AEC s.r.o.

Úvod do kryptologie (*Jaroslav Pinkava* – květen 1998)

Kryptologie:
Zahrnuje kryptografii a kryptoanalýzu (někdy se také uvádí, že obsahuje steganografii – tajnopis).

Kryptografie:
Umění a věda v převedení informace do podoby, v níž je obsah této informace skryt. Je součástí kryptologie. Na rozdíl od steganografie (jejímž úkolem je skrýt existenci zprávy) je jejím úkolem učinit výslednou zprávu nečitelnou i v situacích, kdy je plně prozrazená (známá třetí straně).

Kryptografie zahrnuje **utajení** zpráv a autentizaci.

Základním prostředkem utajení zpráv je jejich šifrování. Šifrování spočívá v převedení zprávy (otevřeného textu) do jedné z astronomického počtu reprezentací (šifrového textu). Cílem šifrování je skrýt obsah zprávy před každým komu tato zpráva není určena. Konkrétní šifrový text je určen klíčem. Vzhledem k tomu, že všechny možné reprezentace mají stejnou pravděpodobnost, je nezbytné vyzkoušet všechny možné klíče, abychom našli ten správný, který dešifruje zprávu. Jestliže je klíčů dostatečně velký počet, je potom tato metoda prakticky neproveditelná.

Kryptografie má za cíl rozvíjet algoritmy, které lze použít ke skrytí obsahu zprávy před všemi s vyjímkou vysílající a přijímající strany (utajení) a k ověření správnosti zprávy přijímající stranou (autentizaci). Původní vysílanou zprávu nazýváme **otevřeným textem**. Tato zpráva je následně šifrována pomocí kryptografického algoritmu. Zašifrované zprávě říkáme **šifrový text**. **Dešifrování** je opačný postup vzhledem k šifrování, je to převedení šifrového textu zpět do podoby otevřeného textu.

Metodu šifrování můžeme považovat za bezpečnou pouze tehdy, jestliže i v situacích, kdy potenciální narušitel (nežádoucí třetí strana) má kompletní přístup ke všem komunikačním uzlům mezi vysílající a přijímající stranou, nemůže tento narušitel odkrýt původní obsah zprávy (získat otevřený text ze šifrového textu). Bezpečnost Vámi použité metody šifrování značí, že nelze získat z vašeho šifrového textu příslušný otevřený text známými metodami v dostupné době (v době, kdy data ještě nejsou zastaralá).

Všechny moderní algoritmy používají klíč, který kontroluje proces šifrování a dešifrování. Zprávu lze dešifrovat pouze tehdy, jestliže klíč použitý při dešifrování odpovídá klíči použitému při šifrování. Klíč použitý pro šifrování a klíč použitý pro dešifrování se nemusí přitom shodovat.

Moderní kryptografie však zahrnuje podstatně více než jen metody vedoucí ke skrytí obsahu zpráv. Autentizace je velmi potřebnou součástí dnešního života. Potřebujeme potvrdit, že druhým účastníkem komunikace, transakce jsme právě my a nikdo jiný. Potřebujeme vědět, že v takovýchto transakcích se nikdo jiný nemůže za nás vydávat. Účinné prostředky v tomto směru jsou vytvářeny právě na bázi kryptografických mechanizmů.

Digitální podpis je jednou z nejznámějších takových technik. Elektronické peníze již tvoří důležitou součást obchodní sféry. Důkazy s nulovým rozšířením informací nám umožňují prokázat svoji znalost nějaké skutečnost druhé straně a přitom samotný obsah této skutečnosti této straně neprozradit. Mechanizmus sdíleného tajemství umožňuje rozdělit tajemství mezi například osm lidí tak, že k rekonstrukci tohoto tajemství je zapotřebí alespoň pěti lidí z těchto osmi. Přitom to může být kterýchkoliv pět, těchto pět stačí, avšak menší počet (např čtyři) již rekonstrukci tajemství provést nemůže.

Diffie, W.; Hellman M. E.: "Privacy and Authentication: An Introduction to Cryptography",
in Proc. IEEE, Vol 67(3) Mar 1979, pp 397-427 (shrnutí)

Kryptoanalýza:
Je to část kryptologie, která se zabývá analýzou odolnosti (síly) kryptografického systému a metodami vedoucími k proniknutí (rozbití) do kryptografického systému. Šifru lze považovat za rozbitou, pokud lze získat informaci ze

zašifrované zprávy bez znalosti klíče použitého pro zašifrování této zprávy. Řada útoků je ve své podstatě teoretická a vyžaduje obrovské množství dat a provedených výpočtů. Takovýto přímý útok může být ve své podstatě obtížnou a drahou záležitostí. Často k úspěchu vedou spíše okolní cesty jako: krádež kopie původní zprávy, vydírání, úplatek anebo monitorování elektromagnetického vyzařování. Žádná šifra nemůže ochránit to, co bylo ukradeno jinou cestou. Při nevhodné konstrukci kryptografického algoritmu či určité kompromitaci existuje celá řada metod vedoucích až k získání klíče či příslušného otevřeného textu.

Steganografie:

Tajnopis. Metody vedoucí ke skrytí existence zprávy (např. mikrotečky, tajné inkousty atd.). Viz např. zde nebo zde.

Šifrování:

Je založeno na dvou komponentách – šifrovacím algoritmu a klíči.

Šifrovací (kryptografický) algoritmus:

je matematická funkce, která převádá srozumitelný text (otevřený text) na nesrozumitelný šifrový text. K zašifrování otevřeného textu používají šifrovací algoritmy jako vstup klíč. Jak klíč tak použitá funkce mají kritický význam pro šifrování. V současné době se používají dvě základní třídy šifrovacích algoritmů: symetrické a asymetrické.

Klíč:

V klasických situacích je pojem „klíče“ používán v souvislosti s pojmem „zámku“. Určité chráněné věci jsou přístupné pouze tehdy, pokud máme správný klíč.

V kryptografii při využití jednoho konkrétního klíče (z obrovské množiny možných klíčů) získáme pro určitý otevřený text jednu jeho konkrétní transformaci na šifrový text. Pouze pokud potom známe správný klíč, můžeme provést zpětnou transformaci šifrového textu na otevřený text. Tím, že množina možných klíčů je dostatečně veliká, zajistíme, že pro potenciálního narušitele je nemožní získat otevřený text prostým ozkoušením všech možných klíčů (útok hrubou silou, totální zkoušky).

V ideálních situacích je klíč vybírán ze základní velké množiny klíčů tak, aby pravděpodobnost volby každého konkrétního klíče z této množiny byla stejná. Pokud klíče nejsou vybírány náhodně, ale určité konkrétní klíče mají podstatně vyšší pravděpodobnost, pak potenciální protivník má samozřejmě možnost ozkoušet nejprve tyto „pravděpodobnější“ klíče. Tímto způsobem pak může výrazně redukovat náročnost probírky všech klíčů a dopracovat se ke správnému klíči i v podstatně kratší době.

V současných kryptografických systémech mají klíče vlastnost difuze. To značí, že při změně jednoho bitu klíče dojde v každém bitu šifrového textu k jeho změně s pravděpodobností jedna polovina. Pokud daný kryptografický systém nesplňuje tuto podmínku pro klíče, pak existují pro třetí stranu přístupy vedoucí k získání klíče (a tedy k rozbití kryptosystému).

Délka klíče (velikost množiny klíčů):

Délkou klíče obvykle rozumíme počet bitů jednotlivého klíče. Tento počet je roven dvojkovému logaritmu (logaritmu se základem 2) velikosti množiny klíčů.

Útok hrubou silou není jediným možným útokem proti konkrétnímu kryptosystému. Je to však útok, který je vždy možný. Tedy schopnost odolat útoku hrubou silou je jednou ze základních vlastností dnešních kryptografických systémů. V současné době je za kryptosystém dostatečně bráněný proti takovémuto útoku považován takový kryptosystém, kde délka klíče je alespoň 90-100 bitů (pro symetrické šifry – viz dále). Tomu odpovídá minimální velikost množiny potenciálních klíčů v rozmezí 2^{90} - 2^{100}.

V roce 1995 se skupina známých kryptografů a vědců pokusila odhadnout minimální délku klíče pro symetrické šifry. Opublikovali svůj odhad v článku:

Blaze, M., Diffie, W., Rivest, R.L., Schneier, B., Shimomura, T., Thompson, E., Wiener, M., "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security", January 1996.

Symetrické šifry:

Symetrické šifrovací algoritmy (také se jim říká šifry s tajným klíčem), používají tentýž klíč jak pro šifrování tak i pro dešifraci (při dešifraci je použita inversní funkce). Všechny klíče zde musí zůstat utajeny, aby bylo

zajištěno, že třetí neoprávněná strana nemůže použít klíč k dešifraci tajných zpráv. Klíč má být často střídán a být dostatečně náhodný.

Různé symetrické algoritmy používají různé délky klíčů. Delší klíč obvykle znamená větší bezpečnost algoritmu.

Symetrické šifry se dále dělí do dvou základních tříd: proudové a blokové šifry. Proudové šifry zašifrovávají vždy po jednom bitu v jednom časovém momentu. Blokové šifry zašifrovávají současně celý blok dat (obvyklá je jeho velikost 64 bitů).

Základní problém metod symetrické kryptografie spočívá v tom, že odesílatel musí nějakým způsobem doručit tento tajný klíč příjemci a zaručit, že klíč nebude cestou zachycen neautorizovanou osobou. Tedy hlavním problémem kryptografie využívající symetrické šifry je klíčové hospodářství (generování, přenos a uchovávání kryptografických klíčů). Zejména v systémech s velkým počtem uživatelů je problematika distribuce klíčů (pokud využíváme pouze prostředky na bázi symetrických šifer) velmi obtížně řešitelný problém.

Oproti asymetrickým šifrám (viz dále) mají symetrické šifry tyto základní přednosti: jsou obvykle podstatně rychlejší a využívají podstatně kratší klíč.

Asymetrické šifry (šifry s veřejným klíčem):

Používají pro šifrování a dešifraci vždy různé klíče. Jeden z klíčů se nazývá veřejným, druhý soukromým klíčem. Při komunikaci dochází k přenosu veřejných klíčů, zatímco soukromý klíč není nikdy nikam přenášen ani není s nikým sdílen. Tj. veřejný klíč je vysílající stranou použit k zašifrování dat pro konkrétního příjemce, zatímco soukromý klíč je tímto příjemcem utajován. Výsledkem je vytvoření bezpečné cesty pro výměnu kryptografických klíčů (např.pro symetrické šifrování). Veřejné klíče musí být asociovány se svými uživateli důvěryhodnou vhodně autentizovanou cestou. Veřejné klíče však nemusí být utajovány. Systémy s veřejným klíčem jsou význačně pomalejší než symetrické šifry. Z hlediska svých unikátních vlastností tvoří však jejich vhodný doplněk. Jsou používány zejména k přenosu klíčů, k vytváření digitálních podpisů (autentizace zpráv), jsou vhodným prostředkem při konstruování řady kryptografických protokolů.

Systémy s veřejným klíčem se poprvé objevily v článku: Diffie, W.; Hellman, M.E.: New Directions in Cryptography, IEEE Transactions on Information Theory, Vol. IT-22, No.6., Nov. 1976, pp. 109-112. Cílem koncepce navržené v tomto článku bylo řešit problematiku klíčového hospodářství. Pro ilustraci základních vlastností systémů s veřejným klíčem uvedeme následující dva příklady.

Příklad 1 (zašifrování):

Jestliže chce Alenka poslat tajnou zprávu Béďovi, použije Béďův veřejný klíč k zašifrování zprávy a pak ji odešle. Béďa pak užije svůj soukromý klíč k dešifrování zprávy, takže si ji může přečíst. Nikdo jiný nemůže dešifrovat zprávu, protože nemá Béďův soukromý klíč. Bezpečnost tohoto komunikačního systému tedy stojí na skutečnosti, že není možné získat soukromé klíče ze znalosti odpovídajících veřejných klíčů.

Příklad 2 (podpis):

Pokud Alenka chce Béďovi zaslat podepsanou zprávu (chce aby Béďa věděl, že pouze Alenka je původcem dané zprávy), zašifruje zprávu svým soukromým klíčem. V druhé fázi pak Alenka výsledek zašifruje veřejným klíčem Bédi. Béďa nejprve svým soukromým klíčem dešifruje získanou zprávu a následně provede její dešifraci pomocí veřejného klíče Alenky. Tímto způsobem má Béďa zaručeno, že původcem zaslané zprávy může být pouze Alenka. Pokud Alence jde pouze o podpis zprávy, nikoliv o utajení jejího obsahu, pak zprávu zašifruje pouze svým soukromým klíčem. Béďa se pak dešifrací pomocí Alenčina veřejného klíče ubezpečí o původci zprávy (to může pak učinit kdokoliv).

Tedy kdokoliv může pomocí veřejného klíče zašifrovat důvěrnou zprávu určenou majiteli klíče. Využije přitom informaci veřejně přístupnou (znalost veřejného klíče). Takto zašifrovanou zprávu může dešifrovat potom pouze majitel adekvátního soukromého klíče, tj. zamýšlený příjemce a nikdo jiný. Neexistuje způsob (pro třetí stranu) jak využít digitální podpis z jednoho dokumentu k podepsání jiného dokumentu. Nejmenší změna v podepsaném dokumentu zapříčiní, že proces verifikace podpisu neproběhne.

Při používání výhradně symetrických šifer existují rovněž určité přístupy umožňující autentizaci. Takovéto přístupy však vyžadují sdílení určitého tajemství a vyžadují někdy i účast důvěryhodné třetí strany. Příkladem takovéhoto přístupu je známý autentizační systém Kerberos. V tomto systému existuje centrální databáze uchovávající všechny tajné klíče. Vysílající strana pak v zásadě může popřít, že je původcem dřívější autentizované zprávy. Může tvrdit, že došlo ke kompromitaci klíče (jinde). Systémy s veřejným klíčem ji toto neumožní. Každý uživatel má plnou zodpovědnost za ochranu svého soukromého klíče. Tato vlastnost autentizace pomocí systémů s veřejným klíčem se nazývá **nepopiratelností**.

Nevýhodou systémů s veřejným klíčem je jejich pomalost (uvádí se, že asymetrické šifry jsou obvykle 1000 krát pomalejší než šifry symetrické). Proto při řešení konkrétního systému ochrany dat opírajícího se o kryptografické

algoritmy je nejlépe kombinovat oba základní přístupy, tj. symetrické a asymetrické algoritmy. Je přitom využívána rychlost symetrických algoritmů na jedné straně a flexibilita nesymetrických šifer na straně druhé.

Tajný klíč:
Klíč, který je používaný v symetrické kryptografii (secret key). Je znám všem zúčastněným stranám, nesmí být znám narušiteli. Viz též symetrické šifry.

Veřejný klíč:
Veřejná část dvojice klíčů v asymetrické kryptografii (public key). Veřejný klíč bývá široce dostupný a může být používán k šifrování zpráv a verifikaci digitálních podpisů.

Soukromý klíč:
Utajovaná část dvojice klíčů v asymetrické kryptografii (private key). Soukromý klíč je výhradním vlastnictvím jedné entity a není nikomu jinému sdělován. Je používán k dešifraci zpráv zašifrovaných veřejným klíčem a k vytváření digitálních podpisů (které lze verifikovat veřejným klíčem).

Tajná hodnota:
Hodnota, která je používána k odvození tajného klíče, sama však jako tajný klíč nesmí být používána (secret value).

Sdílený tajný klíč:
Tajný klíč sdílený dvěma či více stranami, obvykle je výsledkem dohody na klíči (shared secret key).

Sdílená tajná hodnota:
Tajná hodnota sdílená dvěma či více stranami - obvykle v průběhu dohody na klíči (shared secret value).

Funkce pro odvození klíčů:
Funkce s jejíž pomocí je na základě sdílené tajné hodnoty odvozován tajný klíč (key derivation function).

Proudové šifry:
Proudové šifry zašifrovávají vždy po jednom bitu v jednom časovém momentu. Typickým příkladem proudových šifer jsou šifry na bázi lineárních registrů s nelineárním výstupem. Tyto šifry jsou zdrojem binárního hesla, které je bit po bitu přičítáno k otevřenému textu.

Blokové šifry:
Blokové šifry zašifrovávají současně celý blok dat (obvyklá je jeho velikost 64 bitů, stejně veliký je i výstupní blok šifrového textu). Tyto šifry tedy obvykle definují vzájemně jednoznačné zobrazení z množiny 64-bitových čísel na množinu 64-bitových čísel. Většina obecně užívaných šifer (např. IDEA, DES, BLOWFISH) jsou blokové šifry.

Módy blokových šifer:
Pokud jeden a tentýž blok je dvakrát zašifrován tímtéž klíčem, obdržíme jako výsledný blok tentýž šifrový text (této metodě šifrování se říké elektronická kódová kniha - Electronic Code Book mode čili mód ECB). Takováto informace však může být užitečná pro potenciálního narušitele. V praxi by bylo proto výhodnější, aby týmž blokům otevřeného textu odpovídaly různé bloky šifrového textu. Všeobecně jsou užívány následující dvě metody:

mód CFB (Cipher Feedback mode): blok šifrového textu je získán zašifrováním minulého bloku šifrového textu (posledních 64 bitů) a přičtením části vzniklého šifrového textu (obvykle v délce 1 byte) modulo dva k stejně dlouhému bloku otevřeného textu.
mód CBC (Cipher Block Chaining mode): blok šifrového textu je získán tak, že sečteme nejprve mod 2 blok otevřeného textu s minulým šifrovým textem a výsledek zašifrujeme.

Na druhou stranu tyto způsoby šifrování vyžadují k započetí celého procesu určitou konkrétní hodnotu (inicializační vektor IV). Inicializační vektor se má dynamicky měnit, aby nebylo možné získat určité statistiky při opakujících se prvních blocích zpráv.
Pomocí módu OFB (Output Feedback mode) lze vlastně každou blokovou šifru využít jako zdroj binárního hesla a použít ji tedy jako proudovou šifru.

Jednosměrná funkce:

Jednosměrná funkce je matematická funkce, kterou v jednom směru (přímém) lze snadno spočítat, zatímco v opačném směru (inverzní zobrazení) probíhají výpočty velmi obtížně.

Hashovací funkce:

Vstupem (jednosměrné) hashovací funkce je blok proměnné délky (zpráva) a výstupem je blok pevné délky (obvykle 128 či 160 bitů) – hash. Při dané hodnotě hashe je výpočetně nemožné najít zprávu s tímto hashem, ve skutečnosti na základě znalosti hashe zprávy nemůžeme nic říci o obsahu vlastní zprávy. Pro některé jednosměrné hashovací funkce je výpočetně nemožné najít dvě různé zprávy s touž hodnotou hashe.

Příklady známých hashovacích funkcí jsou MD2, MD5 a SHA-1.

Digitální podpis:

Digitální podpis zajišťuje autentizaci. Je to (obecně řečeno) řetězec znaků, který určitým způsobem svazuje veřejný klíč a zprávu. Pouze osoba znající zprávu a odpovídající soukromý klíč mohla vytvořit tento řetězec. Kdokoli, kdo zná zprávu a veřejný klíč, může tento digitální podpis verifikovat.

Většina systémů s veřejným klíčem je pomalá. Provést podpis dlouhé zprávy (viz výše příklad 2) proto může být pro uživatele časově náročnou operací. Řešení poskytují hashovací funkce, jejichž rychlost je srovnatelná se symetrickými šifrovacími algoritmy a je tedy výrazně vyšší než rychlost algoritmů pro systémy s veřejným klíčem. Strana vytvářející digitální podpis určité zprávy nejprve spočte hodnotu hashe této zprávy a podepíše (svým soukromým klíčem) pouze tento hash. Kdokoli, kdo chce ověřit tento podpis postupuje analogicky. Spočte hodnotu hashe podepsané zprávy a porovná ji z hodnotou získanou dešifrací podpisu veřejným klíčem podpisující strany. Tedy podepsaný hash lze považovat za určitý otisk prstu (fingerprint) autora zprávy. Délka samotného hashe je přitom obvykle výrazně kratší než délka celé zprávy a tedy také vytvoření digitálního podpisu tohoto hashe trvá podstatně kratší dobu než kdyby byl vytvářen podpis celé zprávy.

Digitální podpisy ve formě otisku lze rovněž s výhodou využít např. v situacích, kdy je nutné uchovávat velkou řadu ověřených souborů. Pro každý takovýto soubor je spočten jeho otisk (message digest), který je pak spolehlivě uložen. Pokud je třeba prokázat správnost příslušného souboru, stačí znovu spočítat hodnotu jeho hashe a porovnat ji s dříve uloženou hodnotou hashe. Hashovací funkce lze také použít pro důkaz skutečnosti, že v příslušném souboru nebyly provedeny žádné změny (neboť dokonce přidání či změna jediného znaku vede ke kompletní změně hodnoty hashe). Dále důležitou úlohu hrají hashovací funkce při vytváření tzv. digitálních časových razítek. Hodnotu hashe lze totiž zveřejnit bez kompromitace obsahu vlastního dokumentu. Důvěryhodná strana podepíše hash dokumentu a časovou značku svým tajným klíčem, tím je později zaručeno, že dokument v příslušném čase již existoval.

Digitální podpis s rozkrytím zprávy:

Digitální podpis, který obsahuje dostatečné množství informací k tomu, aby z podpisu byla získána podepsaná zpráva. To eliminuje potřebu zasílat zprávu s podpisem.

Digitální podpis jako přídavek:

Takový digitální podpis, který neobsahuje podepsanou zprávu. Zpráva je k podpisu přiložena.

Klíčové hospodářství:

Pojem klíčového hospodářství je poměrně široký a zahrnuje v podstatě veškerou manipulaci s klíči. Jedná se především o generování klíčů, ukládání klíčů, ustavení klíčů (key establishment) a vlastní správu klíčů (key management). **Ustavení klíčů** lze ještě rozdělit na problematiku dohody na klíči (key agreement) a problematiku přepravy klíčů (key transport). **Správou klíčů** rozumíme množinu procesů a mechanismů, které podporují ustavení klíčů a udržující stávající vztahy systému klíčů (např. nahrazení starých klíčů novými atd.). Bezpečné klíčové hospodářství má nesmírnou důležitost pro funkčnost celého systému kryptografické ochrany. Většina konkrétních útoků je směřována právě spíše do oblasti manipulace s klíči, kde narušitel má v některých situacích podstatně vyšší šance než kdyby se pokoušel proniknout do použitého kryptografického algoritmu.

Dohoda na klíči:

Je to způsob, kterým se dvě či více entit dohodne na společném klíči, který znají pouze tyto entity. Využijí k tomu veřejné klíče druhé strany a své vlastní tajné klíče. Dohodnutý společný klíč pak spolu sdílí při užití nějakého symetrického šifrovacího algoritmu.

Kryptografický protokol:

Kryptografický protokol je sdílený algoritmus definovaný posloupností kroků, které precizují aktivity vyžadované na dvou či více entitách s cílem dosáhnout určitého bezpečnostního cíle.Tento algoritmus využívá kryptologické transformace (někdy se používá i pojem autentizační protokol či protokol typu výzva-odpověď). Účelem (cílem) kryptografických protokolů bývá: autentizace účastníků protokolu, utvoření dohody o dále použitém kryptografickém klíči, výměna těchto klíčů apod.

Autentizace:

Autentizace je proces ověření si totožnosti někoho nebo něčeho. Autentizace je kritická pro čestný a důvěryhodný průběh komunikací. Je to proces, pomocí něhož jedna strana (ověřovatel) získává ujištění, že identita druhé strany (žadatel, prokazující strana) je ta, jaká je deklarována. Cílem je zabránit záměně stran. Nejčastější technikou je, že ověřující strana prověří správnost zprávy, která demonstruje, že žadatel vlastní tajemství, které je asociováno se správnou stranou. V systémech s kryptografickou ochranou dat jsou k tomu využívány autentizační nebo také identifikační protokoly (viz kryptografické protokoly). Tyto protokoly mohou užívat jak konvenční kryptografické algoritmy (symetrické šifry) tak i algoritmy s veřejným klíčem. Autentizace v systémech s veřejným klíčem se opírá o digitální podpisy.

Digitální certifikát:

Uživatelé musí být schopni získat bezpečnou cestou klíče, které potřebují k zašifrovaní svých dat. Pro systémy s veřejným klíčem zde musí být cesta, jak se podívat, jaký veřejný klíč používá druhá strana. A na druhé straně musí mít cestu ke zveřejnění svého klíče. To ale nestačí. Uživatel musí mít důvěru v legitimnost takto získaného klíče. V opačném případě by mohl narušitel buď zaměnit veřejný klíč ležící někde v adresáři nebo by se mohl vydávat za někoho jiného. Pro tyto účely slouží certifikáty. Digitální certifikát označuje vlastníka veřejného klíče. Dovoluje verifikaci tvrzení, že daný veřejný klíč patří skutečně danému jedinci. Certifikáty pomáhají chránit se před možností, že někdo falzifikuje klíč s cílem vydávat se za někoho jiného. Ve své nejjednodušší podobě obsahují certifikáty veřejný klíč a jméno. Obecně užívané certifikáty obsahují rovněž:

- dobu vypršení platnosti
- jméno certifikační autority, která vydala certifikát
- pořadové číslo
- informaci o tom jak klíč má být používán
- nejdůležitější je digitální podpis vydavatele certifikátu

Certifikáty nesmí být možné padělat, musí být získány bezpečnou cestou a vytvářeny musí být tak, aby potenciální narušitel je nemohl zneužít. Vydání certifikátu musí rovněž probíhat bezpečným způsobem, musí být odolné proti možným útokům. Pokud by něčí tajný klíč byl ztracen či kompromitován, pak ostatní uživatelé musí být včas varováni a nesmí již déle šifrovat zprávy neplatným veřejným klíčem nebo akceptovat zprávy podepsané tímto zkompromitovaným tajným klíčem. Uživatelé musí své klíče mít bezpečně uloženy, na druhé straně musí mít tyto klíče k dispozici pro jejich legitimní používání. Klíče mají platit pouze do doby než vyprší jejich platnost. Doba platnosti musí být vhodně zvolena a bezpečně opublikována. Je třeba rovněž vzít do úvahy, že některé dokumenty budou mít zapotřebí ověřit platnost podpisu i po uplynutí doby platnosti daného veřejného klíče.

Nejrozšířenějším akceptovaný formát pro certifikáty je definován mezinárodní normou CCITT X.509. Tyto certifikáty mohou být pak čteny či psány libovolnou aplikací vytvořenou ve shodě s X.509. Normu X.509 využívá řada protokolů, např. PEM, PKCS, S-HTTP a SSL.

Certifikační autorita:

Certifikační autorita je organizace (důvěryhodná třetí strana), která podepisuje uživatelův veřejný klíč a jeho jméno (případně další doplňkové údaje jako doba platnosti) svým vlastním tajným klíčem. Certifikát lze ověřit veřejným klíčem certifikační autority. Pokud chtějí nyní dva partneři spolu komunikovat, mohou se vzájemně autentizovat ověřením digitálního podpisu druhé strany veřejným klíčem partnera a posléze ověřením partnerova veřejného klíče verifikací digitálního podpisu certifikátu užitím veřejného klíče certifikační autority. Stačí pak důvěřovat veřejnému klíči certifikační autority. Tímto způsobem je redukován počet veřejných klíčů, kterým každý s uživatelů musí důvěřovat.

Certifikační autority často také provádí verifikaci klíčů, aby bylo zajištěni, že tyto klíče byly správně vygenerovány. Je jim důvěřováno, že správně provedou verifikaci. Na druhou stranu jim není sdělována žádná tajná informace (např. jiné uživatelovy tajné klíče).

Při větším počtu uživatelů jedna certifikační autorita nestačí. Veřejný klíč jedné certifikační autority může být certifikován jinou certifikační autoritou. Vytváří se tak sítě certifikačních autorit, které mohou mít různou hierarchickou strukturu.

Pro konkrétní certifikační autoritu je důležité jak postupuje při vydávání certifikátů, zejména pak jak prověřurje oprávnění žadatele o certifikát. Některé certifikační autority mohou požadovat při identifikaci uživatele velmi málo, ale například banky nebudou zajisté chtít věřit certifikátům s nízkou úrovní jistoty. Každá certifikační autorita musí zveřejnit své požadavky na identifikaci klienta a svoji politiku v této oblasti. Další strany tak mohou posoudit úroveň spolehlivosti certifikátů dané certifikační autority.

Seznam odvolaných certifikací:

Seznam odvolaných certifikací (CRL – Certification Revocation List) je seznam veřejných klíčů, které byly odvolány dříve než skončila doba jejich platnosti. Je řada důvodů, pro které mohl být klíč odvolán a umístěn v CRL. Klíč mohl být kompromitován. Klíč mohl být určen pro zaměstnance firmy, který mezitím z firmy odešel. Při ověřování podpisu je nutné si ověřit zda příslušný klíč není umístěn v CRL. CRL je provozován certifikační autoritou a obsahuje informaci o odvolaných klíčích, které byly původně certifikovány touto certifikační autoritou. Jsou zde umístěny pouze klíče jejichž původní doba platnosti nevypršela (klíče s vypršenou dobou platnosti nesmí být akceptovány v žádném případě).

Autentizovaný klíč:

Klíč, který je vázán autentizovanou cestou na svého uživatele, např. použitím digitálního certifikátu.

Ukládání klíčů:

Utajované klíče (tajné a soukromé klíče) musí být bezpečně uloženy. Jejich kompromitace vede ke ztrátě všech výhod použitého kryptografického systému. Utajovaný klíč by nikdy neměl být uložen v otevřené podobě. Nejjednodušší metodou je zašifrování utajovaných klíčů pomocí **hesla** (password) a uložení výsledku na disk. Samozřejmě heslo musí být voleno tak, aby jeho použití vytvářelo dostatečnou bezpečnost pro uživatele. Heslo musí mít adekvátní délku, nesmí být snadno uhodnutelné a musí být s ním zacházeno odpovídajícím způsobem (například je nevhodné zapsat si heslo do stolního kalendáře atd.). Jinou metodou je umístění utajovaných klíčů na chipovou kartu (popř. hesla - resp. je zde umístěn tzv. hlavní klíč, jímž jsou ostatní klíče zašifrovány). Při zvýšených nárocích na bezpečnost (například soukromý klíč certifikační autority) je vhodné mít klíč umístěný ve speciálním hardwareovém zařízení.

Bezpečnost šifer:

Existuje pouze jediná absolutně bezpečná šifra. Tuto šifru poprvé použily za první světové války Joseph Mauborgne a Gilbert Bernam (1917) a tvoří jí užití náhodně generované posloupnosti (hesla, anglicky one-time pad). Toto heslo je tvořeno znaky a při šifrování je každý z nich použit pouze jednou k zašifrování jednoho znaku otevřeného textu. Po použití nesmí být toto heslo již nikdy více znovu použito. Pokud zůstává heslo bezpečné je bezpečným i obsah zprávy. Užití takovéto šifry má dva hlavní problémy. Jednak je obtížné získavat skutečně náhodná čísla (tzv. kryptograficky náhodná čísla, tj. čísla, která navíc nesmí být predikovatelná) a jednak vzhledem k tomu, že heslo nikdy nesmí být použito dvakrát, je jeho potřebná délka shodná s délkou zašifrované zprávy.

Poznámka: Obsah pojmu hesla v tomto odstavci se liší od obsahu tohoto pojmu v odstavci předcházejícím. Bohužel v české odborné terminologii již toto dvouznačné užívání pojmu hesla dosti pevně zakotvilo.

Pokud je konkrétní algoritmus vhodně konstruovaný, pak jedinou smysluplnou metodou je ověřování všech možných klíčů. Proti tomuto útoku hrubou silou stojí potom velikost této množiny klíčů.

Délka klíče však nemusí být rozhodující. Řadu šifer lze rozbít i jinou cestou. Návrh vlastní šifry se může stát sám o sobě zábavnou činností, nedoporučuje se však, pokud nejste skutečný expert a nevíte přesně co děláte. Zde je třeba zejména varovat před neopublikovanými či tajnými algoritmy. Návrhář takovéhoto algoritmu si často není jistý bezpečností svého algoritmu, neboli bezpečnost algoritmu závisí na jeho utajení. Obecně řečeno, žádný algoritmus, jehož bezpečnost závisí na utajení samotného algoritmu, není bezpečný. Zkušenosti ukazují, že většina tzv. tajných algoritmů, které byly později opublikovány, byla ve skutečnosti směšně slabá. Naopak každý kvalifikovaný kryptolog vám řekne, že dobře navržený šifrovací algoritmus utajovat není třeba.

Pro systémy s veřejným klíčem platí, že požadovaná délka klíčů (z hlediska bezpečnosti příslušné šifry) je zde obvykle podstatně větší než délka klíče pro symetrickou šifru. Vzniklá situace zde totiž kryptoanalytikovi dává další možnosti. Tou bude nejen snaha uhádnout správný klíč, ale bude se i pokoušet odvodit tajný klíč z odpovídajícího veřejného klíče. Například pro RSA, které so opírá opírá o využívání výpočtů v modulu o velikosti 256 bitů, vyřeší odpovídající úlohu faktorizace 256 bitového čísla v podstatě kdokoliv. Klíče o velikosti 384 bitů rozbije každá

univesitní skupina či odborná firma. Klíče o velikosti 512 bitů rozbijí vládní agentury. Vzhledem k rozvoji výpočetních možností není dlouhá budoucnost předpovídána ani klíčům o velikost 768 bitů (i když z dnešního hlediska je lze považovat za bezpečné). Klíče délky 1024 bitů budou bezpečné, pokud nebudou dosaženy zásadní výsledky v řešení úlohy faktorizace. O velikosti klíče 2048 bitů se dnes předpokládá, že zabezpečuje bezpečnost šifry RSA na několik desetiletí. Následující tabulka z knihy:

Schneier, B., *Applied Cryptography Second Edition: protocols, algorithms, and source code in C*, John Wiley & Sons, 1996.

srovnává délky klíče pro symetrickou a asymetrickou šifru (jako RSA či Diffie-Hellman):

Symmetric	**Asymmetric**
56	384
64	512
80	768
112	1792
128	2304

Vždy je však třeba mít na zřeteli, že kryptografický algoritmus je pouze částí většího systému. Systém není nikdy silnější než jakou sílu má jeho nejslabší část.

Anderson, Ross J., "Why Cryptosystems Fail", Communications of the ACM, November, 1994.

Metody kryptoanalýzy:

V kryptoanalýze vždy předpokládáme, že kryptoanalytik zná celý šifrovací algoritmus. Existují čtyři základní typy kryptoanalytických útoků:

- využití pouze šifrového textu (ciphertext-only attack). Jedinou informací, kterou kryptoanalytik disponuje, je znalost šifrového textu různých zpráv zašifrovaných týmž algoritmem (a týmž klíčem).
- využití dvojic otevřený text a šifrový text (known-plaintext attack). Kryptoanalytik zde zná nejen příslušné šifrové texty, ale i jim odpovídající otevřené texty.
- využití volitelných otevřených textů (chosen-plaintext attack). Oproti předešlému typu útoku má kryptoanalytik navíc možnost vybírat si jaké otevřené texty budou zašifrovány (vůči použití tohoto typu útoku je například zranitelný algoritmus RSA).
- využití volitelných šifrových textů (chosen-ciphertext attack). Kryptoanalytik si může vybírat šifrové texty, které budou posléze dešifrovány a má přístup k takto dešifrovaným otevřeným textům.

Útok hrubou silou jsme již zmínili. Jednou z nejúspěšnějších metod posledních let je tzv. diferenciální resp.lineární kryptoanalýza. Proti některým systémům s veřejným klíčem lze použít (v určitých případech) útok založený na měření doby šifrování (timing attack, viz článek Paul C. Kocher: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems). I když existuje celá řada jiných kryptoanalytických technik, výše uvedené pravděpodobně zahrnují ty nejdůležitější. Při návrhu nového šifrovacího algoritmu je třeba podstatně hlouběji znát tuto problematiku. Vhodným zdrojem pro získání výchozích informací jsou např. knihy:

- ***Handbook of Applied Cryptography***, by Menezes, van Oorschot and Vanstone (CRC Press, 1997).
- ***Applied Cryptography*** by Bruce Schneier (John Wiley & Sons, 1996).

Z historie kryptologie:

První známé příklady užití kryptografických postupů pochází již z antiky (před 4000 lety) . Při postupu napoleonských vojsk v Egyptě roku 1799 byl nalezen dávný sloup (Rosetta Stone), pocházející z roku 196 před naším letopočtem. Na jeho povrchu se nachází zpráva zašifrovaná ve třech různých jazycích. Ve starém Řecku byly k utajení zpráv používány hole různých průměrů, na které byl namotáván pásek. Zpráva se zapisovala podélně délky hole. Po napsání zprávy byl pásek rozmotán. Osoba, které byla zpráva určena musela k tomu, aby zprávu přečetla, mít hůl se stejným průměrem.

Julius Caesar využíval jednoduchou záměnu jednotlivých písmen abecedy založenou na posuvu v abecedě (známá Caesarova šifra). Kardinál Richeliu byl vynálezcem šifrovací mřížky (známé např. z knihy Julia Vernea: Matyáš Sandorf. Nový hrabě Monte Christo).

V 20.století pak kryptografie prochází bouřlivým rozvojem a to jak v souvislosti s oběma světovými válkami (diplomacie a armáda vždy potřebovaly chránit své informace), tak pak zejména v návaznosti na hromadné zpracování dat pomocí výpočetní techniky.

Pro zájemce o hlubší poznání historie kryptologie je nejlépe doporučit knihu:

- ***Codebreakers***, by David Kahn, Macmillan Co., New York, 1967,

která prošla již i několika dalšími vydáními.
The Codebreakers obsahuje detailní historii kryptografie, často podanou ve stylu velkých dobrodružství. Obecně je kniha psána nematematicky a netechnicky. Autor však ukazuje i konkrétní příčiny neúspěšnosti jednoduchých šifer. Konkrétní příklady dokumentují, že reálná kryptografie obsahuje daleko více než pouhá šifrovací schemata. Rozhodně by stálo za to vydat tuto knihu v českém jazyce.

Knihou obdobného typu je

- ***Decrypted Secrets***, by Friedrich Bauer (Springer-Verlag, 1997).

Kniha je poněkud více technicky zaměřená a téměř polovina knihy se zabývá problematikou kryptoanalýzy v období druhé světové války.

Důkazy s nulovým rozšířením informací:

V zásadě se jedná o komunikační protokol mezi dvěma stranami. Jedna strana se nazývá dokazovatel, druhá strana ověřovatel. Dokazující strana chce přesvědčit ověřující stranu otom, že zná nějakou skutečnost. Přitom nechce předat ověřující straně žádné informace, které by mohly pomoci této straně zjistit něco z této skutečnosti.

Například dokazující strana zná faktorizaci velkého čísla (použitého jako modul v systému RSA), chce o této skutečnosti přesvědčit ověřující stranu. Zároveň ale nechce poskytnout ověřující straně žádné vodítko, které by ji umožnilo získat tuto faktorizaci.

Praktický smysl mají tyto postupy zejména při identifikaci. Prověřovaná strana (dokazovatel) má takto možnost prokázat, že je tím za koho se vydává a nemusí přitom sdělit jakoukoliv informaci, která by později mohla umožnit někomu jinému, aby se za něho vydával.

Čtenáři zajímajícímu se o teoretické aspekty problematiky doporučuji on-line knihu O.Goldreicha, kde lze nalézt i další odkazy.

Sdílené tajemství:

V praxi často nastává situace, kdy je třeba vyřešit následující problém klíčového hospodářství. Existuje jeden tajný klíč, který zajišťuje přístup k řadě důležitých souborů. Pokud by se tento klíč stal z nějakého důvodu nedostupným (např. příslušná osoba zemře, odejde z firmy apod.), pak všechny tyto důležité soubory zůstanou nedostupné. Podstatou řešení problému sdíleného tajemství je rozdělit toto tajemství více osobám a sice určitým způsobem, který umožňuje následující. Tajemství je rozděleno mezi **n** osob. Pouze v případě, kdy se sejde nejméně **m** z těchto **n** osob (**m< n**) lze toto tajemství rekonstruovat. Nezáleží přitom, kterých **m** konkrétních osob se právě sešlo. Existují dva základní přístupy k řešení této úlohy, jejichž autory jsou Shamir a Blakley (viz zde).

Kryptoschemata:

DES:

Tento algoritmus patří mezi kryptografické standardy (USA – FIPS 46-2). Byl vyvinut firmou IBM v sedmdesátých letech. V roce 1977 se stal americkou vládní normou pro šifrování (certifikován NIST – National Institute of Standards and Technology - naposledy v roce 1993). Je široce používán, zejména ve finanční sféře. DES je symetrická bloková šifra s blokem o velikosti 64 bitů. Klíč má délku 56 bitů (+8 paritních bitů). Algoritmus DES je obvykle užíván v módu CBC či CFB. Vysoká rychlost je dána také menším rozměrem klíče. Na rychlém 486 PC je uváděna dosažená rychlost 400 kb za vteřinu.

Bezpečnost algoritmu je relativně vysoká. Existující metody popsané v literatuře (totální zkoušky, diferenciální a lineární kryptoanalýza) umožňují za určitých předpokladů schema rozbít. Náklady na tyto metody jsou však zatím stále značné. Je také otázkou zda např. lze v praxi splnit předpoklady pro provádění diferenciální kryptoanalýzy

(nezbytnost vytváření obrovského množství dvojic otevřený a šifrový text s využitím téhož klíče) atd. Algoritmus je určen pro aplikace, kde vzhledem k ceně chráněných informací není nutné vyžadovat absolutní bezpečnost dat.

V současné době ho však nelze považovat za perspektivní algoritmus (viz DES-Challenge dokumentující úspěšné rozbití DES, které trvalo několika tisícům počítačů několik měsíců). V tomto článku jsou obsaženy nároky hardwareového řešení v ceně jednoho miliónu dolarů, které rozbije DES zhruba za sedm minut. Slabé, poloslabé a „možná" slabé klíče algoritmu jsou shrnuty v materiálu zde.

Diferenciální kryptoanalýzou DES se zabývali E.Biham a A.Shamir (viz článek Differential Cryptanalysis of the Full 16-Round DES).

Vývoz implementací DES z USA je kontrolován NSA (National Security Agency).

3-DES

3DES je zesílená varianta kryptografického standartu. Využívá dvojnásobně dlouhý klíč, tj. 112 bitů. Algoritmus DES je použit třikrát, v prvním a třetím kroku šifruje (pomocí první části klíče), v druhém kroku dešifruje (pomocí druhé části klíče). Někdy je 3-DES implememtována tak, že v třetím kroku používá rovněž odlišný klíč. Celková délka klíče pak dosahuje 168 bitů. Algoritmus 3-DES je považován za mnohem bezpečnější než standartní algoritmus DES. Kriticky se k vícenásobným používáním jednoho schematu staví Eli Biham v práci Cryptanalysis of Triple-Modes of Operation .

Rychlost 3-DES je téměř třikrát menší než rychlost DES (např. se uvádí pro rychlé 486 PC dosažená rychlost šifrování 150 kb za vteřinu). Výhodou 3-DES je skutečnost, že aplikace používající DES lze jednoduše převést na používání 3-DES (pokud se neukáže kritickou právě rychlost šifrování).

BLOWFISH:

Blowfish je algoritmus s proměnnou délkou klíče (32 – 448 bitů). Blowfish kombinuje Feistelovu šifru, neinvertibilní funkci F a S-boxy závislé na klíči. Na počátku probíhá fáze, kdy z vstupního kliče je vytvořeno osmnáct 32-bitových subklíčů a čtyři 8 x 32 bitové S-boxy (celkem 4168 bytů). Algoritmus je považován za bezpečný a je velmi rychlý. Blowfish byl navržen v roce 1993 a opublikován v roce 1994 Bruce Schneierem, (mezinárodně uznávanou autoritou v oboru kryptografie, autor knihy Applied Cryptography.). Článek B.Schneiera lze získat zde. Algoritmus má určité slabé klíče. Tyto klíče produkují stále týž vstup do S-boxu. Pravděpodobnost vybrání takovéhoto klíče je zanedbatelná.

IDEA:

IDEA (International Data Encryption Algorithm) je algoritmus s délkou klíče 128 bitů. Pro svou značnou bezpečnost a vysokou rychlost je považován za perspektivní algoritmus.

Algoritmus IDEA byl vyvinut společným projektem Swiss Federal Institute of Technology in Zürich (Dr. Xuejia Lai / Prof. James Massey) a firmou Ascom. Algoritmus IDEA je podle známých zkušeností resistantní vůči diferenciální kryptoanalýze. IDEA je symetrický algoritmus, bloková šifra s velikostí bloku 64 bitů. Tentýž algoritmus je použit při šifrování i dešifraci. Algoritmus se opírá o využití tří základních operací: XOR (součet modulo dva), součet mod 2^{16}, součin mod $2^{16} + 1$. IDEA běží v 8 velkých cyklech a používá přitom 52 subklíčů. Každý cyklus používá 6 subklíčů, zbývající 4 jsou použity pro výstupní transformaci. V každém cyklu se pracuje se čtyřmi 16 bitovými bloky, se kterými jsou prováděny výše uvedené základní operace. Algoritmus IDEA je obvykle užíván v módech CFB a CBC (jako DES).

IDEA má známý význačně slabý klíč sestávající ze samých nul. V knize Menezes A.J.; van Oorschot, Paul C.; Vanstone, Scott A.: Handbook of Applied Cryptography, CRC Press 1997 jsou obsaženy odvolávky na další autory (Daemen) zabývající se problematikou slabých klíčů u algoritmu IDEA. Pravděpodobnost použití takovéhoto slabého klíče při náhodné generaci klíčů je 2^{-77}. Existuje metoda, jak se použití těchto klíčů plně vyhnout. Různé skupiny kryptologů prováděli kryptoanalýzu šifrovacího algoritmu IDEA, zatím žádná nepřišla se zveřejněním nějakých slabin algoritmu. Některé metody kryptoanalýzy byly vyzkoušeny na schematu s redukovaným počtem cyklů.

Algoritmus IDEA je při šifrování přibližně dvakrát rychlejší než algoritmus DES (a současně nabízí podstatně vyšší úroveň bezpečnosti).

IDEA je patentována v USA a ve většině evropských zemí (patent neplatí ve Finsku). Majitelem patentu je firma Ascom-Tech. Nekomerční použití algoritmu je bezplatné.

CAST:

CAST, navržený autory Carlisle Adams a Stafford Taverns, je solidní moderní algoritmus (bloková šifra s délkou bloku 64 bitů). Jeho design je velmi podobný algoritmu Blowfish, obsahuje S-boxy závislé na klíči, dále neinvertibilní funkci f a má strukturu Feistelovy šifry. David Wagner, John Kelsey a Bruce Schneier objevili určitý typ útoku proti 64 bitové (tj. zjednodušené) verzi algoritmu CAST, který vyžaduje přibližně 2^{17} zvolených otevřených textů a 2^{48} offline výpočtů (popsáno v tomto článku). CAST je patentován firmou Entrust Technologies, která ho postoupila pro volné užití. Popis algoritmu CAST-128 lze získat například zde nebo zde(128-bitová verze). V materiálu CAST Design Procedure Addendum (na téže adrese) jsou obsaženy specifikace pro užití algoritmu CAST s různou délkou klíče (40 až 128 bitů) a testovací vektory pro délky klíče 40, 80 a 128 bitů. C.Adams zpracoval postup při návrhu algoritmu CAST a obdobných šifer v tomto článku. Algoritmus nemá slabé klíče a je odolný vůči diferenciální a lineární kryptoanalýze.

SHA-1:

SHA-1 (Secure hash algoritnm) je hashovací funkce odpovídající normě FIPS PUB 180-1.Vytvoří 160 bitů dlouhý kontrolní hash. Algoritmus byl vyvinut NIST jako součást SHS (Secure Hash Standard). Původně publikovaný algoritmus SHA byl stažen a toto je jeho opravená verze. Algoritmus je zhruba o 25% pomalejší než MD5 (je však bezpečnější, poskytuje delší hodnotu hashe – 160 namísto 128 bitů). Byl navržen v souvislosti s normou DSS (Digital Signature Standard). Tento hash je při vytváření digitálního podpisu (autentizaci) zašifrován pomocí tajného klíče asymetrického algoritmu (RSA či ELLIPT).

MD2:

Algoritmy MD2 a MD4 jsou starší verze hashovacích algoritmů firmy RSA Data Security Inc. Vytvořený hash má délku 128 bitů.V současné době nejsou tyto algoritmy doporučovány. Algoritmu MD2 je vytýkána jeho pomalost, k jeho bezpečnosti snad připomínky nejsou (Schneier, B.: Applied Cryptography, Second Edition, str. 441). MD2 byl konstruován pro práci na 8 bitových procesorech. Algoritmus MD2 je v některých situacích stále používán (vzhledem ke kompaktnosti své implementace). Algoritmus MD2 je popsán v RFC 1319. Algoritmus MD4 je popsán v RFC 1320.

MD5:

Algoritmus MD5 vyvinula společnost RSA Data Security Inc. Lze ho použít k vytvoření hashe v délce 128 bitů ze zprávy libovolné délky. Je považován za dostatečně bezpečný algoritmus a je široce používán. Avšak např. Hans Dobbertin ukázel, že pro kompresní funkci MD5 lze nalézt kolize zhruba za 10 hodin na PC. Pokud však tento typ útoku nebude rozšířen na plné MD5 nelze pochybovat o bezpečnosti algoritmu. MD5 je zhruba o 33% pomalejší než MD4. Detailně je algoritmus popsán v RFC 1321. Algoritmy MD4 a MD5 jsou veřejně dostupné k libovolnému použití.

RIPEMD-160:

Nejnovějším hashovacím algoritmem je RIPEMD-160, který byl navržen s cílem nahradit MD4 a MD5. Vytváří (jak vyplývá z jeho názvu) hash v délce 160 bitů. Byl vyvinut v rámci evropského projektu RIPE. Jeho autoři našli kolize ve ve verzi RIPEMD omezené na dva cykly. Úplný popis RIPEMD-160 lze nalézt zde.

DSS:

DSS značí Digital Signature Standard, který specifikuje Digital Signature Algorithm (DSA). Byl vybrán NIST (ve spolupráci s NSA) jako vládní norma pro digitální autentizaci. Je založen na problému diskrétního logaritmu a je odvozen ze systému, který původně navrhli Schnorr a ElGamal:

C.P. Schnorr. Efficient identification and signatures for smart cards. In Advances in Cryptology --- Crypto '89, pages 239--251, Springer-Verlag, New York, 1990.
T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, IT-31:469--472, 1985.

Algoritmus je určen pouze k autentizaci. Po svém zveřejnění byl algoritmus poměrně značně kritizován, zejména z následujících důvodů: chybí prostředky pro výměnu klíčů, o bezpečnosti příslušného kryptosystému se zatím ví poměrně málo, verifikace podpisů pomocí DSS je pomalá, byl očekáván standart na bázi RSA. V systému DSS je generování podpisu rychlejší než verifikace tohoto podpisu, u RSA je to obráceně (při vhodné volbě veřejného a tajného exponentu), řada odborníků se domnívá, že je lépe, když rychlejší je verifikace.

Nejvíce však byl algoritmus kritizován z hlediska jeho bezpečnosti. Původní návrh obsahoval délku klíče 512 bitů. Po široké kritice provedl NIST revizi návrhu a prodloužil délku klíče na 1024 bitů. I když úloha diskrétního logaritmu má již svou historii, její speciální verze použitá v DSS byla poprvé navržena Schnorrem v roce 1989 a zatím nebyla příliš široce studována. DSS je úřadem NIST patentována.

RSA

Algoritmus (1977) pro výměnu klíčů a tvorbu elektronického podpisu patřící mezi nepsané standardy. Jedná se o patentovanou (US Patent 4,405,829, 20.9.1983 vlastníkem je Public Key Partners (PKP), of Sunnyvale, California; patent vyprší po 17 letech, tj. v roce 2000). RSA není patentován mimo Severní Ameriku. Na základě využívání RSA vznikla i známá americká společnost RSA Data Security Inc. Bezpečnost RSA je založena na skutečnosti, že je obtížné rozložit velká čísla (z nichž každé je součinem dvou velkých prvočísel), závisí tedy na možnostech řešit úlohu faktorizace.

Popíšeme stručně vlastní algoritmus:

Jednotliví uživatelé si vytváří veřejný a tajný klíč pro RSA následovně:

a) nejprve náhodně (a nepredikovatelně – viz následující odstavec) si vygenerují dvě dostatečně velká prvočísla **p** a **q** (jejich přibližná velikost tj. počet bitů je zadána)

b) Spočtou **$n = pq$** a **$\Phi = (p-1)(q-1)$**.

Poznámka: stačí použít číslo **$\lambda =$ NSN (p-1, q-1)**, tj. nejmenší společný násobek čísel **p-1** a **q-1**.

c) Zvolí náhodné číslo e, kde **$1 < e < \Phi$**, tak, že **NSD (e, Φ) =1**. NSD značí největšího společného dělitele.

d) Užitím Eukleidova algoritmu spočte jednoznačně definované číslo **d** takové, že **$1 < d < \Phi$** a **$ed \equiv 1 \mod \Phi$** .

Veřejným klíčem je potom **(n,e)**, tajným klíčem uživatele je **d**.

Popíšeme nyní jak probíhá vlastní šifrování a dešifrace. Předpokládejme, že strana B zná autentický veřejný klíč strany A, kterým je **(n,e)** a zašifrovává zprávu **M** pro A. Strana B vyjádří zprávu **M** jako číslo **m**, **$0 \le m \le n-1$** (resp. posloupnost takových čísel). Dále strana B spočte

$$c = m^e \mod n$$

a zašle šifrový text straně A. Strana A nyní při dešifraci spočte pomocí tajného klíče **d**

$$m = c^d \mod n$$

Výsledkem je skutečně m, což lze dokázat následovně (např. Menezes at all.: Handbook of Applied Cryptography): Jelikož **$ed \equiv 1 \mod \Phi$,** existuje tedy k tak, že **$ed = 1 + k\Phi$**.Dále, pokud **NSD (m,p) = 1**, pak podle Fermatovy věty

$$m^{p-1} \equiv 1 \mod p.$$

Umocníme obě strany této kongruence číslem **k(q-1)** a posléze vynásobíme obě strany rovnice číslem **m**. Dostaneme

$$m^{1 + k(p-1)(q-1)} \equiv m \mod p.$$

Pokud je **NSD (m,p) = p** (druhá možná situace), pak tato rovnost platí rovněž (obě strany jsou rovny nule **mod p**). Vždy tedy

$$m^{ed} \equiv m \mod p .$$

Obdobně se dokáže **$m^{ed} \equiv m \mod q$** . Odsud plyne **$m^{ed} \equiv m \mod n$**, a tedy **$c^d \equiv (m^e)^d \equiv m \mod n$**.

RSA (Rivest-Shamir-Adleman) je v současné době asi nejrozšířenější algoritmus s veřejným klíčem. Lze ho použít pro šifrování i pro podpisy. Vhodná délka klíče je popsána v odstavci bezpečnost šifer. Podrobnější informace lze získat z článku Bruce Schneiera. Je třeba vědět, že RSA je zranitelná vůči útokům s volitelným otevřeným textem. Nový typ útoku s měřením času (timing attacks) jsme již rovněž zmiňovali. Algoritmus RSA je považován za bezpečný algoritmus, musí se však aplikovat velmi opatrně, aby bylo možné se těmto útokům vyhnout.

Matematické operace, které využívá algoritmus RSA spočívají především v násobení. Často je veřejný exponent (šifrovací) volen jako malé číslo (používá ho více uživatelů, šifrování je tedy rychlejší než dešifrace).

Ve srovnání s DES je šifrování samozřejmě podstatně pomalejší. Při softwareových realizacích se uvádí, že DES je přibližně 100 krát rychlejší než RSA, při hardwareových realizacích dokonce 1000 až 10000 krát.

RSA je součástí řady oficiálních norem. Norma ISO 9796 (International Standarts Organization) bere RSA jako kompatibilní kryptografický algoritmus, stejně tak Norma CCITT X.509 (Consultative Committee in International Telegraphy and Telephony). Je součástí normy SWIFT (Society for Worldwide Interbank Financial Telecommunications), normy ETEBAC 5 francouzského finančního průmyslu a draftu normy ANSI X9.31 pro americký bankovní průmysl. Australská norma pro správu klíčů AS2805.6.5.3 rovněž specifikuje RSA.

Diffie-Hellman:

Tento protokol k dohodě na klíči byl navržen W.Diffie a M.E.Hellmanem v jejich základním článku New Directions in Cryptography (1976). Tento protokol umožňuje dvěma uživatelům vyměnit si tajný klíč pomocí veřejných medií. Neobsahuje žádnou metodu umožňující podpis zaslané zprávy, ani není možné provést autentizaci, že daný klíč skutečně pochází od daného uživatele.

Algoritmus je považován za bezpečný, pokud je zvolen dostatečně dlouhý klíč a použit vhodný generátor prvočísel. Velikost tajného exponentu je rovněž kritická pro bezpečnost celého algoritmu. Bezpečnost Diffie-Hellmanova algoritmu závisí na obtížnosti řešení úlohy diskrétního logaritmu (obvykle je složitost této úlohy považována za ekvivalentní složitosti úlohy faktorizace, tj. nároky na velikost použitého prvočísla odpovídají nárokům na velikost n, které je součinem dvou prvočísel pro kryptosystém RSA). Je opět třeba zmínit timing attack .

Ve Spojených státech byl Diffie-Hellmanův systém pro výměnu klíčů patentován (M. E. Hellman and R. C. Merkle: Public Key Cryptographic Apparatus and Method. US Patent 4,218,582, 1980), ale patent vypršel 29.dubna 1997.

Samotný protokol probíhá následovně.

P1. Generování a zveřejnění klíčů.

Je zvoleno vhodné prvočíslo **p** a generátor **a** ε Z_p^* , které jsou opublikovány.

P2. Zprávy protokolu.

A → B: $a^x \bmod p$ (Z1)

B → A: $a^y \bmod p$ (Z2)

P3. Činnosti během provádění protokolu.

(a1) A zvolí náhodné číslo **x, 1 ≤ x ≤ p-2**, a zašle B zprávu (Z1).
(a2) B zvolí náhodné číslo **y, 1 ≤ y ≤ p-2**, a zašle A zprávu (Z2).
(b1) A spočte klíč **K** jako $K = (a^x)^y \bmod p$.
(b2) B spočte tentýž klíč jako $K = (a^y)^x \bmod p$.

El-Gamalovo schéma k dohodě na klíči (napůl certifikovaný Diffie – Hellmann):

V některých situacích je používána následující jednodušší varianta Diffie-Hellmanova protokolu.
Vlastní protokol k dohodě na klíči probíhá takto:

P1. Generování a zveřejnění klíčů.

Každý uživatel B provádí následující:

- zvolí vhodné prvočíslo **p** a generátor **a** ε Z_p^* .
- zvolí náhodné **b**, **1 ≤ b ≤ p-2** a spočte $a^b \bmod p$.
- opublikuje jako svůj veřejný klíč trojici (p, a, a^b), klíč **b** utajuje.

P2. Zpráva protokolu:

A → B: $a^x \bmod p$ (Z1)

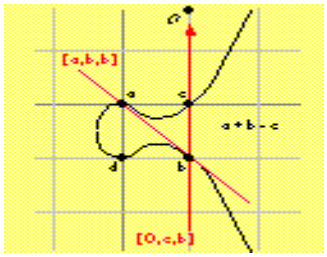
P3. Činnosti během provádění protokolu:

(a) A získá autentickou kopii veřejného klíče strany B: (p, a, a^b).
A zvolí náhodné číslo **x**, **1 ≤ x ≤ p-2**, a zašle B zprávu (Z1).
A spočte klíč **K** jako $K = (a^b)^x \bmod p$.
(b) B spočte tentýž klíč jako $K = (a^x)^b \bmod p$.

Eliptické křivky:

Řada současných systémů s veřejným klíčem je založena na využití operace umocňování ve velkých konečných matematických grupách. Kryptografická síla těchto systémů je odvozována ze složitosti řešení úlohy diskrétního logaritmu v těchto grupách, resp. z obtížnosti úlohy faktorizace velkých čísel na prvočísla. Obvykle jsou využívány grupy $\mathbf{Z_p}$ (celá čísla modulo nějaké prvočíslo **p**; v případě, kdy bezpečnost systému stojí jako pro RSA na obtížnosti úlohy faktorizace, je $\mathbf{p = r.s}$, kde **r** a **s** jsou velká prvočísla). Nejsou to však jediné možné grupy, které lze využít pro konstruování kryptografických systémů.

V roce 1985 přišli nezávisle na sobě pánové Neil Koblitz (University of Washington) a Victor Miller (tehdy IBM, Yorktown Heights) s návrhem využívat pro kryptografické účely grupy na eliptických křivkách. Bezpečnost těchto systémů opět závisí na obtížnosti řešitelnosti úlohy diskrétního logaritmu v těchto grupách. Dokonce nejsou pro tyto logaritmy známy žádné subexponenciální algoritmy (jako pro klasický diskrétní logaritmus, resp. úlohu faktorizace), nejlepší algoritmy mají plně exponenciální charakter (platí pro obecné křivky, nikoliv však pro některé speciální podtřídy eliptických křivek).



Primární výhodou kryptosystémů na bázi eliptických křivek je jejich velká kryptografická bezpečnost vzhledem k dané velikosti klíče. Význačně kratší délka klíčů (např. oproti RSA) vede ke kratším certifikátům i menším parametrům systému a tedy i k větší výpočetní efektivnosti algoritmů. Druhá výhoda je v tom, že fakticky všechna již známá použití v systémech na bazi diskrétního logaritmu (kryptografické protokoly, ElGamalův podpis atd.) lze převést do systémů na bázi eliptických křivek. To se podařilo zejména při převodu normy DSA na ECDSA.

V praxi jsou používány algoritmy pro eliptické křivky v tělesech $\mathbf{F_p}$, kde **p** je rovno buď prvočíslu nebo číslu $\mathbf{2^m}$ pro nějaké přirozené číslo m.

Eliptické křivky v polích GF (2^m)

V tělesech, kde je $\mathbf{p = 2^m}$ existují v zásadě dva typy eliptických křivek. Prvním typem jsou tzv. supersingulární křivky, které mají rovnici:

$$\mathbf{y^2 + cy = x^3 + ax + b}.$$

Supersingulární křivky mají tu výhodu, že pro ně snadným výpočtem lze stanovit řád křivky. Jak ukázali A.J. Menezes, T. Okamoto and S. Vanstone (MOV attack) lze vhodnou konstrukcí úlohu diskrétního logaritmu pro tyto eliptické křivky převést na úlohu klasického diskrétního logaritmu (s určitým koeficientem **B**). Proto také tyto křivky v současnosti nejsou navrhovány pro konstrukci kryptosystémů.

Druhým typem jsou tzv. nonsupersingulární křivky s rovnicí:

$$\mathbf{y^2 + xy = x^3 + ax^2 + b}.$$

Obecně platí, že pro tyto křivky je matematicky dosti obtížnou úlohou stanovení jejich řádu. Existující algoritmy jsou značně náročné matematicky i výpočetně.

Pro praktická použití je zvažována také třída tzv. křivek s komplexním násobením (CM křivky).Pro tuto třídu existuje jednoduchá metoda výpočtu počtu bodů křivky (řád křivky).

Koblitz, Neal: CM-curves with good cryptographic properties, Advances in Cryptology: Proceedings of Crypto ´91, Lecture Notes in Computer Science, Vol.576, Springer-Verlag, Berlin, 1992, pp. 279-287.

Speciální podtřídou jsou tzv. Koblitzovy křivky. To jsou křivky $\mathbf{E_0}$ a $\mathbf{E_1}$ definované rovnicí:

$$E_a : \quad y^2 + xy = x^3 + ax^2 + 1.$$

Pro tyto křivky platí následující způsob výpočtu počtu bodů křivky.
Nechť α je kořen charakteristického polynomu $T^2 - T + q$, tj. $\alpha = (1 + \sqrt{(1 - 4q)})/2$. Potom pro $a = 1$ označíme počet bodů křivky jako N_m, pro $a = 0$ jako M_m. Výpočet těchto hodnot probíhá pomocí Fibonacciovy posloupnosti a_m. Přitom $a_0 = 2$, $a_1 = 1$, $a_{m+1} = a_m - qa_{m-1}$ pro $m = 2,\ldots$. Pak $N_m = q^m + 1 - a_m$ a $M_m = q^m + 1 + a_m$.
Co činí Koblitzovy křivky atraktivními je kromě snadnosti výpočtu řádu křivky také fakt, že pro tyto křivky jejich kardinalita může být pro určitá m jednoduše rozložitelná (kromě velkého prvočísla rozklad řádu obsahuje již jen velmi malé číslo).
Pro $a = 1$ je řád křivky roven dvojnásobku velkého prvočísla pro následující m:
$m = 3, 5, 7, 11, 17, 19, 23, 101, 107, 109, 113, 163, 283, 311, 331, 347, 359, \ldots$.
Pro $a = 0$ je řád křivky roven čtyřnásobku prvočísla pro následující m:
$m = 5, 7, 13, 19, 23, 41, 83, 97, 103, 107, 131, 233, 239, 277, 283, 349, 409, \ldots$.
V článku

Solinas, Jerome A.: An Improved Algorithm for Arithmetic on a Family of Elliptic Curves, CRYPTO ´ 97, pp. 357-371,

je pro Koblitzovy křivky popsána metoda efektivní implementace základních algoritmů (sčítání, násobení bodu číslem) resp. vylepšení jejich výpočetních vlastností.

V praxi jsou ještě využívány křivky, jejichž kardinalita (počet bodů na křivce) je počítána užitím tzv. Weilovy věty. Řád křivky $y^2 + xy = x^3 + ax^2 + b$ nad tělesem $F(2^{lk})$ lze spočítat jako

$$E = 2^{lk} + 1 - \mathrm{LUCAS}\,(2^l - (e - 1), 2^{lk}, K).$$

Zde e je řád eliptické křivky nad tělesem $F(2^l)$ a funkce $\mathrm{LUCAS}\,(r,s,k)$ je definována rekurentním vztahem

$$V(r,s,0) = 2,\ V(r,s,1) = r \quad \text{a} \quad V(r,s,k) = r\,V(r,s,k-1) - sV(r,s,k-2).$$

Zde a a b musí být prvky tělesa $F(2^l)$. Dále - E je dělitelné e. Pokud l je dostatečně malé lze e spočítat „hrubou silou“. Často je pak E/e velké prvočíslo a můžeme potom zvolit vhodný pevný bod P této křivky (zvolíme libovolný bod R a spočteme $P = eR$, musí být P různé od nuly). Bod P má pak řád E/e.
Číslo k musí být prvočíslo. K užití Weilovy věty je třeba určit podtěleso tělesa $F(2^{lk})$ řádu 2^l.
Jestliže $\alpha \in GF(2^m)$, potom stopou α nazýváme číslo

$$Tr(\alpha) = \alpha + \alpha^2 + \alpha^{2^2} + \ldots + \alpha^{2^{m-1}}.$$

$Tr(\alpha)$ je v polovině případů rovna 0, v polovině je rovna 1. Pro malá l lze řád eliptické křivky křivky $y^2 + xy = x^3 + ax^2 + b$ nad tělesem $F(2^l)$ spočítat následovně. Nechť

$$\mu = (-1)^{Tr(a)}.$$

$$\lambda(x) = Tr(x + b/x^2).$$

kde x je nenulové. Potom

$$e = 2^d + 1 + \mu \sum_{x \neq 0} (-1)^{\lambda(x)}.$$

Eliptické křivky v polích GF(p).

Ukazuje se, že pro softwareové realizace je výhodnější používat eliptické křivky definované v prvočíselném poli, tj. v poli GF(p), kde p je prvočíslo. Příslušné eliptické křivky ($p > 3$) zde mají následující Weierstrassovu rovnici:

$$y^2 = x^3 + ax + b,$$

kde **a** a **b** jsou celá čísla mod **p** pro něž $4a^3 + 27b^2 \not\equiv 0 \pmod p$.
Nejobtížnější (matematicky) částí celého postupu je výpočet řádu (počtu bodů) příslušné eliptické křivky. Platí následující tzv. Hasseho meze pro hodnotu n řádu eliptické křivky:

$$p - 2\sqrt{p} + 1 \leq n \leq 2\sqrt{p} + 1$$

Jestliže **E** není supersingulární křivkou nad **GF(q)** - zde **q** může být jak prvočíslo **p**, tak číslo 2^m - s řádem **u**, pak

$$Z = 4q - (q + 1 - u)^2,$$

je kladným číslem (vzhledem k tomu, že platí Hasseho meze). Tudíž existuje jednoznačná faktorizace:

$$Z = DV^2$$

kde **D** obsahuje pouze činitele bez čtverců. Vzhledem k tomu pro každou nesupersingulární elitpickou křivku nad **GF(q)** existuje jediné kladné číslo **D** (neobsahující ve svém rozkladu čtverce) tak, že

$$4q = W^2 + DV^2, \qquad \text{(R1)}$$

a

$$u = q + 1 \pm W, \qquad \text{(R2)}$$

pro nějaká **W** a **V**. V takovémto případě říkáme, že pro E existuje komplexní násobení vzhledem k **D** (správněji vzhledem k $\sqrt{(-D)}$). Číslo **D** je nazýváno CM diskriminantem pro **q**.

Pokud pro danou křivku E známe její diskriminant **D**, pak ze vztahů (R1) a (R2) spočteme snadno řád odpovídající eliptické křivky. Je tedy vhodné nalézt takové eliptické křivky, jejichž řád u splňuje tyto rovnice s malou hodnotou D.

Adekvátní CM technika pro hledání řádu křivek se nazývá Atkin-Morainova metoda (v poli **GF(p)**), resp. Lay-Zimmerova metoda (v poli $\mathbf{GF(2^m)}$).

Potřebujeme tedy následující, nalézt vhodný řád a zkonstruovat křivku mající tento řád. Resp. začínáme volbou pole o velikosti q, minimálním řádem r_{min} a volbou meze (pro pokusné dělení) l_{max}. Při daných těchto číslech říkáme, že D je vyhovující, pokud existuje eliptická křivka nad polem GF(q) s CM pro D a mající přibližně prvočíselný řád (tj. řád je součin velkého prvočísla a dalšího dostatečně malého čísla).

Celý postup obsahují následující dva kroky:

Krok 1.: Nalezení přibližně prvočíselného řádu.

Najdeme vhodné D. Pokud je toto nalezeno, je zaznamenáno D, velké prvočíslo r a přirozené číslo k tak, že u = kr je přibližně prvočíselný řád (viz annex A k draftu P1363, odstavce A.14.2 a A.14.3, kde je popsán podrobný postup).

Krok 2.: Konstrukce vhodné křivky a bodu.

Při daných D, k a r zkonstruovat eliptickou křivku nad GF(q) a bod řádu r (viz annex A k draftu P1363, odstavce A.14.4 a A.14.5, kde je popsán podrobný postup).

Problematika bezpečnosti kryptologických systémů na bázi eliptických křivek je dnes stále ještě poněkud méně prozkoumanou oblastí kryptologie. Pro RSA např. lze na základě již dlouhodobě vyvíjených algoritmů poměrně exaktně zhodnotit současnou i perspektivní odolnost tohoto systému. Existující algoritmy mají subexponenciální charakter a (samozřejmě pokud nedojde k nějakému revolučnímu objevu např. pokud se nepodaří zrealizovat kvantový počítač) vylepšování jejich možností jde v zásadě ruku v ruce se zvětšující se výpočetní silou, kterou má lidstvo k dispozici. Ve stejné situaci je i řešení úlohy diskrétního logaritmu v konečných tělesech.

Vlastní bezpečnost kryptosystémů na bázi eliptických křivek se odvíjí z výpočetní složitosti metod pro řešení diskrétního logaritmu pro eliptické křivky.
Řada kryptologů je zatím skeptická k zavádění eliptických křivek do praxe. Poukazují zejména na malý stupeň poznání problematiky. Samotná praxe však předbíhá tuto skepsi. Bez řešení konkrétních úloh souvisejících jak s konkrétními implementacemi ECC (hledání vhodných křivek, určování jejich mohutnosti, posuzování bezpečnosti konkrétního ECC atd.) nelze předpokládat ani rozvoj příslušné teoretické baze. Určitou roli zde může hrát i setrvačné trvání na RSA (i když právě RSA comp. je jednou ze společností, která rozvíjí kryptosystémy na bazi eliptických křivek). Naopak pozitivní roli zde sehrává zejména snaha zavést obecnou normu na bazi ECC – ECDSA (IEEE P1363 Draft Standard).

Při určování bezpečnosti (konkrétních) kryptosystémů na bázi eliptických křivek je nutné vycházet ze současného stupně poznání problematiky, resp. z některých možných extrapolací pro nejbližší budoucnost. Draft X9.62 obsahuje určité konkrétní matematické podmínky:

1. Řád použité křivky (nikoliv tělesa, v kterém je křivka definována) by měl být dělitelný velkým prvočíslem $\mathbf{n > 2^{160}}$;
2. Měly by být splněny podmínky MOV a Anomalous (tj. ověřit, že daná křivka není supersingulární ani anomální)

Eliptické křivky navrhované pro praktické použití jsou dnes především dvojího druhu. Jednak křivky s náhodně vygenerovanými parametry a jednak jsou pro praktické použití zvažovány křivky s komplexním násobením (viz také Certicom Chalenge).

Pro oblasti, kde je otázka utajení zvláště citlivá (vojenství, bezpečnost státu) požadovaná velikost mohutnosti n (největšího prvočísla dělícího řád křivky) eliptické křivky vzrůstá ze **160** bitů na **180**.

Rozpracovávané normy

1.IEEE, P1363 - Eliptické křivky jsou uvedeny v návrhu normy IEEE P1363 (Standard pro kryptografii s veřejným klíčem), který zahrnuje mechanismy šifrování, digitálních podpisů a dohody na klíči. Jsou podporovány eliptické křivky jak nad $\mathbf{F_p}$, tak i nad $\mathbf{F(2^m)}$.
2.ANSI X9.62 - Algoritmus pro digitální podpis pomocí eliptických křivek (Americal National Standard X9.62-199x, Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm neboli ECDSA) je návrh normy pracovní skupinou X9F1. ECDSA popisuje jak generovat digitální podpis dat podepisující stranou a jak ověřující stranou zverifikovat autentičnost tohoto podpisu. Rovněž jsou podporovány eliptické křivky jak nad $\mathbf{F_p}$, tak i nad $\mathbf{F(2^m)}$.
3.ANSI X9.63 - Transportní protokoly a dohoda na klíči s využitím eliptických křivek (Americal National Standard X9.63, Public Key Cryptography For The Financial Services Industry: Elliptic Curve Key Agreement and Transport Protocols), X9.63, je návaznou položkou skupiny X9F1. Tento materiál se zabývá problematikou ustavení klíčů v kryptografických systémech s využitím aparátu eliptických křivek.

Na webovské stránce Marca Joye lze nalézt řadu dalších referencí a adres k problematice eliptických křivek. Některé praktické aspekty kryptografie na bázi eliptických křivek posuzuje materiál G.Barwooda.

Aplikace systémů s veřejným klíčem:

Dohoda na tajném klíči (pro symetrickou šifru) s využitím kryptosystému s veřejným klíčem na bázi eliptických křivek:

Východiska:

1) Strany A a B se dohodli na užívaní kryptosystému s eliptickou křivkou s parametry **(q, a, b, r, G)**. Zde:
   q je velikost tělesa ve kterém je křivka definována,
   a, b jsou paramery v rovnici křivky,
   G je zvolený bod na eliptické křivce,

r je prvočíselný dělitel řádu křivky.
Tyto parametry jsou veřejné.

2) Dále strana A zvolí svůj soukromý klíč s a spočte svůj veřejný klíč **U = sG**.
Obdobně strana B zvolí svůj soukromý klíč t a spočte svůj veřejný klíč **V = tG**.
3) Strana A spočte sdílenou tajnou hodnotu z následovně:
Nejprve spočte bod **P = sV**, potom $\mathbf{z} = \mathbf{x}_P$ (**x**-tá souřadnice bodu **P**).
4) Stran B si obdobně spočte sdílenou tajnou hodnotu z takto:
Spočte bod **Q = tU** a pak $\mathbf{z} = \mathbf{x}_Q$ (**x**-tá souřadnice bodu **Q**) - neboť platí **Q = tU = tsG = s (tG) = sV = P**.
5) Strany A a B jsou dohodnuty na hodnotách parametrů $\mathbf{h}_i$ (lze i veřejnou cestou).
6) K výpočtu hodnot jednotlivých klíčů je použita funkce pro odvození klíčů:

$$\mathbf{K}_i = \mathbf{hash}\ (\mathbf{z} \,||\, \mathbf{h}_i)$$

Jako hashovací funkce je použita např. SHA-1.

Poznámka: V popisu protokolu jsou vynechány některé detaily - formáty a délky jednotlivých veličin.

Náhodná čísla v kryptografii:

Kryptografické systémy potřebují náhodná čísla. Tato čísla musí však kromě požadavků náhodnosti (vyhovění běžným statistickým testům) splňovat i podmínku nepredikovatelnosti. Tj. tato kryptograficky bezpečná náhodná čísla nesmí být předvidatelná potenciálním narušitelem. Náhodná čísla jsou obvykle využívána jako tajné klíče pro symetrickou šifru pro konkrétní spojení (session key) a jejich kvalita je rozhodující pro kryptologickou odolnost celého systému. Generátor náhodných čísel lze snadno analyzovat a může se stát nejslabším bodem systému. Některé počítače proto mají speciální hardwareové šumové generátory (šum diod, tranzistorů, nejméně význačné bity audio vstupů, intervalů mezi přerušeními) jsou dobrými zdroji náhodnosti, zejména pokud jsou zpracovány vhodnou hashovací funkcí. Je nanejvýš vhodné užívat skutečný náhodný šum kdykoliv je to možné. Příklad kryptografického náhodného generátoru lze nalézt např. v zdrojovém kódu PGP.

Softwareový mechanismus produkce náhodných čísel má tu vlastnost, že pokud je dána vstupní inicializační hodnota, pak je jí určena celá následující posloupnost "náhodných čísel". Těmto číslům říkáme proto také **pseudonáhodná** na rozdíl od čísel reálně náhodných (vzniklých např. na základě fyzikálního generátoru). Pseudonáhodnost má výhodu opakovatelnosti, často je tak využívána při analýze statistických vlastností různých procesů.

Konstruovat generátory **kryptografických náhodných čísel** není jednoduchou záležitostí. Pokud není k dispozici fyzikální zdroj náhodnosti (což je obvyklá situace pro softwareová řešení šifer), je třeba zvažovat ostatní potenciální zdroje náhodnosti. Těmi mohou být např. různé síťové statistiky, statistiky některých procesů, délky intervalů mezi stiskem kláves a jiné. Obvykle se vytváří jakýsi zdroj či zásoba (pool) v délce alespoň 128 (160) bitů obsahující "náhodnost". S tímto zdrojem náhodnosti se pak dále pracuje, obvykle se připojí nějaká další data a na celý blok je použita hashovací funkce. Lze také použít nějaký kryptografický algoritmus, kde klíč je vytvořen na základě náhodných hodnot z naší zásoby náhodných čísel. Část takto získaného výstupu je opět "zamixována" do naší zásoby. Konstrukci generátoru kryptografických náhodných čísel je třeba věnovat velkou pozornost, pokud v ní budou nějaké nedostatky, mohou sa tato čísla snadno stát největší slabinou našeho systému.

Úloha faktorizace:

Faktorizací celého čísla rozumíme jeho rozklad na součin menších čísel (faktorů). Například faktory čísla 299 jsou čísla 13 a 23. Prvočíselnou faktorizací rozumíme takovou faktorizaci, kde všechny faktory jsou prvočísla. Každé celé číslo má jednoznačnou prvočíselnou faktorizaci.

Z výpočetního hlediska je násobení celých čísel jednoduchou úlohou. Zdaleka to však již neplatí o úloze obrácené, tj. o úloze faktorizace. Existuje celá řada faktorizačních technik. Mezi nejznámější patři:

- Eratosfenovo síto, pokusné dělení
- Pollardova **ρ**-metoda s výpočetní složitostí **O(sqrt(p))**.
- Pollardova **p-1** metoda se složitostí **O(q)**, kde **q** je největší prvočíselný činitel rozkladu **p-1**.
- polynomiální kvadratické síto (multiple polynomial quadratic sieve) se složitostí **O(exp (sqrt (ln n ln ln n)))**.
- síto číselného pole pro obecná čísla (general number field sieve), příslušná výpočetní složitost je dána vztahem

$$\mathbf{C} = \mathbf{K}\ \mathbf{2}^{\mathbf{L(p)}},$$

kde

$$\mathbf{L(p)} = \sqrt[3]{(((64/9)\ (\log_2 p)\ (\log_2 (\ln p))^2)}.$$

(a současný odhad **C** je roven 4.4066).

- faktorizační metoda využívající eliptické křivky (složitost metody je **O(exp (sqrt (2 ln p ln ln p))))**.

Neustálý vývoj výpočetní techniky na jedné straně a zdokonalování používaných faktorizačních algoritmů na straně druhé vede k přehodnocování možností faktorizačních algoritmů. V současné době je v dosahu možností faktorizace čísel, která mají řádově 130 až 140 dekadických míst (pro speciální čísla, např. čísla tvaru $2^n - 1$ existují ještě efektivnější faktorizační algoritmy). Proto také schéma RSA s modulem o délce 512 bitů není již považováno za bezpečné, doporučuje se délky 768, 1024 atd.

Úloha diskrétního logaritmu:

Úlohou diskrétního logaritmu rozumíme úlohu nalézt neznámou hodnotu exponentu **x** z rovnice $y = g^x \bmod p$. Existují i jiné obecnější formulace této úlohy. Pro řešení tohoto matematického problému nejsou známy algoritmy s vhodnou (např. polynomiální) výpočetní složitostí. Všeobecně je proto považováno řešení úlohy diskrétního logaritmu za obtížné a na bázi úlohy DL je formulována řada kryptografických systémů (El-Gamal, DSS atd.).

Pro podrobnější informaci odkazuji na článek autorů Brian A. LaMacchia and Andrew M. Odlyzko, Computation of Discrete Logarithms in Prime Fields, Designs, Codes and Cryptography 1 (1991), 47-62. Z jejich závěrů vyplývá, že při provedení určitých předběžných výpočtů lze počítat diskrétní logaritmus efektivněji. Při srovnání s úlohou faktorizace (rozklad složeného čísla téže velikosti) je ukázáno, že úloha diskrétního logaritmu má srovnatelnou nebo nepatrně vyšší složitost. Historicky je skutečností, že jakýkoliv pokrok při řešení jedné úlohy (úlohy diskrétního logaritmu či úlohy faktorizace) bylo možné aplikovat na druhou úlohu. Ukazuje to, že složitost obou problémů úzce souvisí .

Využitím metody síta číselného tělesa (number field sieve) pro úlohu diskrétního logaritmu se zabývá článek Damian Weber, "An Implementation of the General Number Field Sieve to Compute Discrete Logarithms mod p". Advances in Cryptology - EUROCRYPT '95, Louis C. Guillou and Jean-Jacques Quisquater, editors. Lecture Notes in Computer Science, volume 921, Springer-Verlag, New York, 1995. Pages 95--105. Obecná metoda síta číselného tělesa má přibližně tutéž složitost jako odpovídající faktorizační algoritmus:

$$C = K\, 2^{L(p)},$$

kde

$$L(p) = \sqrt[3]{((64/9)\,(\log_2 p)\,(\log_2 (\ln p))^2)}.$$

Pro praktická užití je doporučovano použit prvočíslo s délkou větší než 512 bitů, nejlépe 1024 bitů.

Úloha diskrétního logaritmu pro eliptické křivky:

Oproti situaci při hodnocení výpočetní složitosti klasického diskrétního logaritmu má zatím vývoj algoritmů pro řešení diskrétního algoritmu pro eliptické křivky poměrně krátkou historii. Především je třeba poukázat na skutečnost, že pro úlohu není znám žádný subexponenciální logaritmus (jako je tomu v případě obvyklého diskrétního logaritmu). Pro úlohu v obecné podobě je nejlepším známým algoritmem Pollardova **ρ**-metoda, jejíž složitost je $\sqrt{(\pi n/2)}$, každým jednotlivým krokem je součet na eliptické křivce. Číslo **n** je zde největší prvočíselný dělitel velikosti grupy eliptické křivky. Při paralelizaci na w procesorech je očekávaný počet kroků (než získáme jeden diskrétní logaritmus) roven $\sqrt{(\pi n/2)}/w$.

V článku (Michael Wiener a Robert Zuccherato: Faster Attacks on Elliptic Curve Cryptosystems) a článku (Robert Gallant, Robert Lambert a Scott Vanstone: Improving the parallelized Pollard lambda search on binary anomalous curves) jsou analyzovány určitá vylepšení Pollardovy metody při výpočtu eliptického diskrétního logaritmu. Tyto výsledky však mají charakter pouze částečného vylepšení a v zásadě neovlivňují posuzování náročnosti výpočtu diskrétního logaritmu pro eliptické křivky.

Existují ale speciální situace, kdy se úloha stává výpočetně jednodušší. Jedná se o situace, kdy je použitelný MOV attack (pro supersingulární eliptické křivky) anebo jestliže platí tzv. Anomalous condition (počet bodů křivky je roven počtu bodů tělesa, ve kterém je křivka definována).

Teorie čísel v kryptografii:

Současné kryptosystémy s veřejným klíčem se opírají o řadu výsledků teorie čísel. Existuje celá řada materiálů, z nichž lze načerpat vhodné informace. Některé z nich jsou přístupné i na Internetu. Uvedu alespoň následující: Annex A:"Number-Theoretic Background." normy P1363, Certicom ECC Tutorials and Whitepapers, kurs 2D1440, Advanced Algorithms a dlouhá řada jednotlivých publikací.

Testy prvočíselnosti:

Prvočíslo - přirozené číslo, které je dělitelné pouze jedničkou a samým sebou. Například prvních několik prvočísel je 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,… . Existuje nekonečně mnoho prvočísel, jak ukázal již Eukleidos.

Eratosfenovo síto : Je to nejstarší písemně zadokumentovaná metoda vyhledávání prvočísel (asi rok 250 př.n.l.). Metoda vyhledává všechna lichá prvočísla, která jsou menší než zadané přirozené číslo **p**, pomocí následujících pravidel :

1) Vypiš všechna lichá čísla od 3 do **p**,

2) Vyškrtni 3^2, a dále každé třetí číslo, potom vyškrtni 5^2 a dále každé páté číslo,

3) Pokračuj tak dál, až první zbývající číslo, které následuje za tím, jehož násobky byly vyškrtnuty v předcházejícím kroku, již nemá druhou mocninu menší než **p**.

Čísla, která nebyla vyškrtnuta, jsou všechna prvočísla menší než zadané **p**. Výpočetní složitost tohoto algoritmu je řádově **p log log p**.

Pokusné dělení : Pravděpodobně Leonardo z Pisy (1202) byl první, kdo v opublikované práci konstatoval, že k prověrce zadaného čísla **p** na prvočíselnost stačí toto číslo vydělit všemi čísly menšími či rovnými $\sqrt{\mathbf{p}}$. Také první uvedl tabulku prvočísel od 11 do 97.

Fermatův test prvočíselnosti:

Fermatova (Malá) věta: Jestliže **p** je prvočíslo a **a** je libovolné přirozené číslo, pak $\mathbf{a^p \equiv a \bmod p}$.
Na základě této věty máme následující test prvočíselnosti. Máme dané $\mathbf{n > 1}$, zvolíme $\mathbf{a > 1}$ a spočteme $\mathbf{a^{n-1} \bmod n}$. Pokud výsledek je různý od jedné, pak **n** není prvočíslo. Pokud však výsledek je roven jedné, pak to ještě neznamená, že **n** je prvočíslo. Můžeme pak vzít jiné **a** a provést celý test znovu. Ukázalo se však, že dokonce existují taková n (která nejsou prvočísly), pro která Fermatův test je splněn při libovolné bazi **a**. Tato složená čísla se nazývají Carmichaelova čísla. Carmichaelova čísla menší než 100 000 jsou následující:
561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, 46657, 52633, 62745, 63973 a 75361.

Carmichaelových prvočísel (i když jsou vzácná) je nekonečně mnoho. (R.Pinch uvádí seznam všech Carmichaelových čísel menších než 10^{16}).

Solovay-Strassenův test prvočíselnosti:

Předpokládejme, že **n** je liché přirozené číslo a tážeme se, zda je to prvočíslo či číslo složené. Zvolme **k** celých čísel $\mathbf{0 < b < n}$ náhodným způsobem. Pro každé b spočteme nejprve obě strany vzorce

$$\mathbf{J(b/n) = b^{(n-1)/2} \bmod n}.$$

Spočtení levé strany $\mathbf{b^{(n-1)/2}}$ trvá $\mathbf{O(\log^3 n)}$ bitových operací při použití metody opakovaného umocňování; spočtení Jacobiho symbolu trvá rovněž $\mathbf{O(\log^3 n)}$ bitových operací. Nejsou-li obě strany kongruentní **mod n**, pak víme, že **n** je složené číslo a test končí. V opačném případě přicházíme k dalšímu **b**. Pokud vzorec platí pro všech **k** náhodně vybraných **b**, pak pravděpodobnost, že složené **n** splní všechny tyto testy, je nejvýše $\mathbf{1/2^k}$. Tedy Solovay-Strassenův test je pravděpodobnostní algoritmus, který vede k závěru, že **n** je buď složené číslo, nebo k závěru, že **n** je "pravděpodobné" prvočíslo.
Poznámka: (**b** | **n**) značí tzv. Legeandrův symbol, který je roven (n je prvočíslo):

0 pokud **n** | **b** (**b** dělí **n**),
1 pokud **b** je kvadratickým reziduem **mod n**,
-1 pokud **b** je kvadratickým non-reziduem **mod n**.

Jacobiho symbol **J(b/n)** je zobecněním Legeandrova symbolu na neprvočíselná n. Jestliže rozložíme **n** na odpovídající prvočíselné činitele (i s příslušnými mocninami), pak Jacobiho symbol získáme jako součin Legeandrových symbolů odpovídajících prvočísel (opět včetně příslušných mocnin).
Číslo **b** je kvadratickým reziduem **mod n**, jestliže existuje **a** tak, že $\mathbf{a^2 \equiv b \bmod n}$. Číslo **b** je kvadratickým non-reziduem **mod n** v opačném případě (tj. jestliže příslušné **a** neexistuje).

Miller-Rabinův test primality

Popíšeme nyní Miller-Rabinův test na prvočíselnost. Předpokládáme, že chceme určit zda velké kladné liché celé číslo **n** je prvočíslo, nebo zda je to číslo složené. Napíšeme $\mathbf{n - 1 = 2^r t}$, kde **t** je liché a najdeme náhodné celé číslo **b**, $\mathbf{0 < b < n}$. Nejprve spočteme $\mathbf{b^t \bmod n}$. Pokud dostaneme jedničku, pak **n** vyhovělo tomuto testu pro naše speciální **b**, a přecházíme k jinému náhodně vybranému b. V opačném případě výsledek ($\mathbf{b^t \bmod n}$) umocníme na druhou, tento opět umocníme na druhou **mod n**, atd. dokud nedostaneme **-1**. Pokud jsme dostali **-1**,

pak **n** prošlo testem. Jestliže však nikdy nedostaneme **-1**, tj. v kroku **r+1** dostaneme **1** a v kroku **r** jsme neměli **-1 mod n**, pak **n** nevyhovělo tomuto testu a víme, že **n** je složené číslo. Jestliže **n** splní test pro všechny naše náhodné volby **b** (předpokládejme, že jsme zkoušeli **k** bazí **b**) potom **n** je složené číslo s pravděpodobností menší než je $\mathbf{1/4^k}$. To je proto, že nejvýše **1/4** bazí **0 < b < n** může vyhovět tomuto testu. Tento test je poněkud lepší než Solovay-Strassenův, kde obdobný odhad dává pravděpodobnost $\mathbf{1/2^k}$.

Dokazování prvočíselnosti:

Kromě pravděpodobnostních algoritmů k testování prvočíselnosti existují i postupy, které umožňují poněkud více. V případě, že p je skutečně prvočíslo, pak existují algoritmy, které toto dokáží. Toto umožňuje **Cohen-Lenstrův test** (viz seznam literatury) a **Atkin-Morainův test.** Atkin-Morainův test je implementován v softwareovém balíku ECPP, který je dostupný na Internetu (včetně on-line služby k prověřování prvočísel). Algoritmus zároveň slouží jako certifikace prvočíselnosti (tj. jeho výstupem je i vygenerovaná speciální posloupnost prvočísel, která umožňuje nezávislému pozorovateli ověřit fakt prvočíselnosti zadaného čísla).
Program ověří např. prvočíselnost 320 bitového čísla za dobu menší než dvě minuty. Jeho složitost (heuristicky) je dána výrazem $O((\log n)^{6+\varepsilon})$ pro nějaké $\varepsilon>0$.

Další informace k problematice prvočísel lze nalézt např. na tzv. The Prime Page nebo na stránce History of Prime Numbers.

Související matematické problémy:

NP=P?:

Polynomiální algoritmy jsou algoritmy, pro které čas k jejich provedení je dán v podobě polynomiální funkce vstupu úlohy či může být takovýmto polynomem shora omezen. Pro většinu praktických úloh řešených současnými počítači existují polynomiální algoritmy (např. řešení soustav lineárních rovnic, řešení kubické a bikvadratické rovnice, v posledních letech byly dokázána polynomiální řešitelnost úlohy lineárního programování, atd.).

Nedeterministické polynomiální algoritmy, jsou algoritmy, kde výpočet může být proveden v polynomiálním čase na nedeterministickém počítači. Takovýto počítač (po určení správného řešení oráklem) ověří správné řešení v polynomiálním čase.

Typickou úlohou řešitelnou nedeterministickým polynomiálním algoritmem je úloha splnitelnosti Booleovského výrazu. Pokud známe správnou hodnotu, lze ji v polynomiálním čase (dosazením správných hodnot za jednotlivé proměnné) ověřit.

Řada známých složitých problémů z NP (problémů řešitelných nedeterministickým polynomiálním algoritmem) má jednu význačnou vlastnost - tyto problémy jsou vzájemně převoditelné (tj. existuje tzv. polynomiální redukce umožňující převod jedné úlohy na druhou). Pokud by byl nalezen polynomiální algoritmus pro řešení jediné z těchto úloh, pak by bylo možné pomocí polynomiálního algoritmu řešit všechny tyto úlohy. Tuto třídu úloh nazýváme **NP-úplnými problémy**.

Známý problém **NP=P?** spočívá právě v otázce, zda existuje polynomiální algoritmus pro řešení NP-úplných úloh. Zatím nikomu se nepodařilo takovýto algoritmus nalézt, nikdo však také nedokázal, že takovýto algoritmus neexistuje. Přesto právě existence třídy NP-úplných úloh vede k dosti rozšířenému přesvědčení, že třída NP je širší než třída P (třída úloh řešitelných v polynomiálním čase).

Kvantové počítače:

Kvantová mechanika v sobě obsahuje potenciál, který může přinést revoluci i v oblasti, kde se to ještě nedávno nezdálo být možné - ve výpočetní technice, v konstruování počítačů založených na principech odlišných od klasických. Všechny současné počítače jsou z výpočetního hlediska ekvivalentní klasickému Turingovu modelu. Využití kvantových fenoménů pro oblast výpočetní techniky má smysl samozřejmě pouze tehdy pokud mohou kvantové výpočty přinést novou etapu existence počítačů. Tím se míní fakt, že počítače konstruované na nových principech by měli přinést možnost řešit úlohy, které na klasické výpočetní technice řešit neumíme nebo umíme řešit pouze v určitých případech (úlohy s malou výpočetní složitostí, úlohy malé svým rozsahem).

Dnes je známa řada úloh, jejichž řešení je pro klasické počítače problematickou resp. více či méně nemožnou záležitostí. Typickým příkladem jsou NP-úplné problémy. Přestože hypotéza NP=P nebyla dodnes dokázána ani vyvrácena, všeobecně se věří, že výše uvedená rovnost neplatí. V kryptografii se právě obtížně řešitelné úlohy používají ke konstruování šifrových systémů. O obtížnost řešení NP-úplných úloh se opírají používané blokové šifry (DES atd.).

Kromě NP-úplných problémů ale existují i jiné úlohy jejichž řešitelnost na současných počítačích je obtížná a tyto úlohy jsou rovněž využívány v kryptografii. V podstatě téměř všechny systémy s veřejným klíčem počínaje RSA a Diffie-Hellmanovým systémem výměny klíčů jsou založeny na známé skutečnosti obtížné řešitelnosti úlohy diskrétního logaritmu a úlohy faktorizace velkých čísel.

Již vlastně od vzniku idejí vedoucích k myšlenkám k realizaci počítačů na kvantové bazi zde byla zjevná (a pochopitelná - v opačném případě jaký smysl by kvantové počítače měly?) snaha o nalezení úloh, které jsou kvantovými počítači řešitelné význačně lépe než klasickými počítači.

Tato snaha byla v roce 1994 korunována Shorovým výsledkem o existenci kvantového polynomiálního algoritmu pro řešení úloh diskrétního logaritmu a úloh faktorizace velkých čísel. Pokud by dnes již existoval kvantový počítač, pak by bylo v podstatě nezbytné přestat používat většinu systémů s veřejným klíčem (tyto systémy pro některé své praktické přednosti jsou v současnosti hojně využívány - např. právě RSA).

Shorův algoritmus využívá kvantových principů ke konstrukci diskrétní Fourieovy transformace umožňující odhadnout periodičnost určitých funkcí. Výpočet délek obřích period pak umožňuje buď výpočet diskrétního logaritmu nebo nalézt neznámé faktory známého velkého čísla.

V základě všeho je pojem kvantového bitu (qubitu), který podle kvantové mechaniky může být v lineární superpozici dvou klasických stavů. Heisenbergův princip neurčitosti formuluje základní vlastnosti tohoto qubitu - principiální nemožnost (obecnou) měření těchto qubitů.

Východiskem algoritmů kvantového počítače jsou tzv. unitární transformace pracující s vektory qubitů. Na rozdíl od transformací probíhajících v klasickém počítači jsou unitární transformace vždy reversibilní, tj. vždy existuje možnost jít algoritmem pozpátku.

Pomocí diskrétní Fourieorovy transformace (užitečný je například Coppersmithův algoritmus pro provedení aproximace Fourierovy transformace na kvantových počítačích) a některých faktů z teorie čísel Shor zkonstruoval (zatím teoreticky) efektivní polynomiální kvantový algoritmus pro řešení úloh faktorizace a diskrétního logaritmu. Groverův algoritmus ukazuje, že pomocí kvantových počítačů existují výrazně efektivnější techniky pro vyhledávání v rozsáhlých databázích. Pokud připustíme existenci i nelineárních kvantových počítačů, pak již bylo ukázáno, že s jejich pomocí lze řešit i NP-úplné problémy.

Kromě zaměření na konstrukci nových efektivních algoritmů existují v literatuře věnované teorii kvantového výpočtů další směry. Jsou to např.: návaznost na současnou teorii výpočetní složitosti, matematické úlohy související s vlastní konstrukcí kvantových počítačů (nezbytný počet a vlastnosti základních konstrukčních prvků - „gate“ - těchto počítačů, konstrukce vhodných opravných kódů k zabezpečení funkčnosti kvantových počítačů). Řada stěžejních otázek teprve na svou formulaci a následující řešení teprve čeká.

Na internetu existuje velká řada zdrojů poskytujících kvalitní informace k problematice kvantových počítačů. Většinu odborných publikací (včetně nejnovějších) lze nalézt v tomto archivu.

Základní otázkou je realizovatelnost kvantových počítačů. Za slibnou cestu je v současné době např. považováno využití nuklární magnetické rezonance (NMR). Jestli je však konstrukce výkonného kvantového počítače reálnou možností, či zda v tomto směru existují jiné pro nás nepřekonatelné překážky, ukáže teprve budoucnost.

Mezinárodní kryptografické normy a standarty

Vhodný výchozí přehled lze nalézt na následující adrese Information Security Standards .

Literatura

Bibliografie:

Lawrie Brown: (lze vyhledávat)
IACR Conference Proceedings, uspořádané dle jména autora
Sean Irvine
Journal of Cryptology – bibliografie a obsah (IACR).
Kevin McCurley: bibliografie z výpočetní teorie čísel
Itribe (Cryptography Technical Report Server)
Quantum Computing Bibliography from University of Montreal.
Ron Rivest's Crypto and Security bibliography (Bibtex).
STOC Bibligraphies
FOCS Bibligraphies
Doug Stinson: schémata pro sdílené tajemství

Unified Computer Science Technical Report Index

Některé vybrané citace:

G. Agnew, R. Mullin and S. Vanstone, "An implementation of elliptic curve cryptosystems over F_2^{155}," IEEE Journal on Selected Areas in Communications, **11** (1993), 804-813.

James Bamford: The Puzzle Palace (Houghton Mifflin, 1982).

E. Biham and A. Shamir. Differential Cryptanalysis of the Data Encryption Standard. Springer-Verlag, New York, 1993.

E. Biham and A. Shamir. Differential cryptanalysis of the full 16-round DES. In Advances in Cryptology --- Crypto '92, Springer-Verlag, New York, 1993.

M. Blum and S. Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In Advances in Cryptology --- Crypto '84, pages 289--299, Springer-Verlag, New York, 1985.

J. Brandt and I. Damgard. On generation of probable primes by incremental search. In Advances in Cryptology --- Crypto '92, Springer-Verlag, New York, 1993.

G. Brassard. Modern Cryptology. Volume 325 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1988.

D.M. Bressoud. Factorization and Primality Testing. Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1989.

E. Brickell, D. Gordon, K. McCurley and D. Wilson, "Fast Exponentiation with precomputation," *Advances in Cryptology - EUROCRYPT* '92 Lecture Notes in Computer Science, **658** (1993), Springer-Verlag, 200-207.

E.F. Brickell and A.M. Odlyzko. Cryptanalysis: A survey of recent results. Proceedings of the IEEE, 76:578--593, 1988.

J. Brillhart, D.H. Lehmer, J.L. Selfridge, B. Tuckerman, and S.S. Wagstaff Jr. Factorizations of b^n +/- 1, b=2,3,5,6,7,10,11,12 up to High Powers. Volume 22 of Contemporary Mathematics, American Mathematical Society, Providence, Rhode Island, 2nd edition, 1988.

J. Buchmann, J. Loho, and J. Zayer. An implementation of the general number field sieve. In Advances, in Cryptology --- Crypto Ô93, Springer-Verlag, New York, 1994. To appear.

M.V.D. Burmester, Y.G. Desmedt, and T. Beth. Efficient zero-knowledge identification schemes for smart cards. Computer Journal, 35:21--29, 1992.

K.W. Campbell and M.J. Wiener. Proof that DES is not a group. In Advances in Cryptology --- Crypto '92, Springer-Verlag, New York, 1993.

H.Cohen and A.K.Lenstra: Implementation of a new primalit test, Mathematics of Computation, 48 (1987), 103-121

H.Cohen and H.W.Lenstra, Jr.: Primality testing and Jacobi sums, Mathematics of Computation, 42 (1984), 297-330

D. Coppersmith, A.M. Odlyzko, and R. Schroeppel. Discrete logarithms in GF(p). Algorithmica, 1:1--15, 1986.

B. den Boer and A. Bosselaers. An attack on the last two rounds of MD4. In Advances in Cryptology --- Crypto '91, pages 194--203, Springer-Verlag, New York, 1992.

B. den Boer and A. Bosselaers. Collisions for the compression function of MD5. In Advances in Cryptology --- Eurocrypt '93, 1993. Preprint.

Dorothy E. Denning. The Clipper encryption system. American Scientist, 81(4):319--323, July-August 1993.

W. Diffie. The first ten years of public-key cryptography. Proceedings of the IEEE, 76:560--577, 1988.

W. Diffie and M.E. Hellman. Exhaustive cryptanalysis of the NBS Data Encryption Standard. Computer, 10:74--84, 1977.

W. Diffie and M.E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, IT-22:644--654, 1976.

T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, IT-31:469--472, 1985.

A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Advances in Cryptology --- Crypto '86, pages 186--194, Springer-Verlag, New York, 1987.

S. Goldwasser and J. Kilian, "Almost all primes can be quickly certified," *Proceedings of the 18th Annual ACM Symposium on Theory of Computing* (1986), 316-329

S. Goldwasser and S. Micali. Probabilistic encryption. J. of Computer and System Sciences, 28:270--299, 1984.

D. Gordon, "Discrete logarithms in *GF* (*p*) using the number field sieve," *SIAM Journal on Discrete Mathematics*, **6** (1993), 124-138

D.M. Gordon and K.S. McCurley. Massively parallel computation of discrete logarithms. In Advances in Cryptology --- Crypto '92, Springer-Verlag, New York, 1993.
M.E. Hellman. A cryptanalytic time-memory trade off. IEEE Transactions on Information Theory, IT-26:401--406, 1980.
D. Kahn. The Codebreakers. Macmillan Co., New York, 1967.
B.S. Kaliski. A survey of encryption standards. RSA Data Security, Inc., September 2, 1993.
B.S. Kaliski Jr., R.L. Rivest, and A.T. Sherman. Is the data encryption standard a group? J. of Cryptology, 1:3--36, 1988.
D.E. Knuth. The Art of Computer Programming. Volume 2, Addison-Wesley, Reading, Mass., 2nd edition, 1981.
N. Koblitz. A Course in Number Theory and Cryptography. Springer-Verlag, New York, 1987.
N. Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48:203--209, 1987.
X. Lai and J.L. Massey. A proposal for a new block encryption standard. In Advances in Cryptology --- Eurocrypt '90, pages 389--404, Springer-Verlag, Berlin, 1991.
B.A. LaMacchia and A.M. Odlyzko. Computation of discrete logarithms in prime fields. Designs, Codes and Cryptography, 1:47--62, 1991.
S. Landau. Zero knowledge and the Department of Defense. Notices of the American Mathematical Society, 35:5--12, 1988.
G. Lay and H. Zimmer, "Constructing elliptic curves with given group order over large finite fields," *Algorithmic Number Theory: First International Symposium, Lecture Notes in Computer Science*, **877** (1994), Springer-Verlag, 250-263.
A.K. Lenstra and H.W. Lenstra Jr. Algorithms in number theory. In J. van Leeuwen, editor, Handbook of Theoretical Computer Science, MIT Press/Elsevier, Amsterdam, 1990.
A.K. Lenstra and M.S. Manasse. Factoring with two large primes. In Advances in Cryptology --- Eurocrypt '90, pages 72--82, Springer-Verlag, Berlin, 1991.
H.W. Lenstra Jr. Factoring integers with elliptic curves. Ann. of Math., 126:649--673, 1987.
M. Matsui. Linear cryptanalysis method for DES cipher. In Advances in Cryptology --- Eurocrypt '93, Springer-Verlag, Berlin, 1993.
R. Lercier and F. Morain, "Counting the number of points on elliptic curves over finite fields: strategies and performances," *Advances in Cryptology - EUROCRYPT '95*, Lecture Notes in Computer Science, (1995), Springer-Verlag.
A.Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.
A.Menezes, van Oorschot and Vanstone: Handbook of Applied Cryptography (CRC Press, 1997).
A.Menezes, S. Vanstone, and R. Zuccherato, *Counting points on elliptic curves over F_{2^m}*, Mathematics of Computation, **60** (1993), 407-420
R.C. Merkle and M.E. Hellman. Hiding information and signatures in trapdoor knapsacks. IEEE Transactions on Information Theory, IT-24:525--530, 1978.
R.C. Merkle and M.E. Hellman. On the security of multiple encryption. Communications of the ACM, 24:465--467, July 1981.
V.S. Miller. Use of elliptic curves in cryptography. In Advances in Cryptology --- Crypto '85, pages 417--426, Springer-Verlag, New York, 1986.
A.M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In Advances in Cryptology --- Eurocrypt Ô84, pages 224--314, Springer- Verlag, Berlin, 1984.
A.Odlyzko, The Future of Integer Factorization, CryptoBytes **1**, 2 (Summer 1995), RSA Laboratories,
S.C.Pohlig, M.E.Hellman. An improved algorithm for computing logarithms over *GF(p)* and its cryptographic significance, *IEEE Trans. Information Theory*, Vol. 24, No. 1 Jan. 1978, *pp*. 106-110
J. Pollard. Monte Carlo method for factorization. BIT, 15:331--334, 1975.
J. Pollard. Theorems of factorization and primality testing. Proc. Cambridge Philos. Soc., 76:521--528, 1974.
M.O. Rabin. Digitalized signatures as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT, 1979.
R.L. Rivest. Cryptography. In J. van Leeuwen, editor, Handbook of Theoretical Computer Science, MIT Press/Elsevier, Amsterdam, 1990.
R.L. Rivest. Finding four million random primes. In Advances in Cryptology --- Crypto '90, pages 625--626, Springer-Verlag, New York, 1991.
R.L Rivest. The MD4 message digest algorithm. In Advances in Cryptology --- Crypto '90, pages 303--311, Springer-Verlag, New York, 1991.

R.L. Rivest. RFC 1321: The MD5 Message-Digest Algorithm. Internet Activities Board, April 1992.
R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120-- 126, February 1978.
R. Schoof, “Elliptic curves over finite fields and the computation of square roots mod *p*,” *Mathematics of Computation*, **44** (1985), 483-494
R. Schoof, "Counting points on elliptic curves over finite fields," *Journal de Théorie des Nombres* **7** (1995), 255-282
Schneier, B., *Applied Cryptography Second Edition: protocols, algorithms, and source code in C*, John Wiley & Sons, 1996.
C.P. Schnorr. Efficient identification and signatures for smart cards. In Advances in Cryptology --- Crypto '89, pages 239--251, Springer-Verlag, New York, 1990.
M. Shand and J. Vuillemin. Fast implementations of RSA cryptography. In Proceedings of the 11th IEEE Symposium on Computer Arithmetic, pages 252--259, IEEE Computer Society Press, Los Alamitos, CA, 1993.
C. E. Shannon, “The Mathematical Theory of Communication,” Bell System Technical Journal, 1948
R.D. Silverman. The multiple polynomial quadratic sieve. Math. Comp., 48:329--339, 1987.
M.E. Smid and D.K. Branstad. Response to comments on the NIST proposed Digital Signature Standard. In Advances in Cryptology --- Crypto '92, Springer- Verlag, New York, 1993.
M.J. Wiener. Efficient DES key search. August 20, 1993. Presented at Crypto '93 rump session.

Některé další zajímavé webovské stránky ke kryptografii a příbuzným problematikám

Každému zájemci o kryptologii doporučuji RSA-FAQ (Frequently Asked Questions)

Peter Gutmann's Security and Encryption-related Resources and Links:
Ritter's Net Links:
Lewis McCarthy´s Bookmarks:
Tatu Ylönen crypto links:
Richard Pinch´s links:
JoePeschel: Computer Security, Encryption, and Cryptanalysis:
Tom Dunigan's security pointers
Pat Farrell´s Crypto Sources:
Enryption and Privacy links: (archivy, USENET, IDEA, DES, PGP, steganografie, …)
Chris Vidler´s Cryptography Page:
Security and Cryptography:
Cryptography archive:
COAST: (Purdue University, Computer Sciences Department - řada článků a dalších adres)
John Savard's Home Page: (stručný úvod do kryptologie včetně historie)
Cryptographic Algorithms: (současné kryptografické algoritmy)
International Encryption Standards: (Mezinárodní normy)
Pointers to Cryptographic Software: (kryptografický software na webu)
Cryptographers' Home Pages: (kryptografové z celého světa, adresy)
crypto publications on line (odborné články na webu)
Pini Computer Trading's CRYPTO-CD: (CD obsahující informace z kryptologie na webu)
Encryption privacy and security resource page: (poslední informace z politiky týkající se kryptografie)
Cryptome: (kryptografie v tiskových zprávách)
Cryptography and Information Security Group (CIS Group): (MIT)
Ron Rivest's collection of links on Cryptography and Security!
Crypto Links (Counterpane): (Bruce Schneier)
Cryptography FAQ: (Usenet FAQ)
NSA (National Security Agency):
NSA (unofficial page):
SET (Mastercard site):
Pointers to Cryptographic Software:
CRYPTO•LOG, The Internet Guide to Cryptography: (různé adresy)
Řada krypto ftp serverů (v němčině)
Europe and Cryptography (politika, legislativa a technologie - Evropa)
IEEE Cipher Newsletter Archive: (řada nových informací v kryptografii)

Quadralay´s Cryptography Archive
International Association for Cryptologic Research (IACR, konference CRYPTO, Eurocrypt)
Computational Number Theory and Cryptography (LIX, Ecole Polytechnique)
Aegean Park Press
Cryptobytes a Bulletins (RSA Laboratories)
NIST (National Institute of Standarts and Technology)
Anglie-ftp archiv
Some interesting references on elliptic curves...
George Barwood´s FAQ on elliptic curves cryptography
CERTICOM - eliptické křivky
McCurley´s list of web pages of interest to cryptography researchers
Cryptography: The Study of Encryption: (Francis Litterio)
Cryptography and Security: (Ronald Rivest)
Cyphernomicon: (FAQ, PGP, Clipper, Key Escrow,…)
Publications of Eli Biham
Cryptography Online courses
Ciphers by Ritter
Security and Encryption-related Resources and Links (bohatý zdroj dalších adres)
CRYPTOGRAPHY AND LIBERTY AN INTERNATIONAL SURVEY OF ENCRYPTION POLICY
Crypto Law Survey: (přehled existujích a navrhovaných zákonů a opatření dotýkajících se kryptografie)
Journal of Cryptology
RFC Index
Internet RFC
Internet Engineering Task Force (IETF)
Nordic mirror for Internet drafts (by FTP)
Internet drafts (by FTP)
Bellare - Crypto links (některé další adresy, konference, organizace)
Security links
Security server: (v němčine)
IPSEC: (IP Security Protocol Working Group)
PKI: (Public Key Infrastructure - přehled)
PKCS: (RSADSI Public Key Cryptography Standards)

Zdroje informací ke kryptologii v českém jazyce:

Data Security Management (dvouměsíčník věnovaný problematice bezpečnosti dat orientovaný na manažery v dané oblasti)
Computerworld (Seriál o bezpečnosti a informačním soukromí redigovaný V.Matyášem)
CHIP (populární měsíčník věnovaný výpočetní technice otiskuje i články ke kryptografické problematice)

Autorova poznámka na závěr:

Materiál na této webovské stránce si klade za svůj cíl přiblížit zájemci (především českému) problematiku kryptologie a zpřístupnit mu alespoň část z velkého objemu informací v této oblasti, zejména informace dostupné na Internetu. Materiál bude dále průběžně doplňován. Autor proto bude vděčný za každý podnět, kritickou poznámku, námět k doplnění.

e-mail: Jaroslav.Pinkava@aec.cz