. Úvod do kryptologie. Dokument společnosti AEC, s.r.o., 1998 [cit. 2006-04-10]. Dostupný z WWW: <http://crypto-world.info/pinkava/uvod/uvod98.pdf>

[19] POŽÁR, J. *Informační bezpečnost*. Plzeň : Vydavatelství a nakladatelství Aleš Čeněk, 2005. 309 s. ISBN 80-86898-38-5.

[20] PŘIBYL, J, KODL, J. *Ochrana dat v informatice*. 1. vyd. Praha : Vydavatelství ČVUT, 1996. 299 s. ISBN 80-01-01664-1.

[21] PŘIBYL, J. *Informační bezpečnost a utajování zpráv*. 1. vyd. Praha : Vydavatelství ČVUT, 2004. 239 s. ISBN 80-01-02863-1.

[22] *Security technet : The Technology Portal for Cryptography and Network Security* [online]. Future Systems, 2000 [cit. 2006-04-12]. Dostupný z WWW: <http://www.securitytechnet.com/>.

[23] VLČEK, K. *Teorie informací, kódování a kryptografie*. 1. vyd. Ostrava : Vysoká škola báňská - Technická univerzita Ostrava, 1999. 182 s. ISBN 80-7078-614-0.

[24] What is DES?. *Next Wave Software* [online]. Rok vyd. neuveden [cit. 2006-04-12]. Dostupný z WWW: <http://www.thenextwave.com/page19.html>.

[25] What is Diffie-Hellman?. *RSA Security* [online]. c 2004 [cit. 2006-04-10]. Dostupný z WWW: <http://www.rsasecurity.com/rsalabs/node.asp?id=2248>.

[26] ZELENKA, J., et al. *Ochrana dat : Kryptologie*. 1. vyd. Hradec Králové : Gaudeamus, 2003. 198 s. ISBN 80-7041-737-4.

[27] EFF DES Cracker. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 2:32, 9. 3. 2008 [cit. 2008-03-02]. Dostupný z WWW: <http://en.wikipedia.org/wiki/EFF_DES_cracker>.

[28] Data Encryption Standard. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 3:03, 11. 4. 2008 [cit. 2008-03-02]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Data_Encryption_Standard>.

[29] Advanced Encryption Standard Process. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 0:16, 10. 1. 2008 [cit. 2008-03-03]. Dostupný z WWW: < http://en.wikipedia.org/wiki/Advanced_Encryption_Standard_process>.

[30] Advanced Encryption Standard. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 20:22, 8. 4. 2008 [cit. 2008-03-03]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Advanced_Encryption_Standard>.

[31] XSL attack. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 1:11, 9. 3. 2008 [cit. 2008-03-04]. Dostupný z WWW: <http://en.wikipedia.org/wiki/XSL_attack>.

[32] Avalanche effect. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 17:22, 15. 1. 2008. Dostupný z WWW: <http://en.wikipedia.org/wiki/Avalanche_effect>.

[33] Cryptographic hash function. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 17:12, 8. 4. 2008 [cit. 2008-03-05]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Cryptographic_hash_function>.

[34] Merkle-Damgård hash function. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 18:16, 26. 7. 2007 [cit. 2008-03-05]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Merkle-Damg%C3%A5rd_construction>.

[35] Hash collision. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 15:13, 4. 3. 2008 [cit. 2008-03-05]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Hash_collision>.

[36] Birthday attack. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 0:55, 23. 3. 2008. Dostupný z WWW: <http://en.wikipedia.org/wiki/Birthday_attack>.

[37] Preimage attack. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 2:28, 18. 3. 2008 [cit. 2008-2008-03-05]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Preimage_attack>.

[38] SHA-1. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 18:46, 2. 4. 2008 [cit. 2008-03-08]. Dostupný z WWW: <http://en.wikipedia.org/wiki/SHA-1>.

[39] MD5. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 1:21, 9. 4. 2008 [2008-03-08]. Dostupný z WWW: <http://en.wikipedia.org/wiki/MD5>.

[40] KLÍMA, V. Co se stalo s hašovacími funkcemi? aneb přehled událostí z poslední doby, část 1. *Crypto-World : informační sešit GCUCMP*, 3/2005 [online]. Dostupný z WWW: <http://crypto-world.info/casop7/crypto03_05.pdf>.

[41] WANG, FENG, LAI, YU. Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD. rump session, CRYPTO 2004, *Cryptology e-Print Archive*, Report 2004/199 [online]. Dostupný z WWW: <http://eprint.iacr.org/2004/199>.

[42] MIKLE, O. Practical Attacks on Digital Signatures Using MD5 Message Digest. *Cryptology ePrint Archive*, Report 2004/356 [online]. Dostupný z WWW: <http://eprint.iacr.org/2004/356>.

[43] KLÍMA, V. Nalézání kolizí MD5 – hračka pro notebook. *Crypto-World : informační sešit GCUCMP*, 3/2005 [online]. Dostupný z WWW: <http://crypto-world.info/casop7/crypto03_05.pdf>.

[44] KLÍMA, V. Tunnels in Hash Functions: MD5 Collisions Within a Minute (extended abstract). *IACR ePrint Archive*, Report 2006/105 [online]. Dostupný z WWW: <http//eprint.iacr.org/2006/105.pdf>.

[45] KLÍMA, V. Kolize MD5 do minuty aneb co v odborných zprávách nenajdete. *Crypto-World : informační sešit GCUCMP*, 4/2006 [online]. Dostupný z WWW:

<http://crypto-world.info/casop8/crypto04_06.pdf>.

[46] KLÍMA, V. Co se stalo s hašovacími funkcemi? aneb přehled událostí z poslední doby, část 2. *Crypto-World : informační sešit GCUCMP*, 4/2005 [online]. Dostupný z WWW

< http://crypto-world.info/casop7/crypto04_05.pdf>.

[47] VONDRUŠKA, P. Hledá se náhrada za kolizní funkce. *Crypto-World : informační sešit GCUCMP*, 5/2006 [online]. Dostupný z WWW:

<http://crypto-world.info/casop8/crypto05_06.pdf>.

[48] KLÍMA, V. Univerzální posilovače hašovacích funkcí, včetně MD5 a SHA1 aneb záchranné kolo pro zoufalce. *Crypto-World : informační sešit GCUCMP*, 6/2006 [online]. Dostupný z WWW: < http://crypto-world.info/casop8/crypto06_06.pdf>.

[49] The Whirlpool Hash Function. P*aulo Barreto´s Crypto Page* [online]. Naposledy editován 28. 10. 2006 [cit. 2008-03-10]. Dostupný z WWW: <http://paginas.terra.com.br/informatica/paulobarreto/WhirlpoolPage.html>.

[50] Evil twin phishing. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 17:59, 19. 2. 2008 [cit. 2008-03-12]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Evil_twin_phishing>.

[51] Wired Equivalent Privacy. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 20:15, 5. 4. 2008 [cit. 2008-03-12]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy>.

[51] Related-key attack. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 15:59, 27. 2. 2008 [cit. 2008-03-12]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Related-key_attack>.

[52] Wi-Fi Protected Access. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 20:21, 5. 4. 2008 [cit. 2008-03-14]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access>.

[53] Temporal Key Integrity Protocol. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 15:48, 26. 3. 2008 [cit. 2008-03-14]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Temporal_Key_Integrity_Protocol>.

[54] Secure Shell. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 23:40, 29. 3. 2008 [cit. 2008-03-17]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Secure_Shell>.

[55] Secure Socket Layer. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 13:46, 1. 4. 2008 [cit. 2008-03-18]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Secure_Sockets_Layer>.

[56] IPsec. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 14:41, 2. 4. 2008 [cit. 2008-03-18]. Dostupný z WWW: <http://en.wikipedia.org/wiki/IPsec>.

[57] S/MIME. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 14:12, 25. 3. 2008 [cit. 2008-03-19]. Dostupný z WWW: <http://en.wikipedia.org/wiki/S/MIME>.

[58] Pretty Good Privacy. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 14:25, 10. 4. 2008 [cit. 2008-03-19]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Pretty_Good_Privacy>.

[59] Zákon o elektronickém podpisu č. 227/2000 Sb. – Citováno z článku: Elektronické časové razítko jako doplněk el. podpisu. *Connect : Nejlepší časopis pro IT profesionály* [online]. Naposledy editován v 12:24, 26. 4. 2007 [cit. 2008-03-20]. Dostupný z WWW: <http://connect.zive.cz/?q=node/615>.

[60] Jak získat certifikát u CA Czechia. *Certifikační autorita: CA Czechia.cz* [online]. c 2003-2004 [cit. 2008-03-21]. Dostupný z WWW: <http://www.caczechia.cz/article.asp?id=8>.

[61] KLÍMA, V. Truecrypt: Profesionální ochrana dat zdarma. Root.cz [online]. c. 1998-2008. Naposledy editován 16. 7. 2007 [cit. 2008_03-23]. Dostupný z WWW: <http://www.root.cz/clanky/truecrypt-profesionalni-ochrana-dat-zdarma/>.

[62] Global System for Mobile Communications. *Wikipedie : Otevřená encyklopedie* [online]. Naposledy editován v 17:08, 22. 3. 2008 [cit. 2008-03-20]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Global_System_for_Mobile_Communications>.

[63] PENG, Ch. GSM and GPRS Security. Proceedings of the Helsinki University of Technology Seminar on Network Security fall 2000 [online]. Dostupný z WWW: <http://www.tml.tkk.fi/Opinnot/Tik-110.501/2000/papers/peng.pdf>.

[64] VONDRUŠKA, P. O chystané demonstraci prolomení šifer A5/1 a A5/2. *Crypto-World : informační sešit GCUCMP*, 2/2008 [online]. Dostupný z WWW: <http://crypto-world.info/casop8/crypto02_08.pdf>.

[65] The A5 Cracking Project. *Wiki : The Hacker´s Choice* [online]. Naposledy editován 22. 2. 2008 [cit. 2008-03-21]. Dostupný z WWW: <http://wiki.thc.org/cracking_a5>.

[66] Rainbow table. *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 5:58, 30. 3. 2008 [cit. 2008-03-21]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Rainbow_tables>.

[67] HULTON, S. Intercepting GSM Traffic. Black Hat Annual Conference 2008 White Papers [online]. Dostupný z WWW: <http://www.blackhat.com/presentations/bh-dc-08/Steve-DHulton/Whitepaper/bh-dc-08-steve-dhulton-WP.pdf>

[68] KASUMI (block cipher). *Wikipedia : The Free Encyclopedia* [online]. Naposledy editován v 8:01, 4. 3. 2008 [cit. 2008-03-23]. Dostupný z WWW: <http://en.wikipedia.org/wiki/A5/3>.

[69] PETERKA, J. Daňové přiznání: přes mobil? Lupa : Server o českém internetu [online]. Naposledy editován v 6:25, 28. 3. 2007 [cit. 2008-04-01]. Dostupný z WWW: <http://www.lupa.cz/clanky/danove-priznani-pres-mobil/>. ISSN 1213-0702.

Seznam zkratek

3DES – Triple Data Encryption Standard

3G – Třetí generace mobilních telefonů

3GPP – Partnerský projekt mobilních telefonů třetí generace

AES – Advanced Encryption Standard

AH – Authentication Header

AUC - Authentication Center

BSC - Base Station Controller

BSS – Base Station Subsystem

BTS - Base Transceiver Station

CA – Certifikační autorita

CBC – Cipher Block Chaining

CFB – Cipher Feedback Mode

COPACOBANA – Cost-Optimized Parallel Code Breaker

CRC – Cyclic Redundancy Check

CRL – Certificate Revocation List

DCCH – Dedicated Control Channel

DDoS – Distributed Denial of Service

DEA – Data Encryption Algorithm (viz DES)

DES – Data Encryption Standard

DH – Diffie-Hellman

DoS – Denial of Service

DSA – Digital Signature Algorithm

ECB – Electronic Code Book

EDE – Encryption-Decryption-Encryption

EDGE – Enhanced Data Rates for GSM Evolution

EEPROM – Electrically Erasable Programmable Read-only Memory

EFF – Electronic Frontier Foundation

EIR - Equipment Identity Register

ESP – Encapsulated Security Payload

FPGA - Field-Programmable Gate Array

FTP – File Transfer Protocol

FTPS – File Transfer Protocol - Secured

GPRS – General Packer Radio Service

GSM – Global System for Mobile communications (původně Groupe Spécial Mobile)

HLR - Home Location Register

HTTP – Hyper Text Transfer Protocol

HTTPS – Hyper Text Transfer Protocol - Secured

IDEA – International Data Encryption Algorithm

IEEE – Institute of Electrical and Electronics Engineers

IKE – Internet Key Exchange

IMSI - International Mobile Subscriber Identity

IPSec – IP Security

ISDN – Integrated Services Digital Network

LLC - Logical Link Control

MCC - Mobile Country Code

MD – Message Digest

MIC – Message Integrity Code

MS – Microsoft

MSC - Mobile Switching Centers

MSIN - Mobile Subscriber Identification Number

MSISDN - Integrated Services Digital Network Number

NIST – National Institute of Standards and Technology

NSA – National Security Agency

OFB – Output Feedback Mode

OMC - Operation and Maintenance Center

OMSS – Operation and Maintenance Subsystem

OT – Otevřený text

PGP – Pretty Good Privacy

PIN – Personal Identification Number

PK – Privátní klíč

POP3 – Post Office Protocol 3

RA - Routing Area

RAI - Routing Area Identity

RAM – Random Access Memory

RAND – náhodné číslo (Random)

ROM – Read-only Memory

RSA – Rivest-Shamir-Adleman

S/MIME – Secure / Multipurpose Internet Mail Extensions

SHA – Secure Hash Algorithm

S-HTTP -Secure Hyper Text Transfer Protocol

SIM – Subscriber Identity Module

SMSS – Switching and Management Subsystem

SMTP – Simple Mail Transfer Protocol

SRES - Signature Response

SSH – Secure Shell

SSL – Secure Socket Layer

ŠT – Šifrovaný text

TCP/IP – Transmission Control Protocol / Internet Protocol

TELNET – Telecommunication Network

TKIP – Temporal Key-Integrity Protocol

TLLI - Temporary Logical Link Identity

TLS – Transport Layer Security

UMTS – Universal Mobile Telecommunications System

VK – Veřejný klíč

VLR - Visitor Location Register

WEP – Wired Equivalent Privacy

Wi-Fi – Wireless Fidelity

WPA – Wi-Fi Protected Access

XSL – Extended Sparse Linearization