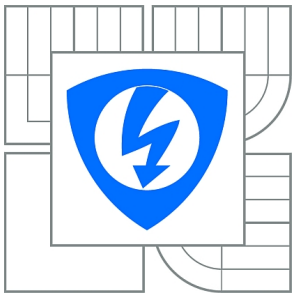
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

EFEKTIVNÍ SCHÉMATA DIGITÁLNÍCH PODPISŮ

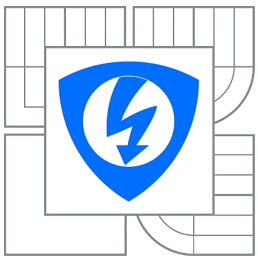
EFFICIENT DIGITAL SIGNATURE SCHEMES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. ONDREJ VARGA
AUTHOR

VEDOUCÍ PRÁCE — Ing. LUKÁŠ MALINA
SUPERVISOR

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Ondrej Varga	***ID:***	83162
Ročník:	2	***Akademický rok:***	2010/2011

NÁZEV TÉMATU:

Efektivní schémata digitálních podpisů

POKYNY PRO VYPRACOVÁNÍ:

Vyhledejte a nastudujte nové schémata digitálních podpisů. Zaměřte se na nové koncepty a návrhy krátkých el.podpisů, skupinových podpisů, kruhových podpisů atd. Proveďte srovnání a zhodnocení jejich efektivity a návrh nejvýhodnějšího řešení pro budoucí praktickou implementaci s ohledem na efektivnost, nízké paměťové nároky, bezpečnost, inovativnost a složitost implementace. V rámci diplomové práce vybraná řešení implementujte ve vybraném programovacím jazyce, porovnejte a analyzujte jejich vlastnosti.

DOPORUČENÁ LITERATURA:

[1] DOSTÁLEK, L. VOHNOUTOVÁ, M. Velký průvodce infrastrukturou PKI. Computer Press, Brno 2006, ISBN: 80-251-0828-7.

[2] MENEZES, Alfred, VAN OORSCHOT, Paul, VANSTONE, Scott. Handbook of applied cryptography. Boca Raton : CRC Press, 1997. 780 s. ISBN 0849385237.

[3] STALLINGS, William. Cryptography and Network Security. 4th edition. [s.l.] : [s.n.], 2006. 592 s. ISBN 0131873164.

Termín zadání:	7.2.2011	***Termín odevzdání:***	26.5.2011

Vedoucí práce: Ing. Lukáš Malina

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

ABSTRAKT

Digitálny podpis preberá vlastnosti klasického podpisu a slúži k zaisteniu obsahu dokumentov, ktoré by mohli byť v priebehu prenosu cez nezabezpečený kanál modifikované. Problémy poskytnutia bezpečnosti a ochrany komunikujúcich účastníkov sú riešené pomocou kryptografických technik. Overenie identity, integrity správ, dôveryhodnosť vlastníctva dokumentov a bezpečný prenos informácií cez nezabezpečený kanál, to všetko sú aspekty, ktorými sa zaoberá bezpečnosť komunikácie - infraštruktúra verejných kľúčov, ktorá používa digitálne podpisy. V dnešnej dobe digitálne podpisy majú veľký význam v zabezpečení dát v komunikácii cez nezabezpečený kanál.

Cieľom nasledujúcej diplomovej práce je oboznámiť čitateľa s nutnými technologickými aspektmi digitálnych podpisov, ich výhodami a nevýhodami. Vzhľadom na stále zväčšujúceho výpočtového výkonu počítačov súčasné digitálne podpisy sa časom budú musieť vylepšovať a modifikovať tak, aby obstáli sofistikovanejším útokom. V tejto práci sú popísané aj návrhy nových efektívnych podpisových schém a ich porovnanie so súčasnými. Sú preskúmané aj ich aspekty pre výpočtovo slabé zariadenia, či nasadenie digitálneho podpisu v systémoch s nízkymi prenosovými rýchlosťami kanálu.

Na základe objasnenia problematiky kryptografii a popísania jeho základných pojmov je následne zavedený pojem digitálneho podpisu. V kapitole 1 sú popísané aj formátovania a architektúry digitálneho podpisu. Druhá časť tejto diplomovej práce sa venuje súčasne používaným digitálnych podpisovým schémam a ich vlastnostiam. Kapitola 3 popisuje niektoré návrhy nových efektívnych digitálnych podpisových schém a ich porovnanie so súčasne používanými. V praktickej časti práce, ktorá je popísaná v kapitole 4 je predstavená implementácia (v prostredí .NET v jazyku C#) dvoch efektívnych podpisových schém v rámci aplikácii typu klient-server. Posledná časť sa zaoberá s porovnaním a analyzovaním vlastností implementovaných schém.

Kľúčové slová: asymetrická kryptografia, digitálny podpis, podpisový kľúč, efektívna digitálna podpisová schéma, kryptografia na báze eliptických kriviek, bilineárne párovanie

ABSTRACT

Digital signatures, which take the properties of classical signatures, are used to secure the actual content of documents, which can be modified during transmission over an insecure channel. The problems of security and protection of communicating participants are solved by cryptographic techniques. Identity verification, message integrity, credibility, the ownership of documents, and the secure transmission of information over an unsecured channel, are all dealt with in secure communications - Public Key Infrastructure, which uses digital signatures. Nowadays digital signatures are often used to secure data in communication over an unsecured channel.

The aim of the following master's thesis is to familiarize readers with the necessary technological aspects of digital signatures, as well as their advantages and disadvantages. By the time digital signatures are being used they will have to be improved and modified to be secure against more sophisticated attacks. In this paper, proposals of new efficient digital signature schemes and their comparison with current ones are described. Also are examined their implications for computationally weak devices, or deployment in low speed channel transmission systems.

After an explanation of cryptography and a description of its basic subjects, digital signatures are introduced. The first chapter describes the possible formatting and architecture of the digital signature. The second part of this master's thesis is about current digital signature schemes and their properties. Chapter 3 describes some proposals of new efficient digital signature schemes and their comparison to those currently in use. In the practical part, the implementations (in the environment .NET in C#) of two effective digital signature schemes as part of a client-server application are presented and described (Chapter 4). In the last chapter the comparison and analysis of the implemented signature schemes are provided.

Key words: public key cryptography, digital signature, signature key, effective digital signature scheme, elliptic curve cryptography, bilinear pairing

VARGA, O. *Efektivní schémata digitálních podpisů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 70 s. Vedoucí diplomové práce Ing. Lukáš Malina.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Efektivní schémata digitálních podpisů jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedeného diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

podpis autora

Poděkování

Děkuji vedoucímu práce Ing. Lukášovi Malinovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne …….. .....

podpis autora

OBSAH

ÚVOD

V minulosti bolo vypracované ľudstvom mnoho metód pre overovanie obsahu a podpisu dokumentov, ktoré sa používali pre dokazovanie vlastníctva, alebo pre komunikáciu fyzicky vzdialených osôb. Problémom je, že aj základné záznamové médium ľudstva (papier) je ľahko sfalšovateľné a že podpis je možné ľahko napodobniť, dokonca zmeniť obsah už podpísaného dokumentu. Postupom času a hlavne stále zväčšujúcou popularitou Internetu, ako globálneho komunikačného prostriedku, sa začali vyskytovať tieto problémy aj v elektronickej komunikácií.

Problémy spojené autentizáciou, overovaním komunikujúcich strán a zaistením integrity správ sú zamerané hlavne na overovanie digitálneho podpisu komunikujúcich strán a na podmienku nepopierateľnosti, kedy je prenášaná správa vytvorená práve jedným z komunikujúcich osôb. V dnešnej dobe práve autentizácie komunikujúcich strán sú veľmi diskutovanou témou a s tým súvisiaci digitálny podpis, ktorý vo svojej podstate zahrňuje najmodernejšie kryptografické metódy, kombinujúce symetrické a asymetrické šifrovacie a kódovacie algoritmy.

Cieľom diplomovej práce Efektivní schémata digitálních podpisů je popísať, a následne porovnať súčasné digitálne podpisové schémy s návrhmi nových efektívnych schém. Ďalej preskúmať ich aspekty pre výpočtovo slabé zariadenia, či nasadenie digitálneho podpisu v systémoch s nízkymi prenosovými rýchlosťami kanálu. V praktickej časti previesť implementáciu niektorých efektívnych schém vo vybranom programovacom jazyku, a pomocou aplikácii porovnať a analyzovať ich vlastnosti.

Na začiatku sa bude venovaná pozornosť kryptológií ako nutnému základu pre pochopenie celej tematiky. Po vysvetlení základných pojmov tejto vedy ešte v prvej kapitole sa zavedie pojem digitálneho podpisu a budú popísané jeho možnosti formátovania a architektúry. V ďalšej kapitole sa popíšu súčasné digitálne podpisové schémy a vlastnosti jednotlivých schém. V najzákladnejších schémach bude uvedený aj samotný algoritmus generovania a overovania podpisov. Tretia kapitola sa venuje teoretickým návrhom niektorých nových efektívnych podpisových schém. Tu sa zhrnú aj vlastnosti nových a súčasných schém zamerané na efektívnosť a dĺžky vytvoreného podpisu, ale aj ich možnosti nasadenia podľa typov systémov.

Praktická časť tejto práce je popis praktickej implementácii v prostredí .NET v jazyku C#, kde budú popísané funkcie implementovanej aplikácie a rozbor vlastnosti použitých podpisových schém.

V záveru potom bude zhrnuté použitie a zrovnanie analyzovaných podpisových schém a ich zhodnotenie s výsledkami z praktickej časti.

1 KRYPTOGRAFIA

Kryptografia alebo šifrovanie je náuka o metódach utajovania zmyslu správ prevodom do podoby, ktorá je čitateľná len so špeciálnou vedomosťou. Slovo kryptografia pochádza z gréčtiny – kryptós je skrytý a gráphein znamená písať. Moderná kryptografia križuje vedy ako matematika, počítačová technika a inžinierstvo. Aplikáciám tejto vedy patria napr. kreditné karty, zabezpečené VoIP prenosy, dátové prenosy, archivácia tajných informácii, počítačové heslá a elektronické obchody.

Po druhej svetovej vojne s príchodom počítača, v kryptografii používané metódy začínali byť oveľa komplexnejšie a ich použitie oveľa rozsiahlejšie čo mnohokrát ovplyvnilo beh dejín. Súčasne s vývojom technológii súvisiacich s kryptológiou sa v praxi vyskytli aj také problémy, ktoré zostali dodnes nevyriešené.

1.1 Úvod do kryptografie

Ochrana kryptografických systémov je väčšinou založená na základe nejakého matematického problému, ale existujú aj iné spôsoby utajenia. Napríklad substitučná metóda funguje na princípe, že každý prvok „abecedy“ otvoreného textu je nahradeným iným prvkom „abecedy“, a tým je správa prevedená do nečitateľnej podoby pre nepovolané osoby. V použitých operáciách je snaha o maximalizácii difúzie (zmena otvoreného textu sa premieta do mnohé miest zašifrovaného textu) a konfúzie (nie je možné predikovať, akú zmenu zašifrovaného textu vyvolá len malá zmena otvoreného textu). Použité matematické problémy sú zvolené tak, aby ich útočník nebol schopný vyriešiť v reálnom čase. Užívateľ vlastní nejakú tajnú informáciu, pomocou ktorej je schopný daný problém vyriešiť a tým pádom získať prístup k aktívam (všetko čo je majiteľom považované za cenné). Kryptografická ochrana je teda technická ochrana, ktorá je založená na obťažnosti riešenia matematických problémov.

Zaistenie dôvernosti je dosiahnuté utajením informácii prenášaných v kanáli, ku ktorým má prístup aj útočník (obr. 1.1). Informácia je zakódovaná všeobecne známym kódom do podoby správy Z (tzv. otvorený text). Množina symbolov, ktoré sa môžu pri kódovaniu použiť sa nazýva abeceda správy a množina všetkých možných správ sa nazýva množina správ Z. Pre šifrovanie sa používa šifrovací kľúč K_E a pre dešifrovanie dešifrovací kľúč K_D. Tieto kľúče podľa použitého šifrovacieho algoritmu môžu byť rovnaké (tj. $K_E = K_D$) alebo

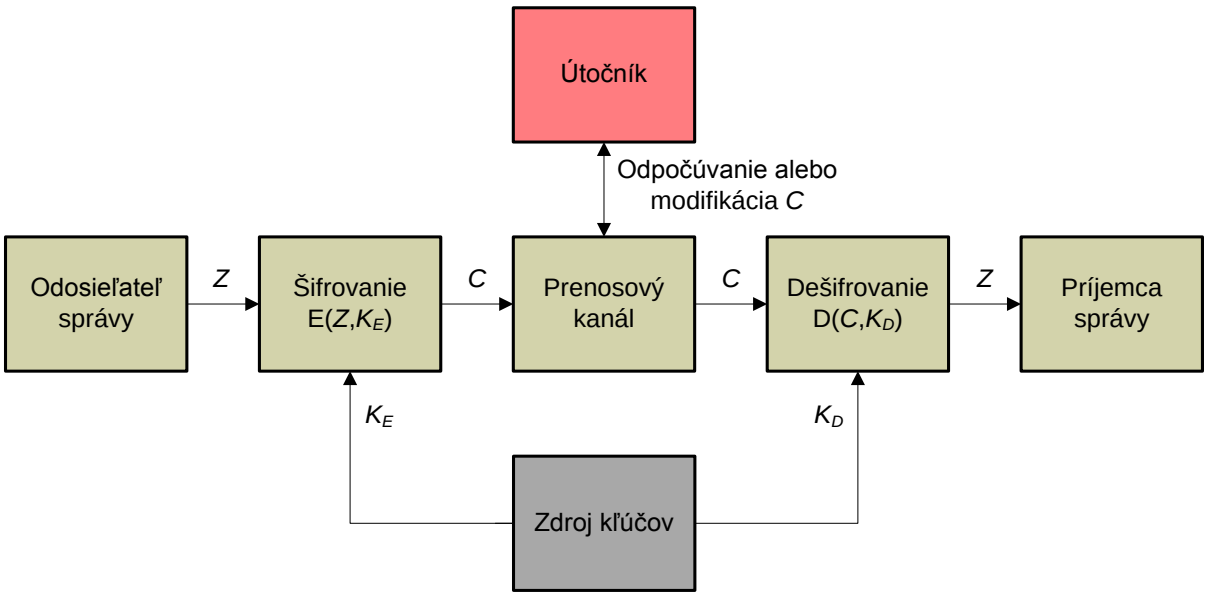
môžu byť odlišné (tj. $K_E \neq K_D$). Zašifrovaný text (tzv. kryptogram) je všeobecne neznámym spôsobom zakódovaná informácia vo forme postupnosti symbolov z abecedy kryptogramu a značíme symbolom *C*. Množina všetkých možných kryptogramov sa nazýva množina kryptogramov *C*. Proces transformácie otvoreného textu na zašifrovaný text sa nazýva šifrovanie a opačný proces je dešifrovanie. Pri šifrovaní dochádza k postupnému zobrazeniu z množiny *Z* do množiny *C* podľa parametra K_E:

$$C = \mathrm{E}\ (Z, K_E).$$

Dešifrovanie je potom inverzné zobrazenie:

$$Z = \mathrm{D}\ (C, K_D).$$

Na ľavej strane kryptografického systému odosielateľ vysiela správu *Z*, ktorá je potom transformovaná pomocou šifrovacieho kľúča K_E do kryptogramu *C*. Tento kryptogram je prenášaný cez prenosový kanál, dešifrovaný pomocou dešifrovacieho kľúča K_D a predaný k príjemcovi správy. Behom prenášania kryptogramu cez prenosový kanál má možnosť útočník zašifrovanú správu zachytiť, ktorej sa potom snaží získať obsah alebo sa snaží ju modifikovať.



Obr. 1.1: Kryptografický systém.

Na základe závislosti dešifrovacieho kľúča so šifrovacím kľúčom môžeme kryptografické systémy deliť do dvoch základných tried:

1. kryptografické systémy s tajným kľúčom (tzv. symetrické kryptosystémy)
2. kryptografický systémy s verejným kľúčom (tzv. asymetrické kryptosystémy)

V systémoch s tajným kľúčom šifrovací a dešifrovací kľúč musí zostať v tajnosti, lebo dešifrovací kľúč je buď rovnaký ako šifrovací (tj. $K_D = K_E$) alebo je zo šifrovacieho kľúča relatívne jednoducho odvoditeľný, teda funkcia $K_D = f(K_E)$ je výpočtovo prakticky realizovateľná. Systémy s tajným kľúčom sú veľmi rýchle a slúžia k šifrovaniu väčšiemu objemu dát. Ich veľkým problémom je bezpečná distribúcia kľúčov od zdroja k odosielateľovi a príjemcovi.

U systémoch s verejným kľúčom platí, že oba kľúče sú rôzne a nie sú odvoditeľné jeden z druhého, teda funkcia $K_D = f(K_E)$, respektíve $K_E = f^{-1}(K_D)$ je výpočtovo prakticky nerealizovateľná. V tomto prípade je jeden z kľúčov tajný a druhý je verejne známy. Skutočnosť, či šifrovací alebo dešifrovací kľúč je tajný, určuje typ kryptografického systému. Keď je verejným kľúčom K_E a tajným kľúčom K_D, tak kryptogram môže zašifrovať verejným kľúčom hocikto, ale dešifrovať ho môže len majiteľ tajného kľúča. V opačnom prípade je zaručená autentičnosť prenesenej správy (tzv. digitálny podpis). Výhodou systémov s verejným kľúčom je jednoduchšia distribúcia kľúčov, lebo tvorca kryptografického systému ponechá pre seba svoj tajný kľúč, ktorý používal na zašifrovanie a druhý kľúč jednoducho zverejní. Nevýhodou týchto systémov je pomalé šifrovanie a dešifrovanie, a preto sa v praxi asymetrické kryptosystémy používajú k podpisovaniu správ a k šifrovaniu pri prenose tajných kľúčov pre symetrické kryptosystémy. Tieto kľúče sú veľmi krátke (napr. 256 bitov) a ich pomalé šifrovanie, respektíve dešifrovanie nie je kritickým problémom.

1.2 Symetrické kryptosystémy

Tieto systémy sú veľmi rýchle, zaručujú dôvernosť a autentičnosť prenášaných informácii na vysokej úrovni, a preto majú v informačných systémoch veľmi široké využitie. Symetrické kryptosystémy sa delia na prúdové a blokové šifry. V prípade prúdových šifier hodnota zašifrovaného bitu závisí na hodnote príslušného bitu správy a na hodnote kľúča. V prípade blokovej šifry zašifrovaného bitu závisí aj na hodnote ďalších bitov danej správy. Blokové šifry sú preto obecne bezpečnejšie a tým pádom perspektívnejšie ohľadom bezpečnosti. Na druhej strane sú prúdové šifrátory rýchlejšie. Viac o symetrických kryptosystémoch je popísané v [2].

1.3 Asymetrické kryptosystémy

Ich charakteristickou vlastnosťou je existencia verejného kľúča *e* a súkromného kľúča *d*. Kým súkromným kľúčom disponuje len tvorca kryptosystému, verejný kľúč pozná každý. Pri návrhu takéhoto systému sa uplatňuje tzv. jednocestná funkcia. Jednocestná funkcia $y = f(x)$ je taká funkcia, pre ktorú je výpočet hodnoty y z argumentu x relatívne jednoduchý ale opačným smerom prakticky nemožný.

Podľa typu použitej jednocestnej funkcie sa v praxi najviac používajú kryptosystémy založené na probléme faktorizácii čísla (FP = „Factorization Problem“), na probléme diskrétneho logaritmu (DLP = „Discrete Logaritmic Problem“) a na eliptických krivkách (ECDLP = „Elliptic Curve Discrete Logarithmic Problem“).

Kryptosystém RSA

Najznámejším a najpoužívanejším kryptosystémom faktorizačného systému je RSA. Je založený na probléme faktorizácii čísla, čo je problém rozkladu daného čísla na súčin mocnín prvočísel.

Konštrukcia kryptosystému:

1. Vybrané sú dve veľké prvočísla p a q.
2. Vypočítajú sa čísla: $n = (p \,.\, q)$, $r = (p - 1).\,(q - 1)$.
3. Zvolí sa verejný šifrovací kľúč e. Toto číslo musí byť nesúdeliteľné s číslom r.
4. Vypočíta sa tajný šifrovací kľúč $d = e^{-1} \bmod r$.
5. Parametre e a n sú zverejnené. Ostatné parametre musia zostať tajné.

Postup šifrovania:

1. Správa Z je rozdelená na bloky symbolov o rovnakej dĺžke. Každý i-tý blok správy sa chápe ako číslo z_i. Musí platiť, že $z_i < n$.
2. Každý blok správy z_i sa zašifruje: $c_i = z_i^e \bmod n$.
3. Z blokov c_i je poskladaný kryptogram C a potom je odoslaný k adresátovi.

Postup dešifrovania:

1. Kryptogram C je rozdelený na pôvodné bloky c_i.
2. Každý blok kryptogramu sa dešifruje: $z_i = c_i^{\,d} \bmod n$.

3. Z blokov z_i je poskladaná správa Z.

Odolnosť RSA voči kryptoanalýze spočíva v tom, že k získaniu súkromného kľúča útočník potrebuje vedieť číslo $r = (p - 1).(q - 1)$. Toto číslo je možné určiť len faktorizáciou verejného parametra n. V súčasnej praxi sa používajú kryptosystémy, kde číslo n je 1024 alebo 2048 bitov dlhé. Faktorizácia tak veľkých čísel je v súčasnosti prakticky nereálna. Viac o tohto kryptosystéme viď. [18].

Logaritmické verejné kryptosystémy

Logaritmické verejné kryptosystémy (DL systémy podľa „Discrete Logarithm“) sú založené na problému nájdenia diskrétneho logaritmu. V systéme je použitá funkcia $y = f(x) = g^x \bmod p$, kde veľké prvočíslo p a základ mocniny g sú známe. Výpočet hodnoty y pre argument x je relatívne jednoduché, ale inverzný výpočet, tj. nájdenie hodnoty x pre dané y je výpočtovo prakticky nemožný.

K bezpečnému vytvoreniu kľúčov pre symetrický kryptosystém prostredníctvom prenosového kanálu je najvýznamnejšou metódou Diffie-Hellmanov protokol. Aj keď útočník zachytí akúkoľvek komunikáciu medzi užívateľmi v kanáli, pomocou verejných parametrov veľkého prvočísla p a primitívneho koreňa g nebude schopný zistiť aký kľúč bol ustanovený. Druhá možnosť útoku je útok stredom, kedy si útočník pri zostavovaní kľúča dohodne uprostred dva rôzne kľúče pre komunikáciu s obidvomi stranami, a tým ďalej môže odpočúvať ich komunikáciu. Na vyriešenie tohto problému sa začali používať infraštruktúry verejných kľúčov (PKI = „Public Key Infrastructure“) a elektronické podpisy.

Pri označení odosielateľa správy symbolom A a príjemcu B, postup ustanovenia kľúča K je nasledujúci:

1. Odosielateľ zvolí náhodne veľké číslo a a príjemca zvolí náhodné veľké číslo b. Tieto čísla sú tajné.
2. Účastník A vypočíta číslo $X = g^a \bmod p$. Podobne účastník B vypočíta číslo $Y = g^b \bmod p$. Vypočítané čísla si účastníci prenosovým kanálom navzájom predajú.
3. Účastník A vypočíta $K = Y^a \bmod p$. Účastník B vypočíta tú istú hodnotu $K = X^b \bmod p$. Hodnota K je kľúčom pre symetrický kryptosystém.

Skutočnosť, že obidvaja účastníci vypočítajú rovnaký kľúč plynie z tejto rovnice:

$$K = Y^a \bmod p = (g^b)^a \bmod p = (g^a)^b \bmod p = X^b \bmod p.$$

Odolnosť Diffie-Hellmanovho protokolu voči kryptoanalýze spočíva v zložitosti riešenia diskrétneho logaritmu. Známe sú verejné parametre p, g a útočník môže zachytiť vymenené hodnoty $g^a \bmod p$ a $g^b \bmod p$. K zisteniu kľúča K treba zistiť buď tajné číslo a alebo b, čomu je možné využiť rovnosť $X = g^a \bmod p$ a $Y = g^b \bmod p$. Táto matematická úloha nie je v súčasnej dobe prakticky reálna pre v praxi používané hodnoty p z oboru hodnôt 2^{768} až $2^{1536} \approx 10^{231}$ až 10^{462}. Bezpečnosť tohto kryptosystému z pohľadu odolnosti je zrovnateľný s kryptosystémom RSA. Viac o tomto protokole je popísané v [18].

1.4 Úvod do elektronických podpisov

U asymetrických kryptosystémoch je zaistenie autentičnosti (pôvodnosti) prijatých dát veľmi dôležité. Jednoduchým spôsobom autentizácii dát je ich zašifrovanie súkromným kľúčom osoby, ktorá sa za pôvodnosť dát zaručuje. Potom už hocikto môže dešifrovať dáta verejným kľúčom podpisujúceho a keď mu dôveruje, bude veriť aj v autentičnosti príslušných dát.

U vyššie popísanom spôsobe sa vyskytnú dve veľké problémy. Prvým je, že autentizačné dáta sú často veľmi objemné a ich zašifrovanie a dešifrovanie asymetrickým kryptosystémom môže byť pomalé. Druhým problémom je overenie pôvodnosti verejného dešifrovacieho kľúča. Prvý problém sa rieši pomocou tzv. hašov („hash“) a druhý pomocou infraštruktúry verejných kľúčov (PKI).

Hašovacie funkcie

Problém pomalého šifrovania a dešifrovania sa rieši tak, že pre celú správu Z sa vypočíta haš h o dĺžke obvykle 128, 160, 192, 256 alebo 512 bitov. Pritom sa požaduje, že prakticky nie je možné nájsť dve rozdielne správy s rovnakým hašom a pre nejakú hodnotu haše musí byť prakticky nemožné nájsť dve rozdielne správy.

Namiesto celej správy sa šifruje len jej haš. Potom sa vyskytne problematika, či doručené dáta boli modifikované behom prenosu. Toto sa rieši pripojením autentizačného kódu správy ku každej správe, ktorý príjemcovi slúži ku kontrole. Autentizačné kódy správ sa tvoria buď hašovaciou (napr. MD5, SHA-1) alebo autentizačnou funkciou (napr. MAC = „Message Authentication Code“).

Digitálny podpis

Digitálny podpis príjemcovi správy umožňuje overiť si, že správu odoslal udávaný odosielateľ a správa nebola pozmenená. Vychádza z princípu klasického ručného podpisu a je aplikovaný pomocou verejného kryptosystému. Odosielateľ správy (tj. podpisujúca osoba) disponuje súkromným kľúčom, ktorý je jednocestnou funkciou viazaný s verejným kľúčom. Hocikto si môže na dôsledku jednocestnej funkcii overiť, či podpisujúca osoba vlastní súkromný kľúč, ale nikto nemôže ho zistiť pomocou verejného kľúča. To, že elektronicky doručený verejný kľúč patrí udávanej osobe a nie nejakému útočníkovi sa rieši pomocou certifikátov v rámci infraštruktúry verejných kľúčov (PKI), čo rovno poskytuje i nepopierateľnosť.

Certifikát

Certifikát je vlastne určité tvrdenie (napr. určitá osoba má určitý verejný kľúč) elektronicky podpísané nejakou dôveryhodnou stranou. Dôveryhodnou stranou môže byť určitá inštitúcia, ktorá má všeobecnú dôveru. Táto inštitúcia sa nazýva certifikačná autorita (CA) a jeho verejný kľúč V_{CA} je bezpečne doručený všetkým užívateľom. Pomocou tohto kľúča potom užívatelia môžu overovať všetky certifikáty CA a tým sprostredkovane aj všetky tvrdenia osôb, ktoré majú certifikát na svoje verejné kľúče.

Infraštruktúra verejných kľúčov (PKI)

K praktickej realizácii k vyššie popísanom princípe sa buduje infraštruktúra verejných kľúčov (PKI). Jeho hlavným prvkom je certifikačná autorita (CA). Jej môžu byť podriadené ďalšie CA a tzv. registračné autority (RA). Úlohou RA je fyzické overovanie údajov žiadateľa o certifikát. Súčasťou PKI je ešte zoznam zneplatnených certifikátov (CRL = „Certificate Revocation List“). Tento zoznam udržuje CA a zapisuje tam certifikáty, ktoré boli zneplatnené ešte pred riadnym vypršaním (napr. z dôvodu kompromitácii tajného kľúča užívateľa). Užívatelia si v tomto zozname môžu overiť certifikáty s ktorými sa preukazuje druhá strana, s ktorou chcú komunikovať.

1.5 Architektúra verifikovaných digitálnych podpisových schém

Verifikované digitálne podpisové schémy využívajú pri komunikácii tretiu dôveryhodnú stranu pre overenie integrity správy a autentizáciu podpísaného. Podľa použitej topológie systému certifikačných autorít pre vydávanie certifikátov môžeme ich rozdeliť na centralizované a decentralizované. Reprezentantom centralizovaného systému je štandard X.509, ktorý používa hierarchický systém. Naopak PGP („Pretty Good Privacy“) používa decentralizovaný model potvrdzovania verejných kľúčov.

X.509

V kryptografii je X.509 štandard pre systémy založený na verejnom kľúči (PKI) pre jednoduché podpisovanie. Špecifikuje medzi iným formát certifikátov, zoznamy odvolaných certifikátov (CRL), parametre certifikátov a metódy kontroly platnosti certifikátov. Prvý štandard X.509 (verzia 1) bol vydaný v roku 1988, následne boli vydané 2. a 3. verzia, ktoré mimo iných rozširujú formát certifikátu o ďalšie voliteľné položky (unikátne identifikátory vydavateľa a vlastníka).

X.509 pôvodne predpokladal prísny hierarchický systém certifikačných autorít (CA) pre vydávanie certifikátov. Verzia 3 už obsahuje podporu ďalších topológii certifikačných autorít, ale používanie alternatívnej topológie (inej ako hierarchickej) nie je časté.

Jednotlivé certifikáty vydávajú certifikačné autority, ktoré potvrdzujú, že daný kľúč patrí uvedenému vlastníkovi. Organizácie môžu vydať svoje vlastné dôveryhodné koreňové certifikáty, ktoré platia v rámci podniku. Rada webových prehliadačov má predinštalované koreňové certifikáty známych poskytovateľov certifikátov (SSL certifikáty). Vo výsledku vlastníci prehliadača určujú, ktoré certifikačné autority sú dôveryhodné pre väčšinu populácii.

V štandardu X.509 je použitý nasledujúci formát certifikátu:

- Verzia: upresňuje formát certifikátu.
- Sériové číslo: jednoznačne identifikujúce číslo pridelené od CA.
- Identifikátor algoritmu: algoritmus, spolu s ďalšími údajmi použitými pri vytváraní podpisu certifikátu.
- CA: identifikátor CA, ktorá vytvorila a podpísala certifikát.
- Doba platnosti: od kedy do kedy je certifikát platný.

- Subjekt: užívateľ, ktorému patrí certifikovaný verejný kľúč.
- Verejný kľúč: podpísaný verejný kľúč, spolu s identifikátormi algoritmov.
- Podpis CA: je funkciou všetkých položiek certifikátov.

Certifikáty môžu byť fyzicky uložené do viacerých formátov s odlišnými príponami. Najčastejšie sa používajú prípony PEM („Privacy Enhanced Mail“) a DER („Distinguished Encoding Rules“). DER používa binárny formát pre kľúče i certifikáty. Jeho prekódovaním do Base64 a obalením hlavičkami vznikne formát PEM, ktorý má ASCII formát a má pôvod v štandarde pre zabezpečenie elektronickej pošty.

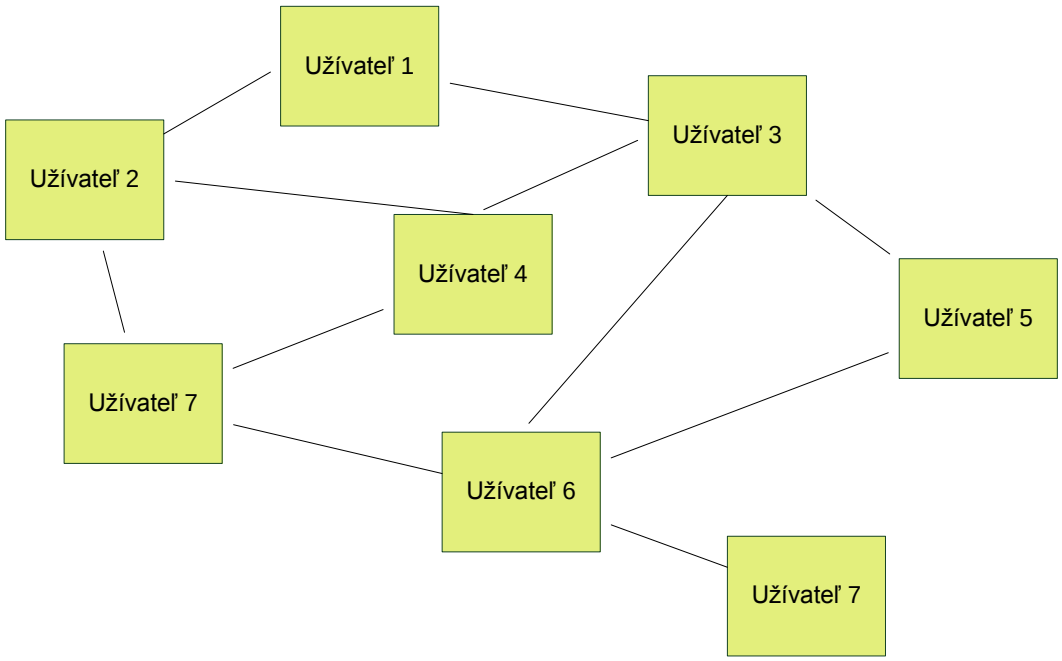
X.509 zavedie špecializované, obecne dôveryhodné CA, ktoré majú na starosti vydávanie a správu certifikátov. Práve preto potrebuje hierarchicky usporiadanú štruktúru, s pevne definovanými vzťahmi, ktoré sa menia len zriedka. V X.509 je len jeden identifikátor užívateľa oproti PGP, kde môže byť aj viacej. Certifikáty a mechanizmy v tomto štandarde je možné použiť pre delegovanie práv. Užívateľ môže vytvoriť certifikát, v ktorom predá časť právomoci svojmu zástupcovi. Po štandardnej autentizácii zástupca predloží tento certifikát a služba mu bude sprístupnená.

PGP

Je to počítačový program, ktorý sa hlavne používa pre autentizácii a šifrovanie e-mailovej komunikácie. Okrem toho umožňuje aj šifrovať súbory, vytvárať a spravovať tajné a verejné kľúče, certifikovať kľúče a digitálne podpísať správy. Prvá verzia bola uvedená Philom Zimmermanom v roku 1991.

PGP malo taký vplyv, že bolo štandardizované, aby bola umožnená spolupráca medzi rôznymi verziami PGP a podobného softvéru. PGP bolo prijaté ako internetový štandard pod názvom OpenPGP. Veľa e-mailových klientov podporuje OpenPGP a umožňuje tak pracovať s digitálnymi podpismi a šifrovanými správami.

Má decentralizovaný model potvrdzovania verejných kľúčov, v ktorom nie je žiadna obecne dôveryhodná CA. Užívateľ PGP môže vystupovať ako CA a potvrdzovať kľúče iných užívateľov, tým pádom každý užívateľ si môže zvoliť, ktorým užívateľom dôveruje a s akou mierou dôvery. Vzniká tak sieť dôvery - the web of trust, príklad viď. obr. 2.1.



Obr. 2.1: Príklad sieti dôvery.

V PGP šírenie kľúčov je riešené formou PGP certifikátov. Formát PGP certifikátu je nasledujúci:

- Verzia PGP.
- Certifikovaný kľúč: verejný kľúč.
- Informácia o majiteľovi kľúča: identifikačné údaje (meno, e-mail, fotografia).
- Podpis vlastníka kľúča: podpis vytvorený súkromným kľúčom odpovedajúcim certifikovanému kľúču.
- Preferované symetrické algoritmy.
- Doba platnosti.
- Podpis: možno pripojiť aj viac podpisov, každý identifikačný údaj sa podpisuje zvlášť.

Užívateľ spravuje zväzok verejných kľúčov, obsahujúce kľúče užívateľov, s ktorými komunikuje. Kľúče sa do zväzku pridávajú a uberajú dynamicky podľa potreby. Vo zväzku kľúčov sú ku každému verejnému kľúču pripojené tri položky, pomocou ktorých sa určuje platnosť kľúčov:

- Dôvera v majiteľovi (Owner Trust, OT): Možné hodnoty sú nedôveryhodné, čiastočne dôveryhodné, dôveryhodné a neznáme.

- Dôvera v podpise (Signature Trust, ST): Táto položka je priradená zvlášť ku každému podpisu na certifikáte. Hodnota položky je kópiou hodnoty OT z verejného kľúča autora podpisu, ktorý je uložený vo zväzku.
- Platnosť kľúča (Key Legitimacy, KL): Miera, s akou PGP verí, že tento kľúč skutočne patrí uvedenému užívateľovi v certifikáte.

V PGP dôvera je založená na samotných užívateľov a na ich rozhodnutiach, komu budú dôverovať, ktoré kľúče podpíšu apod. Tento model nie je vhodný pre masové nasadenie v geografických alebo politicky vzdialených skupinách užívateľov. Dvaja užívatelia, ktorý sa nikdy nestretli a nemajú spoločné prostriedky, majú veľmi ťažkú úlohu pri výmene verejných kľúčov. PGP umožňuje použitie viac identifikátorov a podpisov, pričom podpis sa môže sťahovať len k jednému identifikátorovi. Oproti X.509 nie je vhodný na delegovanie práv, lebo dôvera v platnosti kľúča automaticky neznamená dôveru v podpise vytvoreného týmto kľúčom.

1.6 Formátovanie podpisov

Výstupy digitálnych podpisových schém sú prevedené do praxe použiteľnej formy pomocou digitálnych podpisových procesov, ktoré sú zadefinované vo formátovaniach podpisov. V súčasnosti sa používa formátovanie ISO/IEC 9796 a PKCS („Public Key Cryptography Standards“).

ISO/IEC 9796 formátovanie

Toto formátovanie bolo zverejnené v roku 1991 ako prvý medzinárodný štandard pre digitálne podpisy. Jeho charakteristickými vlastnosťami sú, že je založené na kryptosystéme s verejným kľúčom, ale podpisujúci algoritmus priamo nie je špecifikovaný v štandarde. Použitý algoritmus musí byť deterministický. Tento štandard nepoužíva hašovaciu funkciu, zabezpečuje obnovu správy a je používaný pre podpísanie správy s limitovanou dĺžkou. Princíp tohto štandardu je podrobne popísaný v [11].

PKCS formátovanie

Je to sada špecifikácii, ktorá zahŕňa techniky pre šifrovanie a podpisovanie pomocou RSA. Vzťahuje sa na skupinu štandardov pre kryptografiu s verejným kľúčom navrhovanou a publikovanou spoločnosťou RSA Security.

PKCS #1 definuje matematické vlastnosti a formát RSA verejných a súkromných kľúčov, a základné algoritmy pre šifrovanie, dešifrovanie, digitálne podpisy a ich overovanie pomocou RSA. Digitálny podpisový mechanizmus v štandarde PKCS #1 nepoužíva funkciu obnovy správy zo schémy RSA. Používa hašovaciu funkciu (MD2 alebo MD5), a preto tento štandard sa používa pre digitálne podpisové schémy s dodatkom.

Štandard PKCS #3 popisuje Diffie-Hellmanov protokol pre výmenu tajných kľúčov dvoch užívateľov cez verejný komunikačný kanál. Štandard PKCS #7 je používaný pre podpisovanie a šifrovanie správ pomocou PKI a certifikátov. Z pohľadu digitálnych podpisov je veľmi perspektívna PKCS #13, ktorá ešte je vo vývojovej fáze ale bude popisovať digitálny podpis pomocou eliptických kriviek [16].

2 SÚČASNÉ DIGITÁLNE PODPISOVÉ SCHÉMY

Digitálny podpis je dátová reťaz v takej forme, ktorá kryptografickými metódami zaručuje integritu dokumentu, autentizáciu a nepopierateľnosť podpísaného. Digitálne podpisy sa vytvárajú pomocou algoritmov generujúcich podpisov a na druhej strane ich pôvodnosť je overená pomocou verifikačných algoritmov (kontrola či naozaj bola vytvorená správa tou dôveryhodnou osobou). Digitálna podpisová schéma sa skladá z algoritmu generujúceho podpisu a z priradeného verifikačného algoritmu.

Výstupy digitálnych podpisových schém sú formátované do použiteľnej podoby pomocou už zmienených formátovaní z časti 1.6. Pre schémy s obnovou správy je používané formátovanie ISO/IEC 9796, a pre schémy s dodatkom formátovanie PKCS.

Obecne môžeme kategorizovať digitálne podpisy na priame digitálne podpisy (priama komunikácia dvoch subjektov) a verifikované digitálne podpisy (využíva pri komunikácii tretiu dôveryhodnú stranu – arbitra). Architektúra verifikovaných digitálnych podpisov bola popísaná v časti 1.5.

2.1 Priame digitálne podpisové schémy

Pre overenie integrity správy a autentizáciu podpísaného používajú priamu komunikáciu dvoch strán. Podľa vymenených informácii komunikujúce strany sa môžu rozhodnúť či je správa dôveryhodná alebo nie, nepotrebujú k tomu tretiu dôveryhodnú stranu.

Priame digitálne podpisové schémy sa delia na digitálne podpisové schémy s dodatkom, ktoré k overeniu správy potrebujú celú originálnu správu, a na digitálne podpisové schémy s obnovou správy, ktoré k overeniu správy nepotrebujú celú originálnu správu. V tomto prípade originálna správa je obnovená zo samotného podpisu. Podľa ďalšej vlastnosti podpisové schémy môžu byť deterministické, keď pre jednu správu vždy generujú ten istý podpis, a náhodné, keď z tej istej správy generujú vždy iný podpis.

Digitálne podpisové schémy s dodatkom

Tento typ digitálnych podpisov je najčastejšie používaný v súčasnosti. Opiera sa o hašovacie funkcie a k overeniu dôveryhodnosti prijatej správy prijímateľ potrebuje celú originálnu správu.

Typickými príkladmi mechanizmov vytvárajúcich digitálnych podpisov s dodatkom sú podpisové schémy DSA („Digital Signature Algorithm“), ElGamal, Schnorr a RSA.

Digitálne podpisové schémy s obnovou správ

Tieto schémy majú vlastnosť, že prijímacia osoba nepotrebuje obdržať celú správu, zo samotného podpisu je schopný pomocou určitých algoritmov obnoviť a získať celú originálnu správu. V praxi sú používané tieto schémy len pre krátke správy.

Typickými príkladmi mechanizmov produkujúcich digitálne podpisy s obnovením správy sú podpisové schémy RSA, Rabin a ElGamal s obnovou správy.

Z každej digitálnej podpisovej schémy s obnovou správy môže byť vytvorená jednoduchým hašovaním správy a podpísaním vytvoreného haša digitálna podpisová schéma s dodatkom.

2.2 Digitálne podpisové schémy súvisiace s RSA

Bezpečnosť podpisovej schémy RSA a metód blízko súvisiacich s ním je založená na probléme faktorizácie celých čísiel. Obe nasledujúce schémy majú aj variantu s dodatkom.

RSA podpisová schéma

Je to deterministická podpisová schéma s obnovou správy, ktorá pre podpisovanie správ a verifikáciu podpisov používa algoritmus RSA. Princíp algoritmu už bol popísaný v časti 1.3. Integrovaním hašovacej funkcie (napr. MD5) do podpisovej schémy môžeme vytvoriť RSA digitálnu podpisovú schému s dodatkom, ktorá v praxi môže byť použitá aj pre dlhšie správy.

V dnešnej dobe firma RSA Laboratories [16] odporúča používanie kľúča s dĺžkou 2048 bitov pre komerčné použitie a ešte dlhšie kľúče v prípadoch, ak je vyžiadaná extrémne vysoká bezpečnosť. Pre bežné účely postačí používanie kľúča s dĺžkou 1024 bitov, aj tak bude informácia dobre zašifrovaná a kľúč bude odolný voči všetkým známym algoritmom na prelomenie šifrovacieho kľúča. Používanie dlhších kľúčov znamená samozrejme vyššiu bezpečnosť, ale na druhej strane bohužiaľ aj pomalšie šifrovanie, čo v niektorých prípadoch môže byť veľkou nevýhodou. Avšak pre dlhodobé podpisy, ako sú napr. archivované dokumenty, by sa mali použiť dlhšie kľúče, jednak kvôli bezpečnosti, a v budúcnosti so silnejšími technológiami už doba overenia bude stále krátiť.

RSA je veľmi rýchle v operácii overovania podpisu a to aj s absurdnou dĺžkou kľúča, kde je potrebné 1 exponenciálne násobenie. Naopak RSA je pomalé v operácii podpisovania, Napríklad pri použití 768 bitového modula je potrebné 1152 modulárnych násobení. Dôvodom pomalého podpisovania je používanie dlhého súkromného kľúča. Na urýchlenie algoritmu RSA preto používa metódy napr. CRT („Chinese Remainder Theorem“) alebo Multiprime RSA [18]. Pre implementáciu na čipovej karte je ale v týchto prípadoch nutná prítomnosť špeciálneho koprocesoru, čo karty zdražuje. Problémom sa stáva implementácia RSA na čipových kartách aj v okamžiku zväčšovania kľúča. Napríklad v roku 2000 implementovalo kľúč o dĺžke 2048 bitov len niekoľko kariet. V súčasnej dobe je implementácia 2048 bitového RSA v čipových kartách pomerne bežná a je otázkou, či sa budúcnosť vydá cestou ďalšieho zväčšovania RSA kľúčov alebo cestou eliptických kriviek.

Dĺžka podpisu v RSA rastie lineárne s dĺžkou kľúča. Pri dnešnej dĺžke podpisu s dodatkom približne 2 kb (pre kľúč dĺžkou 2048 bitov). Pre veľkosť kľúča navrhovanej v horizonte 2030 produkuje RSA veľmi dlhé podpisy, čo môže byť v niektorých systémoch veľkou nevýhodou.

Ide o veľmi dobre preskúmanú záležitosť, ktorá odoláva úspešne snahám o ich prelomenie viac ako štvrť storočia. Tento algoritmus je už veľmi dlho skúmaný, bolo v ňom nájdené veľké množstvo potenciálnych slabín súvisiacich hlavne so zlou implementáciou algoritmu. Fakt, že všetkým snahám zatiaľ odolal je ďalší veľký plus. Posledným argumentom je všeobecná rozšírenosť RSA, dobrá štandardizácia a skúsenosti s implementáciou.

Medzi nevýhody RSA je možné zaradiť, že zlá implementácia môže ponúknuť slabiny a predsa len tiež všeobecnú známosť rôznych útokov na implementačné slabiny.

RSA algoritmy v dnešnej dobe majú veľmi široké použitie. Sú použité vo viacerých komerčných aplikáciách a je v pláne jeho použitie aj v budúcnosti. RSA algoritmus je využitý v dnešných operačných systémoch od Microsoftu, Apple, Sunu a Novellu. Tento algoritmus môžeme nájsť aj v hardvérových produktoch ako sú napr. zabezpečené telefóny, Ethernetové sieťové karty a smartkarty. Technológia je použitá vo všetkých známych protokoloch zaisťujúcich bezpečnú komunikáciu na internete ako SSL, S/MIME a S/WAN, ale aj na interne vo viacerých inštitúciách ako sú napr. vládne úrady, veľké firmy, národné laboratóriá a univerzity.

Rabin podpisová schéma

Je to deterministická metóda digitálneho podpisu, ktorú zverejnil Michael O. Rabin v roku 1979. Táto podpisová schéma bola jednou z prvých navrhnutých digitálnych podpisových schém, a bola prvou, ktorá spojila ťažkosť falšovania s problémom faktorizácie celého čísla.

Generovanie podpisu v Rabinovej podpisovej schéme nie je výraznejšie výpočtovo náročnejší ako generovanie v RSA s rovnakým veľkým modulom. Verifikácia podpisu je veľmi rýchla, v niektorých prípadoch potrebuje len jedno modulárne násobenie. Viac o tejto schéme je popísané v [11].

Algoritmus tejto podpisovej schémy je podobný k RSA s efektívnym využitím šírky pásma. Verejným kľúčom je celé číslo $n = p \,.\, q$, kde p a q sú prvočísla, ktoré formujú privátny kľúč. Správa čakajúca na podpis musí obsahovať druhú odmocninu mod n, v inom prípade správa musí byť trošku pozmenená. Približne ¼ všetkých možných správ môže mať druhú odmocninu mod n. Podpísanie je prevedené nasledovne: $s = M^{1/2} \bmod n$, kde M je originálna správa a s je vytvorený podpis. Verifikácia podpisu prebieha pomocou rovnice $M' = s^2 \bmod n$, kde M' je prijatá správa.

2.3 Podpisové schémy Fiat-Shamir

Akákoľvek identifikačná schéma zahŕňajúca sekvenčnú odpoveď svedok-výzva (witness-challenge) môže byť prevedená na podpisovú schému s nahradením náhodnej výzvy s jednosmernou hašovaciou funkciou. Nasledujúce dve podpisové schémy vznikli týmto spôsobom. Základom tejto metodiky je Fiat-Shamirov identifikačný protokol [11].

Feige-Fiat-Shamir

Feige-Fiat-Shamirová podpisová schéma je modifikáciou predchádzajúcej Fiat-Shamirovej schémy, a používa jednocestnú hašovaciu funkciu. Výstupom tejto náhodnej metódy je digitálny podpis s dodatkom.

Na podpísanie správy RSA schémou používajúcou 768 bitové modulo je potrebné previesť 1152 modulárných násobení. Podpísať správu Feige-Fiat-Shamirovou schémou tým istým modulom je potrebné 64 modulárnych násobení, čo znamená menej ako 6% práce oproti podpísaniu správy takým istým kľúčom naivnou implementáciou RSA. Verifikácia podpisu prebieha pomalšie (64 modulárnych násobení) ako u RSA ale stále relatívne rýchlo.

Nevýhodou tejto schémy je, že používa dlhé kľúče (podľa [11] pri 768 bitovom module súkromný kľúč je 98304 bitový, a verejný kľúč 99072 bitový), a preto treba v systéme mať väčšiu pamäť pre uloženie kľúčov.

V aplikáciách, kde generovanie podpisov musí prebiehať rýchle a priestor pre uloženie kľúčov nie je limitovaný, Feige-Fiat-Sahmirová schéma môže byť výhodnejšia ako RSA. Táto podpisová schéma s používaným algoritmom je podrobne popísaná v [11].

Guillou-Quisquater (GQ)

Z Guillou-Quisquater identifikačného protokolu môže byť vytvorený mechanizmus pre vytváranie digitálnych podpisov pridaním jednocestnej hašovacej funkcie.

Generovanie podpisu so 768 bitovým modulom znamená podľa [11] priemerne 384 modulárnych násobení. Verifikácia podpisu je podobne náročná ako generovanie. Pre porovnanie, RSA naivne potrebuje pre generovanie podpisu 1152 a Feige-Fiat-Shamir 64 modulárnych násobení. GQ je výpočtovo o niečo náročnejšie ako Feige-Fiat-Shamir ale postačí mu výrazne menší úložný priestor pre kľúč, lebo v GQ pri 768 bitovom module súkromný kľúč je 768 bitový a verejný kľúč približne o pár desiatky bitov dlhší.

2.4 Podpisové schémy súvisiace s DSA

Do tejto skupiny patrí digitálny podpis DSA a niekoľko digitálnych podpisových schém úzko súvisiacich s ním. Všetky tieto schémy sú náhodné, vytvárajú digitálne podpisy s dodatkom a môžu byť zmenené na digitálne podpisy s obnovou správy. Základom bezpečnosti všetkých týchto schém je, aby výpočet diskrétneho logaritmu nebol možný v reálnom čase.

DSA podpisová schéma

DSA je algoritmom pre americký vládny štandard DSS („Digital Signature Standard“) a používa sa výhradne k podpisovaniu správ. Samotný algoritmus je založený na probléme výpočtu diskrétneho logaritmu a je jedným z variantov algoritmu ElGamal.

Malá dĺžka súkromného kľúča oproti RSA zrýchľuje operáciu podpisovania. Rýchlosť overenia podpisu je významne pomalšia ako v RSA. Vyžaduje dve umocnenia modula, každé do 160 bitového exponenta. V priemere obe potrebujú 240 modulárnych násobenia, dokopy teda 480. Tieto umocnenia môžu byť urobené súčasne, a tým pádom redukované na 280

modulárnych násobení. Najnáročnejšia operácia v DSA je generovanie parametrov, lebo je nutné nájsť jedno prvočíslo v plnej veľkosti. RSA oproti tomu hľadá dve prvočísla polovičnej veľkosti a tak funguje s ich súčinom. Pri 768 bitovom module generovanie podpisu potrebuje 1 modulárne umocnenie (rovná sa približne 240 modulárnych násobeniu), 1 modulárnu inverziu s 160 bitovým modulom, dve 160 bitové modulárne násobenia a jedno sčítanie. Tieto 160 bitové operácie sú oveľa menej náročnejšie ako tá exponenciálna. DSA má tú výhodu, že to modulárne umocnenie môže byť spravené ešte pred generovaním podpisu, a tým pádom celý proces môže byť výrazne urýchlený [11].

Pri nasadení DSA z hľadiska bezpečnosti je treba zvážiť hlavne potenciálne problémy, ktoré môžu vzniknúť pri implementácii. Sú to:

- Pomalé generovanie parametrov a súvisiaca nutnosť validácia parametrov, keď je ich generovanie prenechané externému zdroju.
- Nutnosť zaistenia integrity parametrov DSA.
- Nutnosť zaistenia utajenia a neopakovateľnosti kľúčov.

DSS bolo zverejnené s pevnou dĺžkou kľúča 512 bitov. Po viacerých kritikách, že nie je takto systém dostatočne bezpečný, hlavne pre dlhodobé použitie, bola upravená použiteľná dĺžka kľúčov až do 1024 bitov. DSA v súčasnosti je považované za bezpečné s 1024 bitovým kľúčom. Dĺžka vytvoreného podpisu je vždy len 320 bitov.

Napriek tomu, že niektoré aspekty DSA boli kritizované, tento algoritmus od jeho uvedenia bol implementovaný do viacerých systémov a špecifikácií. Kritika bola zameraná na jeho nedostatky: nemá takú flexibilnosť ako kryptosystém RSA; verifikácia podpisov je príliš pomalá; a existencia druhého autentizačného mechanizmu je náročná pre výrobcov počítačového hardvérov a softvérov, ktorý už štandardizovali svoje produkty s algoritmom RSA. Viac o tejto podpisovej schéme viď. [18].

DSA princíp

Generovanie spoločných prvkov verejného kľúča:

1. Zvolia sa prvočísla p a q, pričom q delí $p - 1$.
2. Vypočíta sa $g = h^{(p-1)/q}$ pričom $1 < h < (p - 1)$ a zároveň $h^{(p-1)/q} \bmod p > 1$.

Generovanie kľúčov:

1. Súkromný kľúč SK je náhodne vygenerovaný, pre ktorý platí $0 < SK < q$.

2. Vypočíta sa parameter y: $y = g^{SK} \bmod p$.
3. Vyberie sa náhodné číslo k, pričom $0 < k < q$.
4. Verejným kľúčom je sada čísel (p, q, g, y).
5. Zdieľané sú parametre p, q, g skupinou užívateľov, ale k musí ostať tajný.

Generovanie digitálneho podpisu (dvojica r, s):

1. Najprv sa vypočíta r: $r = (g^k \bmod p) \bmod q$.
2. Potom číslo s: $s = (k^{-1} (H(M) + SK. r)) \bmod q$, kde M je pôvodná správa a $H(M)$ hašovaciou funkciou SHA-1.

Overenie prijatého digitálneho podpisu (dvojica $r´$, $s^$):

1. Najprv treba zistiť, že platí $0 < r´ < q$ a $0 < s´ < q$. Keď neplatia tieto tvrdenia, tak podpis je neplatný.
2. Potom sa vypočítajú pomocné parametre w, $u1$, $u2$ a v:

   $w = (s^) - 1 \bmod q$,

   $u1 = (H(M^). w) \bmod q$, kde $M´$ je prijatá správa,

   $u2 = (r´. w) \bmod q$,

   $v = ((g^{u1}. y^{u2}) \bmod p) \bmod q$.
3. Keď $r´ = v$, tak podpis a správa sú prehlásené za autentické.

ECDSA podpisová schéma

ECDSA („Elliptic Curve Digital Signature Algorithm“) je variantom DSA, ktorý používa eliptické krivky. Eliptická krivka je množina bodov, ktorá vyhovuje rovnici eliptickej funkcii. Problém diskrétneho logaritmu v eliptických krivkách má za sebou v súčasnej dobe 20 rokov skúmania a zatiaľ zostáva obťažným problémom s exponenciálnou zložitosťou riešenia. Toto môže viesť k používaniu kratších kľúčov a lepšej výkonnosti kryptosystému v tých istých operáciách, pri rovnakej úrovni bezpečnosti. Zdokonalenia vo viacerých aspektoch implementácie, vrátanie generovania eliptických kriviek, spravili z týchto systémov prakticky použiteľnejšie ako boli prvýkrát zverejnené v polovici 80. rokoch.

Podľa informácii z [16], kryptosystémy založených na eliptických krivkách s 160 bitovým kľúčom poskytujú rovnakú bezpečnosť ako RSA systémy s 1024 bitovým kľúčom. S využívaním eliptických kriviek v kryptosystémoch sa dá výrazne zmenšiť dĺžku použitého verejného kľúča. Tieto algoritmy sú obecne rýchlejšie ako odpovedajúce algoritmy na báze diskrétneho logaritmu. Podpisovanie a dešifrovanie je rýchlejšie v týchto systémoch, ale

verifikácia podpisov a šifrovanie je o niečo rýchlejšie v systémoch RSA. Dĺžka vytvorených podpisov je rovnaká ako u DSA.

Malá dĺžka kľúčov a s tým súvisiaca malá výpočtová náročnosť predurčuje ECDSA k nasadeniu v zariadeniach s obmedzenou výpočtovou kapacitou a limitovanou pamäťou ako sú čipové karty, mobilné telefóny a podobné. V najbližšej budúcnosti je možné očakávať nasadenie eliptických kriviek v týchto oblastiach. V dlhšom časovom horizonte je možné očakávať pomalý posun od tradičných schém smerom k eliptickým krivkám. Keď sa medzitým neobjaví žiadny algoritmus zásadne urýchľujúci riešenie problému diskrétneho logaritmu v eliptických krivkách, je možné im predpovedať veľkú budúcnosť.

ECDSA princíp

Keď chce Alice poslať správu M Bobovi, musí sa najprv dohodnúť na parametroch (p, a, b, G, n, h), kde p je prvočíslo, ktorým definuje teleso, konštanty a, b z rovnici eliptickej krivky, bod G na eliptickej krivke, jeho rad n a kofaktor h, ktorý udáva podiel počtu prvkov grupy bodov na eliptickej krivke a radu bodu G. Alice musí mať svoje kľúče vhodné pre kryptografiu eliptických kriviek skladajúce sa zo súkromného kľúča SK, verejného kľúča VK, kde SK je náhodne vybrané celé kladné číslo z intervalu $[1; n - 1]$, $VK = SK \,.\, G$.

Generovanie digitálneho podpisu (dvojica r, s):

1. Alice náhodne zvolí k z intervalu $[1; n - 1]$.
2. Potom spočíta $r = x_1 \bmod n$, kde $k \,.\, G = [x_1; y_1]$. Keď $r = 0$, musí Alice zvoliť iné k.
3. Zistí haš správy $H(M)$, spočíta $s = k^{-1}(H(M) + r \,.\, SK) \bmod n$.
4. Čísla r, s tvoria podpis.

Overenie prijatého digitálneho podpisu (dvojica $r´$, $s´$):

1. Bob musí zverejniť verejný kľúč Alice VK. Keď má pochybnosti ohľadne zdroja, musí si overiť tento kľúč.
2. Overí, že $VK \neq O$ a leží na eliptickej krivke, kde O je bod v nekonečne.
3. Overí, že existuje číslo n také, že $n \,.VK = O$.
4. Keď všetko platí, tak Bob overí, že $s´$ a $r´$ sú z intervalu $[1; n - 1]$. Keď nie, podpis je neplatný.
5. Bob zistí haš správy $H(M´)$, kde $M´$ je prijatá správa.
6. Spočíta $w = s´^{-1} \bmod n$, u_1 a u_2, kde $u_1 = w \,.\, H(M´) \bmod n$ a $u_2 = r´ \,.\, w \bmod n$.
7. Následne spočíta súradnice $[x_1, y_1] = u_1 \,.\, G + u_2 \,.\, VK$.

8. Podpis je platný, keď $r´ = x_1 \bmod n$.

ElGamal podpisová schéma

Podpisová schéma ElGamal je podpisovým mechanizmom, ktorý generuje z binárnych správ o ľubovoľnej veľkosti digitálne podpisy s dodatkom, a používa hašovaciu funkciu. DSA je variantom podpisového mechanizmu ElGamal.

Generovanie podpisu v schéme ElGamal potrebuje 1 modulárne umocnenie, rozšírený Euclidov algoritmus a 2 modulárne násobenia. Generovanie môže byť urýchlené tým, že umocnenie a aplikácia rozšíreného Euclidovho algoritmu sú urobené off-line, a tým pádom generovanie podpisu (v prípadoch keď je výpočet dopredu realizovateľný) môže byť spravené v 2 (on-line) modulárnych násobeniach. Verifikácia podpisu je náročnejšia, lebo potrebuje 3 umocnenia. S modifikovaním verifikácii je možné dosiahnuť, aby všetky tri umocnenia prebehli súčasne, a tým celý proces je 2,5-krát efektívnejší ako pôvodne. V súčasnosti na dlhodobé použitie je doporučené používať minimálne 1024 bitové modulo. Veľkosť vytvoreného podpisu je dvojnásobkom veľkosti použitého modula.

ElGamal digitálna podpisová schéma môže byť modifikovaná tak, aby pracovala na konečnej abelian skupine *G*. Takto modifikovaná schéma je nazývaná obecnou ElGamal podpisovou schémou. Abelian skupina znamená množinu prvkov, v ktorej operácia v rámci množiny je komutatívna (napr. $h \,.\, g = g \,.\, h$). Bezpečnosť tejto schémy je založená na probléme výpočtu diskrétneho logaritmu na skupine *G*. Generovanie a verifikácia podpisu používajú výpočty na skupine *G*. Viac o tejto podpisovej schéme je popísané v [8].

Jedným z najperspektívnejších variantov pre budúcnosť je implementácia obecného ElGamal algoritmu v prípade, keď konečná abelian skupina *G* je zostavená z bodov na eliptickej krivke cez konečné pole F_q. Problém diskrétneho logaritmu v takýchto skupinách sa javí ťažšie riešiteľným ako problémy použité v predchádzajúcich variantov ElGamal schémy. Z toho faktu vyplýva, že môžu byť použité kratšie kľúče pri zachovaniu tej istej mieri bezpečnosti.

ElGamal princíp

Generovanie kľúčov:

1. Vyberie sa veľké prvočíslo *p* aby výpočet diskrétneho logaritmu bol náročný.
2. Zvolia sa náhodne číslo *g* a súkromný kľúč *SK*, pre ktorých platí $g < p$, $SK < p$.

1. Vypočíta sa parameter y: $y = g^{SK} \bmod p$.
3. Verejným kľúčom je trojica (g, p, y).
4. Zdieľané sú čísla g, p skupinou užívateľov.

Generovanie digitálneho podpisu (dvojica a, b):

2. Vyberie sa náhodné číslo k, ktoré nie je súdeliteľné s $(p - 1)$.
3. Prvým parametrom podpisu je a: $a = g^k \bmod p$.
4. Číslo b sa vypočíta z rovnice $M = (SK \,.\, a + k \,.\, b) \bmod (p - 1)$, kde M je originálna správa.

Verifikácia prijatého digitálneho podpisu:

1. Ak platí rovnosť $y^a a^b \bmod p = g^{M'} \bmod p$, kde M' je prijatá správa, tak podpis je považovaný za platný.

Schnorr podpisová schéma

Ďalším známym variantom schémy ElGamal je podpisová schéma Schnorr, ktorá používa algoritmus s rovnakým názvom. Generovanie kľúčov prebieha podobne ako u DSA, ale s tým rozdielom, že nie je obmedzená veľkosť parametrov p a q.

Generovanie podpisu potrebuje 1 modulárne umocnenie a 1 modulárne násobenie, z ktorých modulárne umocnenie môže byť spravené off-line. Verifikácia podpisu potrebuje 2 modulárne umocnenia. Algoritmus použitý v tejto schéme nie je výpočtovo efektívnejší ako v schéme ElGamal, ale vytvára menšie podpisy pri zachovaní tej istej miere bezpečnosti. Viac o tejto schéme je popísané v [17].

2.5 Jednorazové digitálne podpisové schémy

Jednorazové digitálne podpisy sú podpisové mechanizmy, ktoré môžu byť použité len raz. Pre podpísanie každej správy je vyžiadaný nový verejný kľúč. Verejné informácie potrebné k overeniu jednorazových podpisov sú často označované za validačných parametrov. Keď jednorazové podpisy sú kombinované s technikami pre autentifikáciu validačných parametrov, môžu sa vytvárať zložené podpisy.

Väčšinou jednorazové podpisové schémy majú výhodu, že generovanie a verifikácia podpisu je veľmi rýchla. Tieto podpisové schémy je výhodné použiť v čipových kartách, kde je požadovaná nízka výpočtová zložitosť.

Rabin jednorazová podpisová schéma

Táto jednorazová podpisová schéma bola jedným z prvých návrhov na digitálny podpis. Umožňuje podpísanie jednej správy a verifikácia podpisu vyžaduje interakciu medzi podpisujúcim a verifikujúcim. Verifikácia, rozdielne od iných digitálnych podpisových schém, môže byť prevedená len raz. Pri optimálne vybraných parametroch verejný a súkromný kľúč sú 1280 bitové. Táto schéma v praxi sa už vôbec nevyužíva, a je podrobne popísaná v [11].

Merkle jednorazová podpisová schéma

V tejto podpisovej schéme, na rozdiel od Rabin jednorazovej podpisovej schémy, verifikácia podpisu nie je interaktívna s podpisujúcim. Nejaký dôveryhodný prostriedok je vyžiadaný pre autentizáciu validačných parametrov. Výhodou tejto schémy je, že nie je riešiteľná kvantovým počítačom. Pri optimálne vybraných parametroch sú verejný a súkromný kľúč 8704 bitové, a samotný podpis zaberie 4800 bitov. Generovanie podpisu je veľmi rýchle, lebo nevyžaduje žiadny výpočet. Verifikácia podpisu záleží na použitej hašovacej funkcii. Táto schéma je podrobne popísaná v [11].

Autentizačné stromy a jednorazové podpisy

Na autentizáciu väčšieho počtu verejných validačných parametrov jednorazových podpisových schém môžu byť použité autentizačné stromy. Autentizačné stromy spojené s jednorazovými podpismi môžu vytvárať schémy, ktoré umožňujú zložené podpisy. Viac o týchto schém je popísané v [11].

2.6 Ostatné podpisové schémy

Podpisové schémy popísané v tejto časti prirodzene nespadajú ani do jednej z predchádzajúcich kategórií.

Arbitrované digitálne podpisy

Sú to digitálne podpisy, ktoré vyžadujú tretiu dôveryhodnú (TTP = „Trusted Third Party“) stranu pre generovanie a verifikáciu podpisu. Algoritmus generovania a verifikáciu podpisu používa symetrický šifrovací algoritmus. K verifikácii podpisu symetrický kľúč musí byť zdieľaný s TTP. Bezpečnosť tejto schémy záleží na bezpečnosti zvolenej symetrickej šifre

a na schopnosti distribúcii tajných kľúčov. Symetrické šifrovacie algoritmy sú oveľa rýchlejšie ako asymetrické, takže generovanie a verifikácia podpisu sú veľmi rýchle. Len komunikácia s TTP môže trošku spomaliť túto schému. Princíp použitého algoritmu je popísaný v [18].

ESIGN

ESIGN („Efficient digital SIGNature“) je ďalšou digitálnou podpisovou schémou, ktorej bezpečnosť záleží na obťažnosti faktorizácie celých čísiel. Je to schéma s dodatkom a používa jednocestnú hašovaciu funkciu. Na rozdiel od RSA, modulo sa vypočíta podľa rovnice: $n = p^2 . q$. Generovanie podpisov môže byť 10 až 100-krát rýchlejšie ako v RSA s rovnakým modulom. Verifikácia podpisu je tiež veľmi rýchla a výkonnostne zrovnateľná s RSA. Použitý algoritmus je popísaný v [13].

2.7 Podpisy s prídavnými funkciami

Táto skupina digitálnych podpisov kombinuje nejakú základnú digitálnu podpisovú schému (napr. RSA) so špecifickým protokolom aby získala prídavné funkcie, ktoré základná metóda nepodporuje.

Slepé podpisové schémy

Slepé podpisové schémy sú dvojstranné protokoly medzi posielajúcim *A* a podpisujúcim *B*. Základná myšlienka je nasledujúca: *A* pošle nejakú informáciu *B*, ktorú *B* podpíše a vráti naspäť *A*. Na základe tohto podpisu *A* je schopný vypočítať podpis od *B* na základe predchádzajúcej správy *M*. Na konci toho *B* nevie o správe *M* ani o podpise pripojeného k nej. Cieľom slepého podpisu je predbehnúť, aby podpisujúci *B* pozoroval správu, jeho podpísanie a samotný podpis, a preto nie je možné neskôr asociovať podpísanú správu s posielajúcim *A*. Podrobný popis je napísané v [10].

Neodmietnuteľné podpisové schémy

V neodmietnuteľných podpisových schémach verifikačný protokol spolupracuje s podpisujúcim. Prvou charakteristickou vlastnosťou týchto schém je, že verifikácia je interaktívna, takže podpisujúca osoba vie limitovať, že kto môže podpis overovať. Obsahuje tzv. nepopierateľný protokol, ktorý zaisťuje, že odoslaný podpis určite nie je falošný.

Bezpečnosť týchto schém je založená na probléme diskrétneho logaritmu v cyklickej podskupine na množine celých čísiel. Podrobný princíp a ďalšie vlastnosti sú popísané v [1].

Fail-stop podpisové schémy

Fail-stop digitálne podpisy umožňujú užívateľovi *A* preukázať, že daný podpis podpísaným s *A* je falošný. Toto je dosiahnuté ukázaním, že základný predpoklad, na ktorom je založený podpisový mechanizmus, bol kompromitovaný. Schopnosť dokazovania falšovania nespolieha na žiadny kryptografický predpoklad. Fail-stop podpisové schémy majú tú výhodu, že aj keď útočníkovi sa nejako podarí falšovať podpis, môže to byť odhalené a podpisový mechanizmus sa ďalej už používať nebude. Viac o týchto podpisoch je popísané v [19].

3 EFEKTÍVNE DIGITÁLNE PODPISOVÉ SCHÉMY

Táto kapitola obsahuje niektoré návrhy nových efektívnych podpisových schém. Perspektívne sú schémy bez certifikátov, krátke podpisové schémy používajúce bilineárne párovanie, hypereliptické krivky a schémy odolné voči najnebezpečnejších útokov (napr. založené na silnom RSA predpoklade).

3.1 Digitálne podpisové schémy bez certifikátov

Asymetrické kryptosystémy bez digitálnych certifikátov sú veľmi atraktívne v bezdrôtovej komunikácii, kde je limitovaná šírka prenosového pásma a výpočtový výkon bezdrôtového komunikačného zariadenia. Na elimináciu digitálneho certifikátu, Shamir uviedol nový koncept kryptosystému na základe identity užívateľa (ID-based cryptosystem). Najväčšou výhodou ID-based kryptosystémov je, že za verejný kľúč nepoužívajú náhodné celé číslo ako je to v klasických asymetrických kryptosystémoch, ale hašovanú formu identity samotného užívateľa, čo môže byť jeho meno alebo emailová adresa. Všetky podpisové schémy na základe identity (IBS = „Identity Based Signature“) majú ten problém, že generátor súkromného kľúča (PKG = „Public Key Generator“) pozná privátny kľúč každého užívateľa. Z toho vyplýva, že držiteľ PKG je schopný podpísať každú správu na strane užívateľa, čo ohrozuje nepopierateľnosť digitálnych podpisov. Na vyriešenie problému uchovania súkromného kľúča IBS pri zachovaní jeho predností bol uvedený digitálny podpis bez certifikátu (CLS = „Certificateless Signature“).

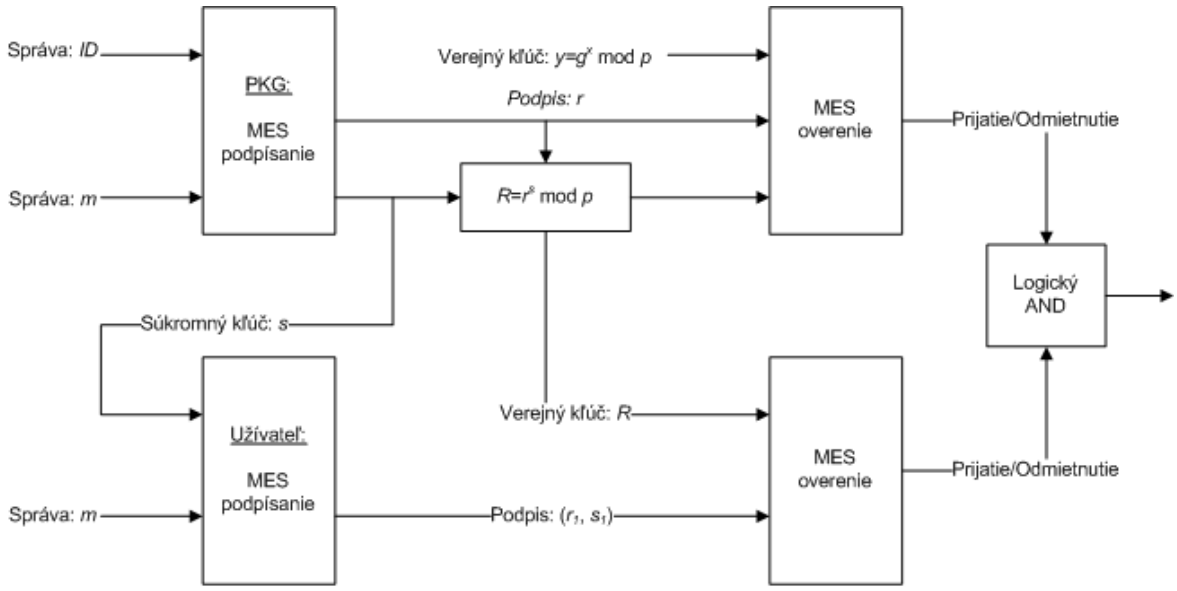
Digitálny podpis na báze DL bez certifikátu

Harn, Ren a Lin publikovali článok [9], v ktorom navrhli pomerne jednoducho implementovateľnú obecnú metódu k vytvoreniu CLS schém z digitálnych podpisových schém typu ElGamal. Cieľom tejto metódy je vytváranie digitálnych podpisov bez certifikátu, kde pri verifikáciu podpisu užívateľovi stačí vedieť identitu jeho komunikačného partnera a verejný kľúč autority PKG. Táto vlastnosť je veľmi užitočná napr. v bezdrôtovej komunikácii, kde predistribúcia autentizovaných kľúčov nie je možná. V ich schéme len sám užívateľ pozná svoj súkromný kľúč, a nie je uložený na PKG. Táto metóda môže byť aplikovaná vo všetkých podpisových schémach postavených na probléme diskrétneho logaritmu na konvertovanie digitálnej podpisovej schémy na CLS schému. Generovanie

podpisu potrebuje 1 modulárne umocnenie a verifikácia 4 modulárne umocnenia (čo je dvojnásobné ako u typu ElGamal). Vytvorená CLS schéma je odolná voči útoku typu adaptive chosen-message attack [11], čo je najvyššia možná bezpečnosť u digitálnych podpisoch, a generovanie podpisu a jeho overovanie je efektívne.

V obecnej podpisovej schéme na báze identity užívateľa PKG generuje súkromné kľúče pre užívateľov pomocou svojho súkromného kľúča. Z tohto vyplýva, že PKG vie podpísať hocijakú správu na strane užívateľa. Tento fakt narušuje nepopierateľnosť digitálnych podpisov.

V schéme popísanej v článku [9] je tento problém vyriešený tým, že PKG vypočíta len čiastočný súkromný kľúč pre užívateľov. Užívateľ potom použije tento čiastočný kľúč so svojou tajnou informáciou na vygenerovanie svojho súkromného kľúča. Týmto spôsobom je dosiahnuté, že užívateľov súkromný kľúč nie je dostupný pre PKG. Užívateľ podobným spôsobom si tiež môže vygenerovať svoj verejný kľúč. Užívatelia svoje verejné kľúče môžu zverejniť vo vytvorenom podpise alebo ich nahrať do verejnej zložky. Na získane verejného kľúča sa nevyžaduje žiadna autentizácia, tým pádom sa nepoužíva žiadny certifikát.



Obr. 3.1: Podpisová schéma bez certifikátu na báze identite.

Princíp tejto podpisovej schémy je znázornený na obrázku 3.1, ktorá je jednoducho implementovateľná a použiteľná pre všetky schémy typu ElGamal. Schéma sa skladá zo štyroch algoritmov: generovanie kľúča PKG, generovanie kľúča užívateľa, podpisovanie a overovanie.

Generovanie kľúča PKG:

1. Generovanie veľkého prvočísla p a generátora $g \in Z_p^*$.
2. Zvolí sa náhodný súkromný kľúč $x \in Z_p^*$ a vypočíta sa verejný kľúč $y = g^x \bmod p$.
3. p, g, y sú verejné parametre a x je súkromným kľúčom PKG.

Generovanie kľúčov užívateľa:

1. Užívateľ si vyberie náhodný súkromný kľúč $v \in Z_{p-1}^*$ s $\gcd(v, p-1) = 1$ a vypočíta $u = g^v \bmod p$. $\{ID, u\}$ je poslané k PKG, kde ID je identita užívateľa.
2. PKG vyberie $k \in Z_{p-1}^*$ s $\gcd(k, p-1) = 1$.
3. Potom sa vypočíta $r = u^k \bmod p$.
4. PKG vyrieši lineárnu rovnosť viď. nižšie.
   $H(ID, r) = xr + k\,z \bmod (p-1) \Rightarrow z = k^{-1}(H(ID, r) - x\,r) \bmod (p-1)$, kde H je jednocestná hašovacia funkcia SHA-1.
5. Výstup (r, z) je poslaný k užívateľovi.
6. Užívateľ vypočíta $s = v^{-1} z \bmod (p-1)$ a $R = r^s \bmod p$.
7. s je užívateľov súkromný kľúč a (r, R) je jeho verejný kľúč.

Generovanie digitálneho podpisu ($\sigma = (r, R, r_1, s_1)$):

1. Náhodne sa vyberie $l \in Z_{p-1}^*$ s $\gcd(l, p-1) = 1$ a vypočíta $r_1 = r^l \bmod p$.
2. Vypočíta $s_1 = l^{-1}(H(M, r_1) - s\,r_1) \bmod (p-1)$, kde M je pôvodná správa.
3. Výsledný podpis je $\sigma = (r, R, r_1, s_1)$.

Overenie prijatého digitálneho podpisu ($\sigma' = (r', R', r_1', s_1')$):

1. $g^{H(ID, r')} = y^{r'} R \bmod p$, a
2. $r'^{H(M', r_1')} = R'^{r_1'}\, r'^{s_1'} \bmod p$, kde M' je prijatá správa.
3. V prípade, že obidve rovnosti platia, podpis je akceptovaný, inak je odmietnutý.

Krátky digitálny podpis s bilineárnym párovaním bez certifikátu

Du a Wen vo svojom článku [6] navrhli podpisovú schému bez certifikátu, ktorá sa opiera o ťažkosti riešenia problémov k-CAA („k-Collusion Attack Algorithm“) a Inv-CDHP („Inverse Computational Diffie-Hellman Problem“). Táto schéma obsahuje všetky žiaduce vlastnosti predtým navrhovaných CLS schém, a vyžaduje bežnú hašovaciu funkciu na rozdiel od ostatných CLS schém, ktoré používajú menej efektívnu funkciu MapToPoint (je použitá na

mapovanie informáciu o identite do bodu eliptickej krivky). Podpisovanie je podobne rýchle ako u typu schém ElGamal, a pre overení podpisu je potrebné jedno skalárne násobenie a jedno bilineárne párovanie, čo znamená, že táto schéma je podstatne viac efektívnejšia ako ostatné CLS schémy. Generovaný podpis má 154 bitov, tým pádom táto schéma je vhodná pre široké použitie, a to najmä v úzkopásmových komunikačných prostriedkoch.

Bilineárne párovanie je matematická funkcia, ktorá pomocou nejakého algoritmu jedinečne zakóduje dve čísla do jedného. Použitie tejto funkcii sa javí ako veľmi efektívne v digitálnych podpisových schémach, takže v budúcnosti sa môže očakávať ich časté integrovanie do nových podpisových schém.

3.2 Schémy pre bezdrôtové senzorové siete

Pre zaistenie bezpečnosti bezdrôtových senzorových sieti je dôležité šifrovanie a autentizácia správ posielané medzi senzorovými uzlami. Kľúče pre šifrovanie a spôsoby autentizácií musia byť dohodnuté cez komunikačné uzly. Vzhľadom na obmedzené zdroje, dosiahnutie takejto dohody na kľúčoch v bezdrôtových senzorových sieťach nie je triviálne. **S**účasne používané schémy v bežných sieťach na ustanovenie kľúčov, ako sú schémy postavené na Diffie-Hellmanovom protokole a na asymetrických kryptosystémoch, nie sú vhodné v bezdrôtových senzorových sieťach. Vzhľadom na veľkú pamäťovú náročnosť v rozsiahlejších sieťach s vysokým počtom senzorov výhodnou metódou môže byť predistribúcia kľúčov.

Du, Deng, Han a Varshney navrhli v ich publikovanom článku [7] predistribučnú schému pre senzorové siete s názvom A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. Pre bezpečnú komunikáciu medzi uzlami v tejto schéme postačí 64 bitový kľúč, a podpísanie z energetického hľadiska je 2621-krát efektívnejšie ako v schéme RSA. Pre porovnanie, výpočtová výkonnosť tejto schémy je podobná ako zašifrovanie 3200 bitovej správy pomocou AES. Je obecne známe, že symetrický kryptosystém AES je veľmi efektívny z energetického hľadiska.

3.3 Digitálne podpisy s hypereliptickými krivkami

V počítačovej kryptografii systémy s hypereliptickými krivkami sú veľmi efektívne, ktoré vznikli rozšírením eliptických kriviek. Na rozdiel od nich používajú grupy bodov Jakobiánu, a ich bezpečnosť záleží na probléme diskrétneho logaritmu v hypereliptických krivkách.

Vo svojom článku [4] Deng, Cheng a Gui navrhli transplantáciu kryptografického systému s hypereliptickými krivkami do algoritmu DSA a vytvorili tým digitálny podpis na základe HEC-DSA systému. Tento digitálny podpis môže vyriešiť problém zistenia úplnosti súboru a ID podpisu, čo je obzvlášť vhodné v operáciách na internete, v ktorých treba overiť identitu.

3.4 Digitálny podpis na báze silného RSA predpokladu

Cramer a Shoup v článku [3] navrhli novú podpisovú schému, ktorá je založená na silnom RSA predpoklade (v ang. strong RSA assumption).

Schéma je odolná proti útokom typu adaptive chosen message attack. Tento typ bezpečnosti je najvyšší možný pre digitálnu podpisovú schému, čo znamená, že schéma s takouto bezpečnostnou úrovňou môže byť bezpečne použitá vo všetkých aplikáciách. Ďalšou dôležitou vlastnosťou je, že táto schéma je bezstavová na rozdiel od ostatných dokázane bezpečných schém a aj jej efektívnosť je vyššia ako u týchto schém. Z praktického hľadiska bezstavovosť je rovnako dôležitá vlastnosť ako efektívnosť.

Schéma používa bezpečnostné parametre l a $l´$, pre ktoré platí $l + 1 < l´$. Doporučené hodnoty pre tieto parametre sú $l = 160$ bitov a $l´ = 512$ bitov. $H(x)$ je hašovacia funkcia SHA-1, a pre pozitívne celé číslo n, QR_n označuje podmnožinu Z_n^* štvorcových čísiel.

Generovanie kľúčov:

1. Náhodne sa zvolí prvočíslo p radu $l´$-bitov, pre ktoré platí, že $p = 2p´ + 1$ a $p´$ je prvočíslo.
2. Náhodne sa zvolí prvočíslo q radu $l´$-bitov, pre ktoré platí, že $q = 2q´ + 1$ a $q´$ je prvočíslo.
3. Náhodne sa zvolia $h, x \in QR_n$.
4. Zvolí sa $(l + 1)$ bitové prvočíslo $e´$.
5. Verejný kľúč je $(n, h, x, e´)$ a súkromný kľúč (p, q).

Generovanie podpisu $(e, y, y´)$:

1. Zvolí sa náhodne $(l + 1)$ bitové prvočíslo $e \neq e´$ a $y´ \in QR_n$.
2. Vypočíta sa $x´$ z rovnici $(y´)^{e´} = x´h^{H(M)}$, kde M je pôvodná správa.
3. Vypočíta sa y z rovnici $y^e = xh^{H(x´)}$ faktorizovaním n.

4. Výsledný podpis je $(e, y, y^{\wedge})$.

Overovanie podpisu $(e, y, y^{\wedge})$:

1. Najprv sa kontroluje či e je nepárne $(l + 1)$ bitové číslo odlišné $e^{\prime}$.
2. Vypočíta sa $x^{\prime} = (y^{\wedge})^{e^{\prime}} h^{-H(M^{\prime})}$, kde $M^{\prime}$ je prijatá správa.
3. Vypočíta sa $x = y^{e} h^{-H(x^{\wedge})}$.
4. Keď $x^{\prime} = x$, podpis je platný.

Poznámka k implementovaniu: Pre zrýchlenie overovania a podpisovania, verejný kľúč môže obsahovať h^{-1} miesto h. V generovaniu podpisu najnáročnejšia operácia je výpočet y, čo môže byť zrýchlené nasledovne:

1. $x = h^{a}$, kde a je náhodné číslo $a \bmod p^{\prime} q^{\prime}$ a je uložené v súkromnom kľúči.
2. d je inverzom $e \bmod p^{\prime} q^{\prime}$.
3. Vypočíta sa $y = h^{b}$, kde $b = da + dH(x^{\wedge}) \bmod p^{\prime} q^{\prime}$.

3.5 Porovnanie súčasných a efektívnych podpisových schém

Porovnanie súčasných a efektívnych podpisových schém je rozdelené do troch skupín podľa použitia digitálnych podpisových schém: systémy zamerané na výkon, systémy s limitovaným výkonom a pamäťou, digitálne podpisové schémy pre webové aplikácie. Tabuľka 3.2 obsahuje orientačné parametre všetkých popísaných digitálnych podpisových schém v tejto práci. U každého systému sa dajú parametre rôzne nastaviť.

Typ schémy	Schéma	Typ problému	Generovanie podpisu	Verifikácia podpisu	Dĺžka SK/VK (bitov)	Dĺžka podpisu (bitov)
RSA	RSA	IFP	1 mod. exp., 1 mod. inv.	1 mod. exp.	2048/1088	2048
	Rabin	IFP	1 mod. exp., 1 Jacobi evaulation	-	- /2048	-
Fiat-Shamir	Feige-Fiat-Shamir	haš	-	-	98 k/99 k	-
	GQ	haš	2 mod. exp., 1 mod. mul.	2 mod. exp., 1 mod. mul.	1024/ -	-
DSA	DSA	DLP	1 mod. exp., 2 mod. mul., 1 mod. inv.	2 mod. exp.	160/1024	320
	ECDSA	ECDLP	1 mod. exp., 2 mod. mul., 1 mod. inv.	2 mod. exp.	160/161	321
	ElGamal	DLP	1 mod. exp., 1 ext. Euc. alg., 2 mod. mul.	3 mod. exp.	2048/2048	2048
	Schnorr	DLP	1 mod. exp., 1 mod. mul.	2 mod. exp.	- / -	kratšie ako ElGamal
Jednora-zové	Rabin	IFP	-	-	1280/1280	-
	Merkle	haš	žiadny výpočet	podľa hašovacej funkcii	8704/8704	4800
Ostatné	Arbitrované	symetrický systém	podľa použitého sym. kryp.	podľa použitého sym. kryp.	256/256	-
	ESIGN	IFP	RSA/(10 až 100)	ako RSA	- / -	-
Prídavný-mi funkciami	Slepé	podľa použitého asym. kryp.	podľa použitého asym. kryp.	podľa použitého asym. kryp.	podľa použitého asym. kryp.	podľa použitého asym. kryp.
	Neodmietnutelné	DLP v cyklickej skupine	-	-	- / -	-
	Fail-stop	-	-	-	- / -	-
Bez certifi-kátu	Krátke podpisy s bilineárnym párovaním	Inv-CDHP	1 scalar mul.	1 scalar mul., 1 pairing	- /160	160
	Podpisy na báze DL bez certifikátu	DLP	1 mod. exp.	4 mod. exp.	- / -	výrazne kratšie ako ElGamal
-	párovaná predistribučná schéma pre senzory	-	RSA/2621	-	64/ -	-

Tab. 3.2: Orientačné parametre jednotlivých digitálnych podpisových schém.

Systémy zamerané na výkon

Systémy, v ktorých je prvoradá výkonnosť použitej digitálnej podpisovej schémy a nie je limitovaná pamäť systému by sme mohli rozdeliť podľa toho, či vyžadujú rýchle podpisovanie alebo rýchlu verifikáciu podpisu.

V praxi v dnešnej dobe verifikácia podpisov prebieha oveľa častejšie ako generovanie podpisu. Preto je výhodné mať schémy, v ktorých verifikácia nie je výpočtovo náročnou operáciou. V súčasnosti najpoužívanejšou takouto schémou je RSA. Už sa podarilo faktorizovať (za veľkú cenu) RSA s 768 bitovým kľúčom, a preto sa odporúča používať minimálne 2048 bitové kľúče vzhľadom na zachovanie bezpečnosti podpísaných správ v budúcnosti.

V súčasnosti najrozšírenejšou schémou pre rýchle podpisovanie je schéma DSA, ale perspektívu do budúcna má už len jeho variant ECDSA, ktorá používa oveľa kratšie kľúče pri zachovaní tej istej bezpečnosti ako ostatné schémy tohto typu. Tým pádom je urýchlený celý proces generovania podpisu ale aj overovanie, ktoré je v týchto schémach výpočtovo náročnejšie.

Schémy v ktorých je podpísanie rýchle, ale verifikácia náročná, sú výhodné v tých systémoch, kde verifikácia podpisov je prenechaná nejakému externému zdroju. Príkladom môžu byť senzory, kde sa podpisy overujú na výpočtovo silnom serveri, ktorý zbiera podpísané dáta od senzorov.

Digitálna podpisová schéma Feige-Fiat-Shamir je perspektívna aj z toho hľadiska, že ani generovanie podpisu ani jeho verifikácia nie sú až také náročné. Výhodnou schémou v tých systémoch je, v ktorej je žiadané rýchle podpisovanie, ale aj rýchla verifikácia podpisu. Nevýhodou tejto schémy je, že oproti ostatným používa veľmi dlhé kľúče.

Vzhľadom do budúcna RSA a DSA nie sú natoľko perspektívne, ako sa javia návrhy nových efektívnych schém. Veľkým problémom týchto schém je, že s rastúcou dĺžkou kľúčov výpočtový nárok rastie oveľa rýchlejšie ako ich bezpečnosť. Výkon týchto schém s veľkými kľúčmi je veľmi nízky. U RSA čas potrebný k podpisovaniu rastie s treťou mocninou dĺžky kľúča. K výpočtu podpisu pomocou RSA s 2048 bitovým kľúčom potrebuje približne 8-krát dlhší čas ako s 1024 bitovým kľúčom. DSA sa správa podobne ako RSA s dlhými kľúčmi.

Nové teoretické návrhy svedčia o tom, že používaním eliptických kriviek a bilineárneho párovania sa dá zmenšiť šifrovací a dešifrovací kľúč, a tým zvýšiť výkonnosť celej podpisovej schémy. V budúcnosti sa preto dá očakávať stále väčšie rozšírenie podpisových schém používajúce tieto techniky.

Systémy s limitovaným výkonom a pamäťou

V týchto systémoch je veľmi dôležité, aby použitá podpisová schéma bola výkonnostne efektívnejšia a vytvárala krátke podpisy. V praxi takýmito systémami môžu byť čipové karty, mobilné zariadenia, senzory, atď. V súčasnosti sa najviac používajú RSA a DSA. RSA má výhodu v rýchlej verifikácii podpisu, ale vytvára dlhé podpisy, čo pri vysokom tempe nutnosti zväčšovania kľúčov do budúcna je veľkou nevýhodou v systémoch s limitovanou pamäťou. DSA vytvára oveľa kratšie podpisy ako RSA, ale na druhej strane verifikácia podpisu je náročnejšia u tejto schémy, čo je často žiadaná v týchto systémoch. Bezpečnostné a výkonnostné problémy vzhľadom do budúcna sú rovnaké, ako boli už popísané u systémoch zameraných na výkon.

Výhodou aplikácii v týchto systémoch jednorazovej podpisovej schémy Merkle oproti typom DSA a RSA je, že schéma je odolná proti útoku kvantovým počítačom. Nevýhodou je zas použitie dlhších kľúčov a vytváranie dlhších podpisov.

V senzorových sieťach pre distribúciu kľúčov súčasné podpisové schémy RSA a DSA nie sú použiteľné kvôli obmedzeným zdrojom senzorov. V súčasnosti sú navrhnuté metódy na vyriešenie tohto problému, jedným z nich je popísaná schéma v časti 3.2, ktorá používa veľmi krátke kľúče a jej výkonnostná efektívnosť je porovnateľná so symetrickým kryptosystémom AES.

V budúcnosti sa očakáva v týchto systémoch používanie krátkych podpisových schém s eliptickými krivkami a bilineárnym párovaním bez certifikátov. Tieto schémy sú výpočtovo veľmi efektívne a vytvárajú krátke podpisy, ale zatiaľ sa objavujú len prvé implementácie.

Digitálne podpisové schémy pre webové aplikácie

Najpoužívanejšou podpisovou schémou je v týchto oblastiach RSA. Technológia je použitá vo všetkých známych protokoloch zaisťujúcich bezpečnú komunikáciu na internete ako SSL, S/MIME a S/WAN. Naďalej je použitá aj v operačných systémoch od všetkých známych výrobcov. V týchto systémoch je podpis oveľa častejšie verifikovaný ako generovaný, a preto

vyžaduje podpisovú schému s efektívnou verifikáciou. V súčasnosti RSA je považované za bezpečné len so stále zväčšujúcim kľúčom, a tým pádom sa zväčšuje aj samotný podpis. Vo webovej komunikácii, aby sieť bola menej zaťažovaná pri posielaní digitálnych podpisov, je výhodnejšie mať kratšie digitálne podpisy. Aj v týchto systémoch sa vyskytne už popísaný problém ohľadom bezpečnosti a výkonu RSA s dlhšími kľúčmi.

V budúcnosti digitálna podpisová schéma s použitím HEC-DSA z časti 3.3 môže vyriešiť problém zistenia úplnosti súboru a ID podpisu v operáciách na internete, v ktorých treba overiť identitu. Je možné očakávať nahradenie RSA efektívnejšími schémami, ktoré používajú eliptické krivky alebo bilineárne párovanie. V súčasnosti ešte nie je žiadna efektívnejšia schéma, ktorá by sa masovo rozširovala.

4 IMPLEMENTÁCIA V JAZYKU C#

K implementácii praktickej časti kvôli jeho perspektívnosti bol zvolený programovací jazyk C#. Implementácia má dve aplikácie, server a klient, ktoré majú za účelom demonštrovať vlastnosti vybraných podpisových schém a zrovnať ich vlastnosti a efektívnosť.

4.1 .NET Framework a jazyk C#

Programovací jazyk C# (v angličtine si-sharp) je súčasťou .NET Framework technológie, ktorú vyvinul Microsoft v roku 2002 [12]. Táto technológia je postavená na princípe CLR (Common Language Runtime), teda podporuje vzájomnú kompatibilitu medzi jednotlivými programovacími jazykmi. CLR je virtuálny stroj, ktorý vykonáva všetok kód, ktorý je kompilovaný do univerzálneho jazyka - byte code (Microsoft Intermediate Language - MSIL), ktorý je podobný assembleru. Byte code je strojový kód virtuálneho stroja. Nezáleží teda v akom jazyku bude daná aplikácia naprogramovaná, všetky jazyky sú si plne rovnocenné. NET Framework umožňuje napísať univerzálne potrebné funkcie v externom súbore vo zvolenom programovacom jazyku a tieto súbory skompilovať do DLL knižníc, ktoré je potom možné ďalej využívať pre rôzne aplikácie.

C# je objektovo-orientovaný programovací jazyk, pre ktorý Microsoft si za základ zobral C++ a jazyk Java. C# bolo navrhované s úmyslom vyvážiť silu jazyka C++ a tú spojiť s možnosťou rýchleho programovania „rapid application development", ktoré ponúkali jazyky ako napríklad Visual Basic, Delphi. Bol vytváraný tak, aby bol jednoduchý, moderný, objektovo orientovaným jazykom pre všeobecné použitie. Jazyk a jeho implementácie by mali poskytovať podporu pre nasledovné princípy softvérového inžinierstva, ako silná typová kontrola, kontrola ohraničenia polí, detekcia pokusov na využitie neinicializovaných premenných a automatickú správu pamäte. Dôležitými vlastnosťami je tiež robustnosť, odolnosť a produktivita.

Ako vývojárske prostredie bola použitá voľne dostupná verzia Visual C# 2010 Express. Produkt je neobmedzene použiteľný a zdarma stiahnuteľný z webovej stránky Microsoftu.

Na implementovanie aplikácii pracujúcej s kryptografickými metódami najnovšia verzia .NET Framework 4 poskytuje v knižnici System.Security.Cryptography triedy pre nasledujúce oblasti:

- Symetrické šifrovanie/dešifrovanie.
- Asymetrické šifrovanie/dešifrovanie.
- Haš.
- Digitálny podpis.
- XML podpis.

4.2 Hašovacie funkcie v .NET

Podobne ako šifrovacie algoritmy sú hašovacie algoritmy implementované v .NET Frameworku s použitím troj-stupňovej dedičnosti. Hlavná trieda HashAlgorithm je abstraktná a obsahuje metódy pre preťaženie a výpočet hašu pre vloženú správu buďto vo formáte bytového poľa alebo streamu.

Výpočet hodnoty hašu pre špecifikované bajtové pole:

```
byte[] ComputeHash(byte[] value);
```

Výpočet hodnoty hašu pre špecifikovaný stream:

```
byte[] ComputeHash(Stream str);
```

.NET poskytuje nasledujúce hašovacie algoritmy:

- MD5 - *System.Security.Cryptography.MD5*
- SHA-1 - *System.Security.Cryptography.SHA1*
- SHA-256 - *System.Security.Cryptography.SHA256*
- SHA-384 - *System.Security.Cryptography.SHA384*
- SHA-512 - *System.Security.Cryptography.SHA512*

4.3 Podpis dát v .NET

Microsoft .NET podporuje nasledujúce tri triedy pre podpisovanie dát:

- RSA - *System.Security.Cryptography.RSACryptoServiceProvider*

- DSA - *System.Security.Cryptography.DSACryptoServiceProvider*
- ECDSA - *System.Security.Cryptography.ECDsaCng*

Digitálny podpis pomocou RSACryptoServiceProvider

RSACryptoServiceProvider poskytuje dva sety metód pre digitálny podpis, jednu pre podpis hašu správy a druhú pre podpis čistého textu, z ktorého je najskorej vytvorený haš.

Nasledujúci pár metód je použitý pre podpis hašu správy:

```
public byte[] SignHash(byte[] rgbHash, string str);
public bool VerifyHash(byte[] rgbHash, string str, byte[] rgbSignature);
```

Pre podpis a overenie podpisu čistých dát je použitý nasledujúci pár metód:

```
public byte[] SignData(byte[] buffer, object halg);
public bool VerifyData(byte[] buffer, object halg, byte[] signature);
```

RSACryptoServiceProvider vie pracovať s hašovacími algoritmami SHA-1 a MD5, a podporuje kľúče s dĺžkami od 384 do 16384 bitov.

Digitálny podpis pomocou DSACryptoServiceProvider

DSACryptoServiceProvider obsahuje rovnaký set metód pre ako RSACryptoServiceProvider, ale používa striktne iba SHA-1 a podporuje kľúče od 512 do 1024 bitov.

Digitálny podpis pomocou ECDsaCng

Táto trieda tiež obsahuje rovnaký set metód ako už popísané dve. Na rozdiel od predchádzajúcich dvoch tried ECDsaCng („Elliptic Curve Digital Signature Algorithm Cryptography Next Generation“) nemá obmedzenie v použití hašovacích algoritmoch a dáta vie podpísať 256 alebo 384 bitovým kľúčom. Z vyššie uvedených faktov vyplýva, že jedine 160 bitov dlhá SHA-1 je podporovaná vo všetkých troch triedach.

4.4 Práca s veľkými číslami v .NET

Samotný .NET Framework umožňuje v najnovšej verzii 4.0 prácu s veľkými číslami s triedou System.Numerics.BigInteger, avšak táto trieda obsahuje len základné operácie s týmito číslami.

Pre prácu s veľkými číslami je možné nájsť na Internete viac voľne stiahnuteľných tried, ktoré umožňujú definovanie takýchto veľkých čísiel. Niektoré z týchto tried majú výhodu oproti integrovanej triedy BigInteger tým, že okrem metód pre prácu s týmito číslami obsahujú aj niektoré v kryptográfii často používané funkcie, ktoré môžu byť užitočné pri implementácii digitálnych podpisových schém.

Pre vlastnú implementáciu dvoch podpisových schém v tejto práci bola použitá trieda BigInt z knižnice Emil.GMP, ktorá je voľne stiahnuteľná z webovej stránky autora [15].

Trieda umožňuje vytvorenie veľkého čísla napríklad z dátového typu string alebo bajtového pola:

```
public BigInt(string s);
public BigInt(byte[] value);
```

Okrem bežných matematických funkcii trieda obsahuje metódu pre výpočet modula a pre umocnenie čísla typu BigInt s jeho následným modulovaním:

```
public BigInt Mod(BigInt mod);
public BigInt PowerMod(BigInt exponent, BigInt mod);
```

Z kryptográfii často používaných funkcii trieda okrem iných umožňuje nájdenie najväčšieho spoločného deliteľa, testovanie, či je číslo štvorcové a testovanie prvočísla pomocou algoritmu Rabin-Miller, ktoré boli použité aj v implementácii:

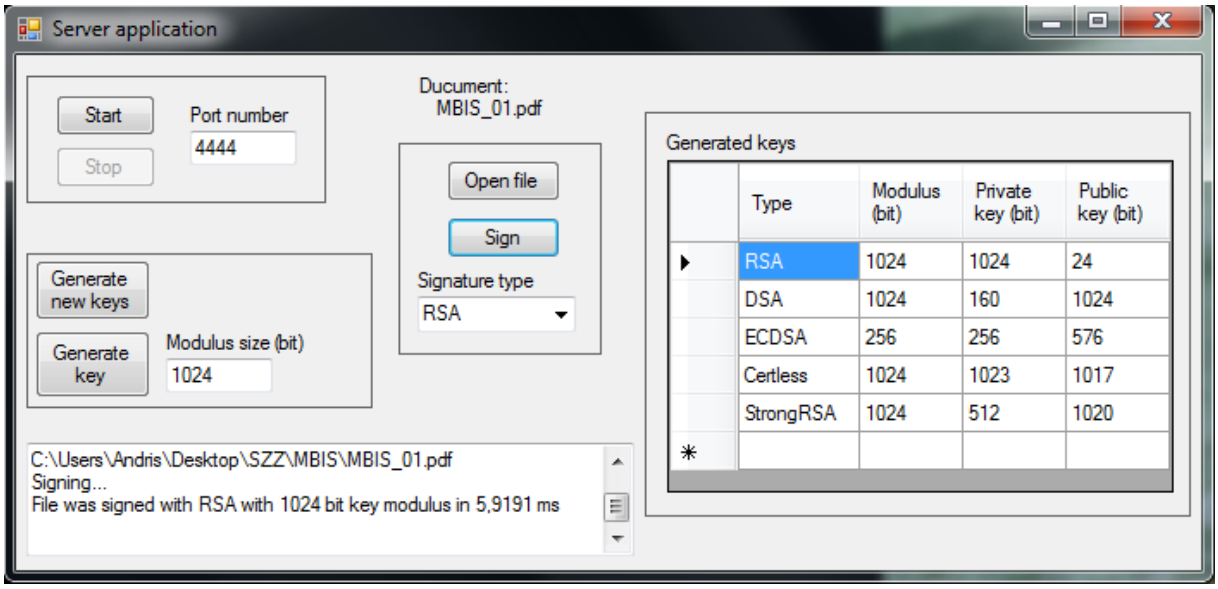
```
public static BigInt Gcd(BigInt x, BigInt y); //najväčší spoločný deliteľ
public bool IsPerfectSquare(); //test štvorcového čísla
public bool IsProbablyPrimeRabinMiller(int repetitions); //test prvočísla
```

4.5 Implementovaná aplikácia server

Vlastné riešenie digitálneho podpisovania dát je znázornené na sieťovej aplikácii klient-server. Server poskytuje klientom stiahnutie serverom podpísaných dokumentov rôzneho typu. Klient si môže overiť, či počas sieťového prenosu nedošlo k modifikovaniu obsahu stiahnutého dokumentu.

Stlačením tlačítka *Start* server začne počúvať a čakať na pripojenia klientov na nastavenom porte v poli *Port number*. Server spustí nové vlákno pre každého nového pripojeného klienta, a tým pádom je umožnené pripojenie viac klientov k serveri v tom istom

čase. Týmto tlačítkom aj automaticky vygeneruje kľúče pre všetky podporované digitálne podpisové schémy. Na zastavenie servera slúži tlačítko *Stop*, ktorého je možné znova spustiť tlačítkom *Start*.



Obr. 4.1: Grafické užívateľské rozhranie aplikácii server.

Aplikácia, v ktorej grafické užívateľské rozhranie je na obrázku 4.1, podporuje podpisové schémy RSA, DSA a ECDSA, ktoré obsahujú .NET Framework, a ďalšie dve, ktoré boli implementované v rámci tejto práce. V aplikácii nazývaný Certless je podpisová schéma bez certifikátu na báze identity užívateľa používajúca algoritmus ElGamal. Táto schéma je popísaná v časti 3.1. Druhá, vlastne implementovaná schéma je názyvaná StrongRSA, ktorá je založená na silnom RSA predpoklade a je popísaná v časti 3.4.

Pre automaticky vygenerované kľúče je použité 1024 bitové modulo u všetkých schém okrem ECDSA, kde sa použije 256 bitové modulo. Z dôvodu výpočtovej náročnosti generovanie kľúčov pre všetkých päť digitálnych podpisových schém nie sú použité väčšie modulá. Vygenerovať nové kľúčové páre je možné aj tlačítkom *Generate new keys*. Aj v tomto prípade sú použité také isté modula ako pri generovaní kľúčov pomocou tlačítka *Start*. Certless je schéma založená na báze identity, takže implicitne je použitá identita emailová adresa *user@email.com*.

Po vybraní schémy z položky *Signature type* a zadaní veľkosti modula v bitoch do pola *Modulus size* je možné vygenerovať nový kľúčový pár len pre vybranú digitálnu podpisovú schému pomocou tlačítka *Generate key*. V prípade Certless sa objaví položka *User*

ID, do ktorého sa zadáva identita použitá ako verejný kľúč. Veľkosti aktuálnych vygenerovaných kľúčov sa objavia v tabuľke *Generated keys*, v ktorej v jednotlivých schémach je uvedená veľkosť modula, súkromného a verejného kľúča v bitoch.

V aplikácii implementované schémy podporujú nasledujúce veľkosti modúl:

- **RSA**: od 384 do 16384 bitov po 8 bitoch.
- **DSA**: od 512 do 1024 bitov po 64 bitoch.
- **ECDSA**: 256 a 384 bitov.
- **Certless**: bez obmedzenia, ale deliteľné 8.
- **StrongRSA**: 384 a od 512 do 2048 bitov po 256 bitoch.

Pre podpísanie dokumentu treba najprv tlačítkom *Open file* vybrať súbor, resp. dokument. Aplikácia umožňuje vybratie a podpísanie akéhokoľvek typu dokumentu, ktorého názov sa po vybraní objaví v položke *Document* nad tlačítkom *Open file*, potom podpisová schéma sa vyberie zo zoznamu *Signature type*. Podpisovanie po stlačení tlačítka *Sign* prebieha tak, že dokument sa prevedie do podoby bajtového pola, čo je zhašované funkciou SHA-1, a výsledný 160 bitový haš je podpísaný pomocou vybranej digitálnej podpisovej schémy s aktuálnym súkromným kľúčom. Veľkosť podpisujúceho kľúča je uvedená v tabuľke *Generated keys* na pravej strane grafického užívateľského rozhrania.

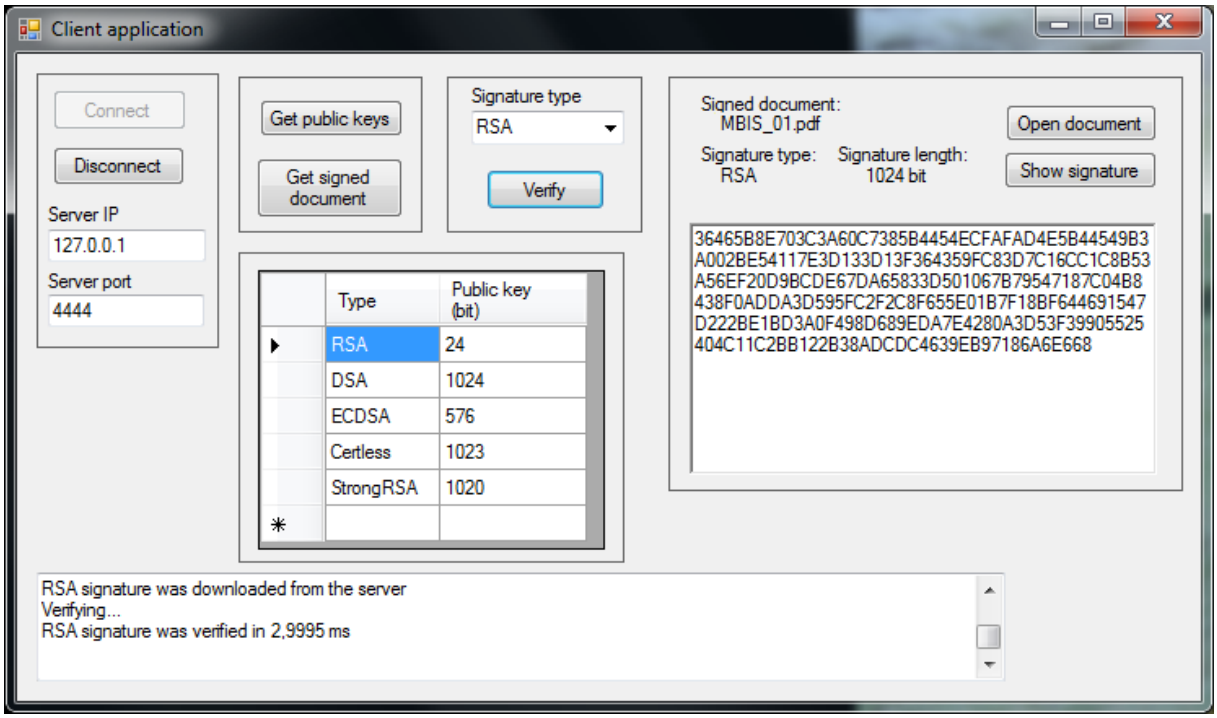
U všetkých schém je na hašovaniu dokumentu použitá funkcia SHA-1, lebo iba táto je podporovaná u všetkých implementovaných podpisových schém. Pri meraní času podpísania a overenia schém kvôli porovnaniu je dôležité aby sa podpísalo, resp. overilo haš rovnakou dĺžkou, čo je 160 bitov v prípade SHA-1.

4.6 Implementovaná aplikácia klient

Klientská aplikácia umožňuje stiahnutie digitálne podpísaných dokumentov zo servera a ich verifikáciu pravosti, a tým pádom je možné overiť, či behom sieťového prenosu neboli modifikované.

Po zadaní ip adresy servera do položky *Server IP* a portu do *Server port* sa klient môže pripojiť k serveru pomocou tlačítka *Connect*. O úspešnom pripojení je užívateľ informovaný v komunikačnom okne v dolnej časti grafického užívateľského rozhrania aplikácii. Tlačítko *Disconnect* je možné použiť na prípadne odpojenie od servera.

Po úspešnom pripojení užívateľ môže stiahnuť dostupné verejné kľúče zo servera tlačítkom *Get public keys*, ktoré po stiahnutí sa okamžite objavia v tabuľke verejných kľúčov. Pre stiahnutie serverom podpísaného dokumentu tlačítkom *Get signed document*, treba najprv zo zoznamu *Signature type* vybrať typ podpisu, ktorý užívateľ chce stiahnuť. Klient podporuje tie isté podpisové schémy, ktoré server používa k podpisovaniu. Po úspešnom stiahnutí podpísaného dokumentu sa objaví jeho názov s príponou v položke *Signed document*. Teraz už je tlačítko pre overenie platnosti podpisu *Verify* aktívne, a klient si môže overenie previesť. Výsledok a čas potrebný k overeniu je vypísaný v komunikačnom okne.



Obr. 4.2: Grafické užívateľské rozhranie aplikácii klient.

Aplikácia, ktorej grafické užívateľské rozhranie je vidieť na obrázku 4.2, tiež umožňuje ukázanie obsahu stiahnutého podpisu tlačítkom *Show signature*. Obsah sa objaví v okne na pravej strane grafického užívateľského rozhrania, a taktiež je vypísaný typ podpisu v položke *Signature type* a aj jeho dĺžka v *Signature length*. Obsah je vypísaný v prípade RSA, DSA a ECDSA v hexadecimálnej forme, a v prípade Certless a StrongRSA sa vypíšu veľkosti parametrov tvoriace digitálny podpis. Priamo z programu je možné otvoriť stiahnutý dokument tlačítkom *Open document*, ktorý sa otvorí v aplikácii priradenej danej prípony v operačnom systéme.

5 VLASTNOSTI IMPLEMENTOVANÝCH SCHÉM

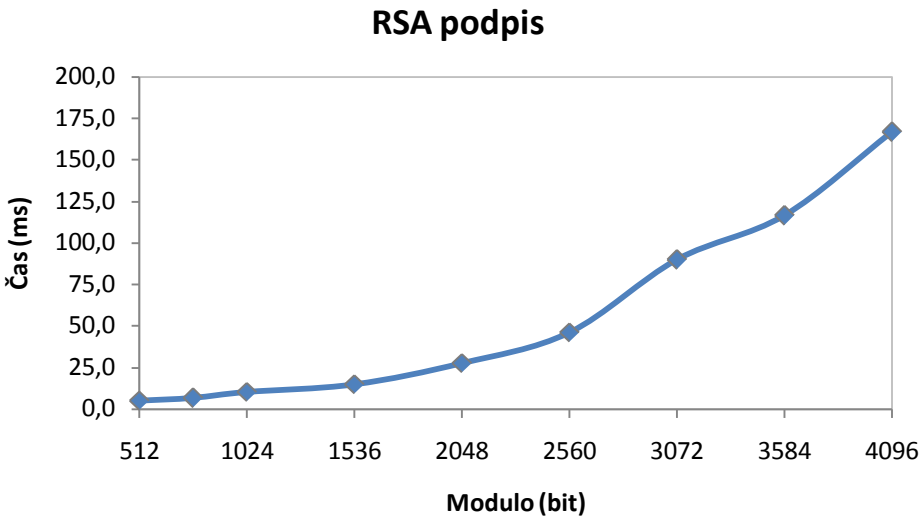
Vytvorené aplikácie umožňujú zistenia niektorých vlastnosti implementovaných digitálnych podpisových schém (RSA, StrongRSA, DSA, ECDSA a Certless), podľa ktorých je možné porovnať ich efektívnosť. Kým aplikáciou server je možné merať čas potrebný pre podpis 160 bitového haša implementovanými schémami, klient umožňuje meranie času potrebný pre overenie podpisu stiahnutého zo servera.

Knižnica Microsoft.Office obsahuje triedu na exportovanie hodnôt do aplikácie Microsoft Excel, čo bolo použité v tejto práci na ukladanie časov nameraných funkciami triedy System.Diagnostics.Stopwatch. Podpis a overenie boli namerané 100-krát za sebou na počítači s procesorom Intel Dual Core 1,86 GHz, a z nameraných časov bol vypočítaný medián. Časti 5.1 a 5.2 popisujú meranie podpisu a overenia a zobrazujú ich grafické závislosti. V časti 5.4 je vynesená tabuľka, v ktorej sú namerané hodnoty porovnané medzi sebou.

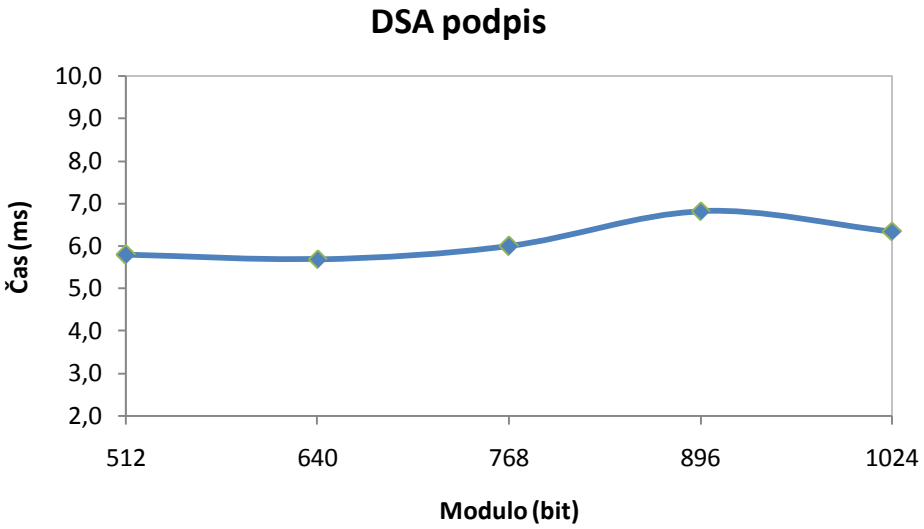
Aplikácia server obsahuje tabuľku znázorňujúcu veľkosti súkromných a verejných kľúčov jednotlivých podpisových schém. V tabuľke klienta sú vypísané veľkosti zo servera stiahnutých verejných kľúčov. Z pohľadu efektívnosti ďalšou dôležitou vlastnosťou je dĺžka podpisov, čo je možné overiť v aplikácii klient po jeho stiahnutím.

5.1 Meranie času podpisu

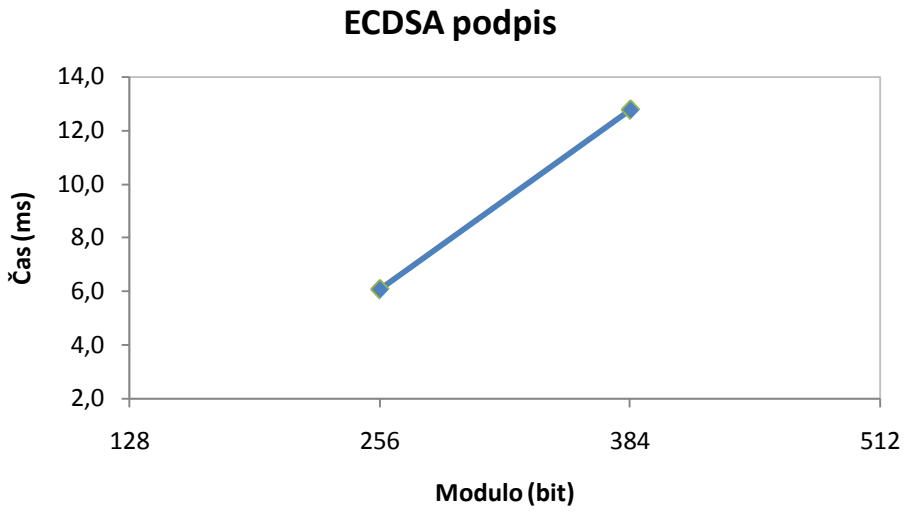
Namerané časy potrebné na podpis 160 bitového haša vytvorené hašovaciou funkciou SHA-1 sú na obrázkoch od 5.1 do 5.5. RSA, ECDSA a Certless používajú na podpisovanie rovnaký veľký súkromný kľúč ako je dĺžka použitého modula a u Strong RSA je veľkosť polovičná. So zvýšením veľkosti modula u DSA je čas potrebný na podpisovanie konštantný, lebo táto schéma vždy používa 160 bitový súkromný kľúč.



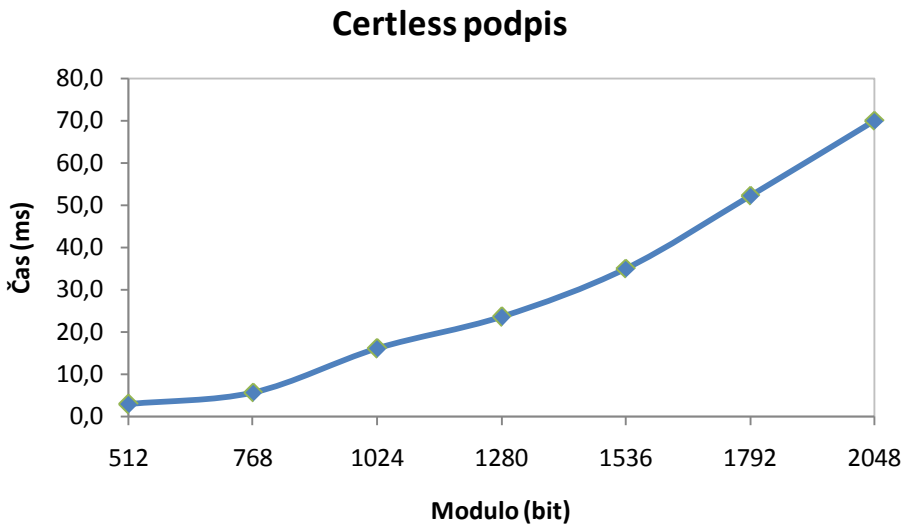
Obr. 5.1: Závislosť času potrebného na podpis 160 bitového haša na veľkosti použitého modula schémou RSA.



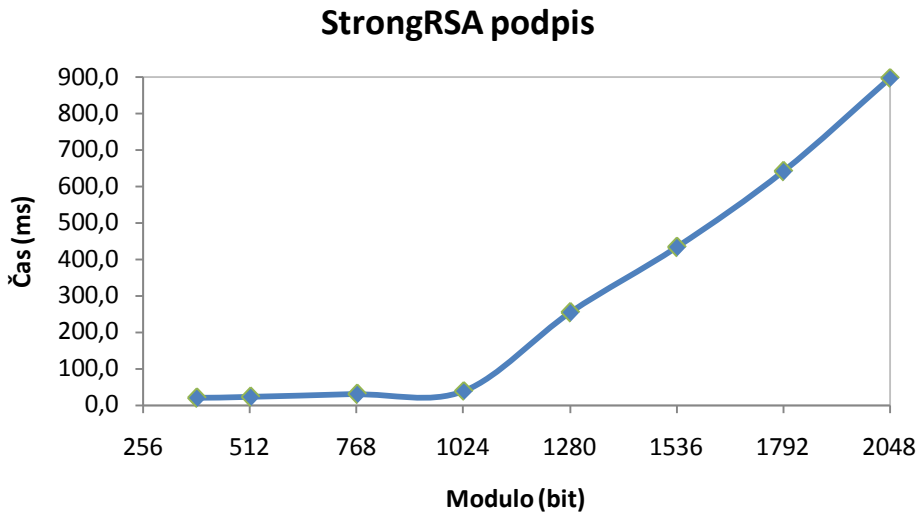
Obr. 5.2: Závislosť času potrebného na podpis 160 bitového haša na veľkosti použitého modula schémou DSA.



Obr. 5.3: Závislosť času potrebného na podpis 160 bitového haša na veľkosti použitého modula schémou ECDSA.



Obr. 5.4: Závislosť času potrebného na podpis 160 bitového haša na veľkosti použitého modula schémou Cerless.

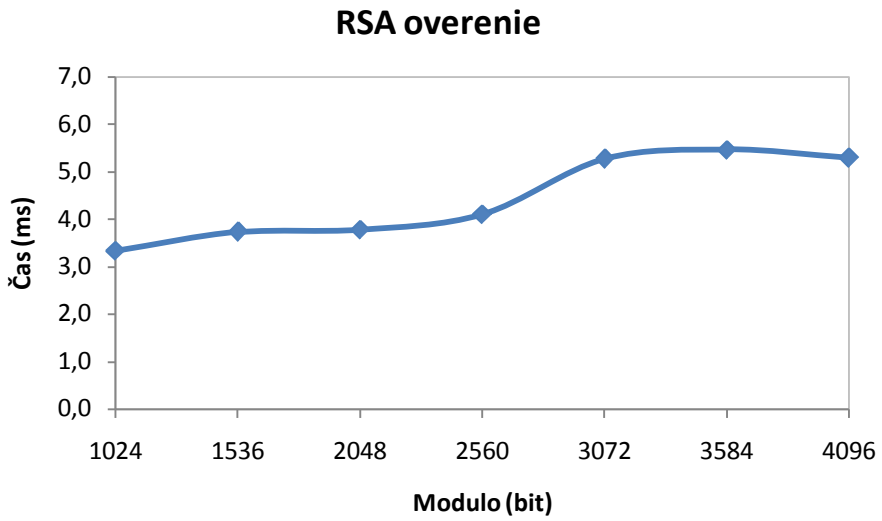


Obr. 5.5: Závislosť času potrebného na podpis 160 bitového haša na veľkosti použitého modula schémou Strong RSA.

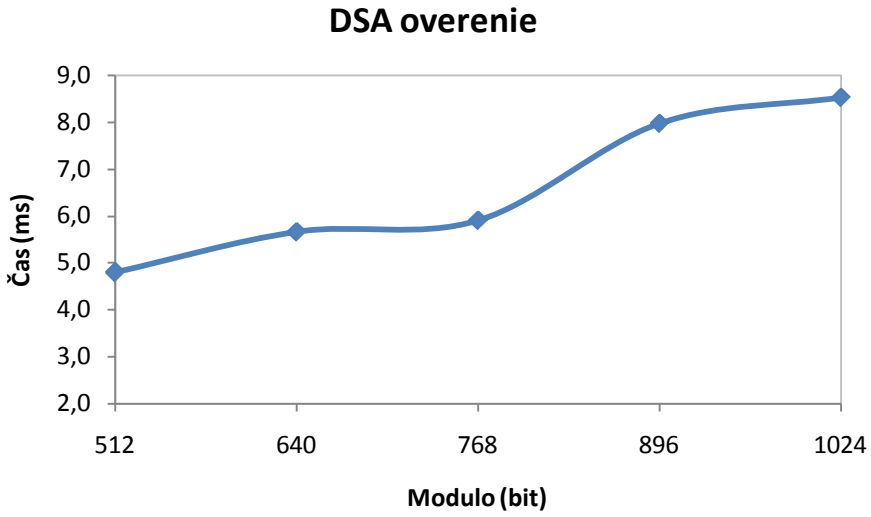
Podpis prebieha najrýchlejšie u schém DSA a ECDSA, ale v prípade ECDSA použité kľúče zaisťujú vyššiu bezpečnosť. Tretím najrýchlejším v poradí je schéma RSA, a v prípade Certless je podpisovanie výpočtovo trošku náročnejšie. Najpomalším z implementovaných schém je Strong RSA, ktorý v prípade 2048 bitového modula podpisuje oveľa pomalšie ako ostatné schémy.

5.2 Meranie času overenia

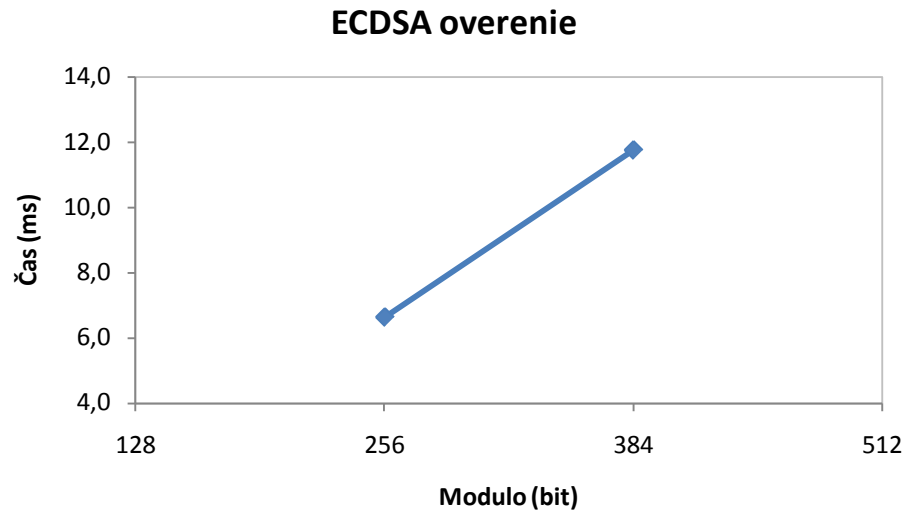
Namerané časy potrebné na overenie podpisov u jednotlivých schém sú znázornené na obrázkoch od 5.6 do 5.10. DSA, Certless a Strong RSA používajú rovnako dlhý verejný kľúč ako je veľkosť ich modula, kým ECDSA o niečo väčší ako dvojnásobok jeho modula. Najkratší kľúč je v prípade schémy RSA, kde verejný kľúč má voliteľnú veľkosť, a v tejto implementácii bol použitý 24 bitový.



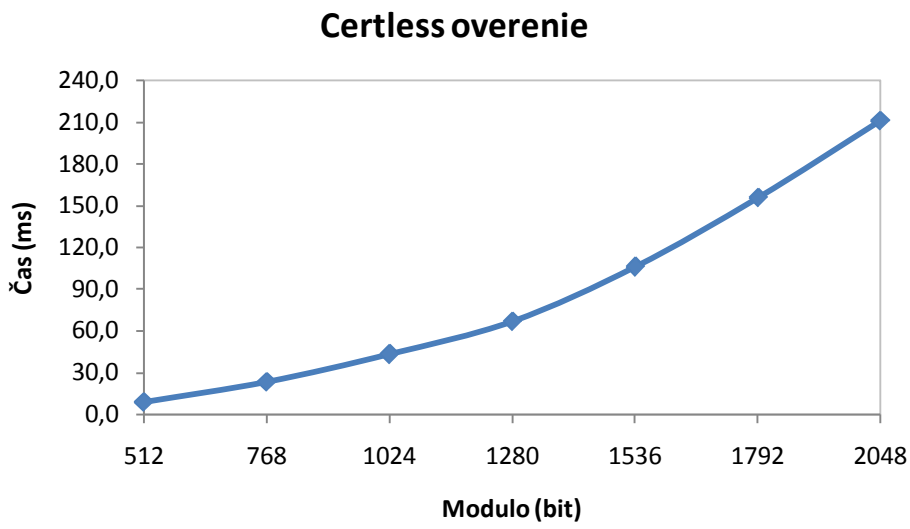
Obr. 5.6: Závislosť času potrebného na overenie RSA podpisu na veľkosti použitého modula.



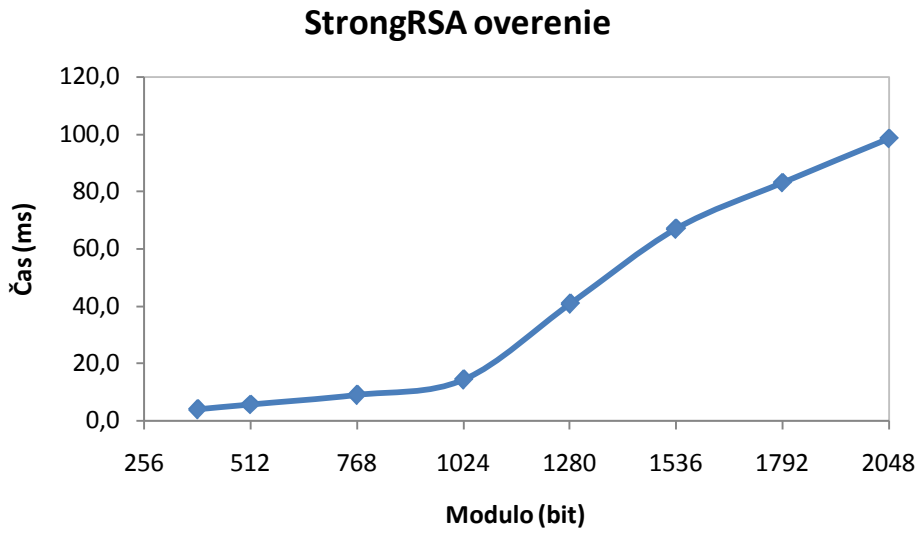
Obr. 5.7: Závislosť času potrebného na overenie DSA podpisu na veľkosti použitého modula.



Obr. 5.8: Závislosť času potrebného na overenie ECDSA podpisu na veľkosti použitého modula.



Obr. 5.9: Závislosť času potrebného na overenie Certless podpisu na veľkosti použitého modula.



Obr. 5.10: Závislosť času potrebného na overenie Strong RSA podpisu na veľkosti použitého modula.

Najrýchlejšie overenie prebieha v prípade RSA, kde čas potrebný na overenie podpisu nezávisí na použitej dĺžky modula (voliteľne zvolená veľkosť verejného kľúča). Podobné hodnoty boli namerané u DSA a ECDSA, avšak u meraných modúl ECDSA je zaistená vyššia bezpečnosť ako v ostatných schémach. Časovo najnáročnejšie overenie z implementovaných schém majú Strong RSA a Certless.

5.3 Dĺžka podpisu

Dĺžky podpisov je možné vypísať na strane klienta po jeho stiahnutí. Najkratší podpis pomedzi implementovaných schém má DSA, kde je konštantne 320 bitov. Podpisy sú krátke aj u ECDSA, rovnajú sa s dvojnásobkom dĺžky modula, ale u tejto schémy boli použité výrazne menšie modulá oproti ostatným. V prípade RSA a Certless dĺžka podpisu sa rovná veľkosti použitého modula. Strong RSA vytvára do 1024 bitov rovnako dlhý podpis, ako je jeho modulo, ale už u vyšších hodnotách rastie na jeho dvojnásobok.

5.4 Zhrnutie nameraných výsledkov

Porovnanie vlastnosti jednotlivých implementovaných schém je znázornené v tabuľkách 5.1 a 5.2.

V prvej tabuľke použité modulá zodpovedajú stupňom bezpečnosti považovaných dnes za bezpečné, a čas potrebný pre ich prelomenie je 10^{12} MIPS rokov [11]. Avšak niektoré odporučenia tieto dĺžky už nepovažujú za bezpečné pre dlhodobé podpisy, ale 1024 bitové boli vybrané kvôli tomu, že väčšina schém túto hodnotu modula podporovala. V prípade ECDSA nebolo možné zvoliť 160 bitové modulo, čo by zodpovedalo týmto bezpečnostným stupňom, lebo implementovaná schéma podporuje iba 256 a 384 bitové. V ECDSA 256 bitové modulo zaisťuje bezpečnosť na úrovni ako približne 2560 bitové u problémoch IFP a DLP.

Problém	Schéma	Modulo (bit)	Podpis (ms)	Overenie (ms)	Dĺžka podpisu (bit)
IFP	RSA	1024	10,6	3,3	1024
	Strong RSA	1024	39,2	14,4	1022
DLP	DSA	1024	6,3	8,5	320
	Cerless	1024	16,1	43,1	1024
ECDLP	ECDSA	256	6,1	6,6	512

Tab. 5.1: Zrovnanie vlastností implementovaných schém (čas potrebný pre ich prelomenie je 10^{12} MIPS rokov).

Druhá tabuľka obsahuje schémy s modulami, ktoré zabezpečujú vyššiu stupeň bezpečnosti ako v predchádzajúcom prípade, a čas potrebný pre ich prelomenie je 10^{20} MIPS rokov. Implementovaná schéma DSA nepodporuje väčšie modulo ako 1024 bitov, a preto bola táto hodnota použitá aj v tejto tabuľke.

Problém	Schéma	Modulo (bit)	Podpis (ms)	Overenie (ms)	Dĺžka podpisu (bit)
IFP	RSA	2048	27,7	3,8	2048
	Strong RSA	2048	899	98,8	4095
DLP	DSA	1024	6,3	8,5	320
	Cerless	2048	70,2	211,5	2048
ECDLP	ECDSA	256	6,1	6,6	512

Tab. 5.2: Zrovnanie vlastností implementovaných schém (čas potrebný pre ich prelomenie je 10^{20} MIPS rokov).

5.5 Vhodnosť digitálneho podpisu pre výpočtovo slabé zariadenie

Vo výpočtovo slabých zariadeniach, ako sú napr. smartkarty a senzorové siete, je veľmi dôležité, aby nasadená digitálna podpisová schéma mala veľmi efektívne podpísanie, overenie alebo v niektorých prípadoch obidve. Vzhľadom na obmedzené pamäťové možnosti, ďalšou

dôležitou vlastnosťou je dĺžka vytvoreného podpisu. Čím kratšie podpisy vytvorí daná schéma, tým viac podpisu sa dá uložiť do pamäte zariadenia.

Z implementácie potvrdili teoretické predpoklady z časti 3.5. DSA má rýchle podpisovanie, naopak RSA rýchle overovanie. ECDSA je efektívnejšie ako ostatné dve schémy z toho hľadiska, že u nej je rýchlosť podpisu na úrovni DSA a rýchlosť overenia na úrovni RSA.

Najkratšie podpisy z implementovaných schém vytvára DSA, u ktorého podpis má konštante 320 bitov. Krátke podpisy vytvára aj ECDSA, kde dĺžka podpisu sa rovná dvojnásobku použitého modula.

Z výsledkov nameraných hodnôt vyplýva, že vlastne implementované schémy, Certless a StrongRSA, v rýchlosti podpisu a overeniu v porovnaní s ostatnými tromi mierne zaostávajú, ale sú odolné voči najnebezpečnejším útokom. Dĺžky vytvorených podpisov sú na úrovni RSA.

Certless je schéma bez certifikátu. Táto vlastnosť môže byť výhodou v systémoch s problematickou predistribúciou autentizovaných kľúčov (napr. bezdrôtové siete). Táto podpisová schéma bola navrhnutá tak, aby z ktorejkoľvek schémy typu ElGamal bolo možné vytvoriť schému bez certifikátu. V tejto prípade bola implementovaná práve ElGamal, ale použitím efektívnejšej schémy, ako napr. DSA, by bolo možné dosiahnuť zvýšení rýchlosť v podpisu a overení a skrátiť podpisy na 320 bitové.

Z tabuľky 5.2 je zrejmé, že najvhodnejšie pre výpočtovo slabé zariadenia z pisatich implementovaných a porovnaných schém je ECDSA.

6 ZÁVER

Hlavným cieľom tejto práce bolo popísať digitálne podpisy a výhody návrhov nových efektívnych podpisových schém oproti súčasným. V praktickej časti previesť implementáciu niektorých efektívnych schém vo vybranom programovacom jazyku, a pomocou aplikácii porovnať a analyzovať ich vlastnosti.

Z teoretického rozboru súčasných digitálnych podpisových schém bolo zistené, že dnes najpoužívanejšie podpisové schémy, RSA a DSA, majú také vlastnosti, ktoré môžu byť menej perspektívne v budúcnosti. Schéma RSA pri jeho uvedení používala hašovaciou funkciu MD5, ktorá časom bola zmenená na bezpečnejšiu SHA-1. Aj DSA používa SHA-1, čo v súčasnosti prestáva byť odporúčanou hašovaciou funkciou pre zaistenie dlhodobej bezpečnosti. Postupom času by mala byť nahradená bezpečnejšou SHA-2.

Výsledky analýzy poukazujú nato, že v súčasnosti najrozšírenejšie podpisové schémy RSA a DSA s vysokým tempom zlepšovania technológií počítačovej techniky potrebujú použiť stále väčšie kľúče. U RSA s rastúcou dĺžkou kľúčov rastie bezpečnosť pomaly oproti výpočtovým nárokom, ktoré rastie oveľa rýchlejšie. Tieto fakty vedú k výraznému zníženiu efektívnosti týchto schém a v prípade RSA aj k vytváraniu dlhých podpisov, čo je veľkou nevýhodou v systémoch s obmedzenou pamäťou. Efektívnosť týchto schém sa dá zvýšiť s použitím eliptických kriviek ako je to v prípade už štandardizovanej digitálnej podpisovej schémy ECDSA.

V budúcnosti možno očakávať rozšírenie bezdrôtových senzorových sietí, v ktorých je problematickou otázkou bezpečná distribúcia šifrovacích kľúčov medzi jednotlivými uzlami, a tiež je problém zaistiť autentičnosť a elektronický podpis dát. Súčasné digitálne podpisové schémy v týchto sieťach nie sú vhodné kvôli veľkým pamäťovým nárokom.

Súčasný trend na zvýšenie efektívnosti sa uberá k teoretickým návrhom podpisových schém s využitím bilineárneho párovania. Pomocou kryptosystémov postavených na základe identity užívateľa možno vytvárať digitálne podpisy bez nutnosti certifikátov, ktoré vyriešia problém s autentizáciou. Kombinovaním technik bilineárneho párovania alebo eliptických kriviek s digitálnym podpisom bez certifikátu je možné vytvoriť efektívne digitálne podpisové schémy, ktoré vytvárajú veľmi krátke podpisy. Zatiaľ sa objavujú prvé návrhy a implementácie takýchto digitálnych podpisových schém.

Praktická časť predstavuje vlastnú implementáciu (v prostrediu .NET v jazyku C#) dvoch efektívnych podpisových schém v rámci aplikácii klient-server. Implementácia prináša dve funkčné aplikácie: SigningServer (podpisujúci server) a ClientApp (klient), ktoré spolu komunikujú. Aplikácia SigningServer umožňuje digitálne podpísanie dokumentu akéhokoľvek typu piatimi podpisovými schémami (RSA, DSA, ECDSA, Certless a StrongRSA). Posledné dve sú vlastné implementácie podľa návrhov z kapitoly 3. Certless je schéma bez certifikátu na báze diskrétneho logaritmu a StrongRSA založená na silnom RSA predpoklade. Aplikácia ClientApp umožňuje stiahnutie podpísaného dokumentu zo servera, a overenie jeho pravosti. Pomocou týchto dvoch aplikácii boli otestované niektoré vlastnosti (rýchlosť podpisu, rýchlosť overenia a dĺžka podpisu) implementovaných schém, ktoré sú zhrnuté a analyzované v kapitole 5. Z výsledkov vyplýva, že Certless a StrongRSA v meraných hodnotách v porovnaní s ostatnými tromi mierne zaostávajú, ale sú odolné voči najnebezpečnejším útokom. Kým prvá menovaná schéma je bez certifikátu a môže byť použitá v systémoch bez autentizácie, druhá má výhodu v tom, že je bezstavová. Z piatich implementovaných a porovnaných schém ECDSA malo najkratší čas pre podpis i overenie.

LITERATÚRA

[1] BOYAR J., CHAUM, D., DAMGARD, I.B., PEDERSEN, T. Convertible undeniable signatures. *Advances in Cryptology - CRYPTO '90 (LNCS 537)*, 189-205, 1991.

[2] BURDA, K. *Bezpečnost informačných systémů*. Brno : Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2005. 104 s.

[3] CRAMER, R.; SHOUP, V. *Signature Schemes Based on the Strong RSA Assumption* [online]. 2000, [cit. 2011-03-12]. Dostupné z WWW: <http://www.zurich.ibm.com/security/ace/sig.pdf>.

[4] DENG, J.; CHENG, X.; GUI, Q. *Design of Hyper Elliptic Curve Digital Signature*. 2009, [cit. 2010-11-18]. Dostupné z WWW: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5190178>.

[5] DOSTÁLEK, L.; VOHNOUTOVÁ, M. *Velký průvodce infrastrukturou PKI*. Computer Press, Brno 2006, ISBN: 80-251-0828-7.

[6] DU, H.; WEN, Q. Efficient and provably-secure certificateless short signature scheme from bilinear pairings. *Computer Standards & Interfaces*, 31:390-394, 2009, [cit. 2010-11-12]. Dostupné z WWW: <http://portal.acm.org/citation.cfm?id=1461086>.

[7] DU, W., DENG, J., HAN, Y., VARSHNEY, P. A *Pairwise Key Predistribution Scheme for Wireless Sensor Networks*. 2003 [cit. 2010-12-11]. Dostupné z WWW: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.1115&rep=rep1&type=pdf>.

[8] ELGAMAL, T. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, July 1985.

[9] HARN, L., REN, J., LIN. Ch. Design of DL-based certificateless digital signatures. *The Journal of Systems and Software*, 82:789-793, 2009.

[10] CHAUM, D. Blind signatures for untraceable payments. *Advances in Cryptology-Proceedings of Crypto 82*, 199-203, 1983.

[11] MENEZES, A., VAN OORSCHOT, P., VANSTONE, S. *Handbook of applied cryptography*. Boca Raton : CRC Press, 1997. 780 s. ISBN 0849385237.

[12] Microsoft Developer Network [online]. 2011 [cit. 5.5.2011]. Dostupné z WWW: < http://msdn.microsoft.com/>.

[13] OKAMOTO, T., SHIRAISHI, A. A fast signature scheme based on quadratic inequalities. *Proceedings of the 1985 IEEE Symposium on Security and Privacy*, 123-132, 1985.

[14] PŘIBYL, J. *Ochrana dat v informatice*. Vydavatelství ČVUT, Praha 1993.

[15] Rozšírenie aplikácie *Emil.GMP.dll* pre prácu s veľkými číslami. Dostupné z WWW: <http://www.emilstefanov.net/Projects/GnuMpDotNet/>.

[16] RSA Laboratories [online]. 2010 [cit. 2010-11-22]. Dostupné z WWW: <http://www.rsa.com/rsalabs>.

[17] SCHNORR, C. Efficient Signatures for Smart Card. *Journal of Cryptology*, No. 3, 1991.

[18] STALLINGS, W. *Cryptography and Network Security*. 4th edition. [s.l.] : [s.n.], 2006. 592 s. ISBN 0131873164.

[19] VAN HEIJST, E., PEDERSEN, T.P., PFITZMANN, B. New constructions of failstop signatures and lower bounds. *Advances in Cryptology - CRYPTO '92 (LNCS 740)*, 15-30, 1993.

ZOZNAM POUŽITÝCH SKRATIEK

AES	Advanced Encryption Standard
asym. kryp.	Asymetrický kryptosystém
CA	Certification Authority
CLR	Common Language Runtime
CLS	Certificateless Signature
CRL	Certificate Revocation List
CRT	Chinese Remainder Theorem
DER	Disinguished Encoding Rules
DL	Discrete Logarithm
DLL	Dynamic-Link Library
DLP	Discrete Logarithm Problem
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ECDLP	Elliptic Curve Discrete Logarithmic Problem
ECDSA	Elliptic Curve Digital Signature Algortihm
ECDsaCng	Elliptic Curve Digital Signature Algorithm Cryptography Next Generation
ESIGN	Efficient Digital Signature
ext. Euc. alg.	Extended Euclidean algorithm
HEC	Hyperelliptic curve
IBS	Identity Based Signature
ID	Identity
IFP	Integer Factorization Problem
Inv-CDHP	Inverse Computational Diffie-Hellman Problem
KL	Key Legitimacy
MAC	Message Authentication Code
MIL	Microsoft Intermediate Language
MIPS	Million Instruction Per Second

mod. exp.	Modular exponentiation
mod. inv.	Modular inversion
mod. mul.	Modular multiplication
MSRP	Modular Square Root Problem
OT	Owner Trust
PEM P	Privacy Enhanced Mail
PGP	Pretty Good Privacy
PKCS	Public Key Cryptography Standards
PKG	Public Key Generator
PKI	Public Key Infrastructure
RA	Registration Authority
SK	Súkromný Kľúč
SSL	Secure Sockets Layer
ST	Signature Trust
sym. kryp.	Symetrický kryptosystém
TTP	Trusted Third Party
VK	Verejný Kľúč

ZOZNAM PRÍLOH

Príloha A: Obsah priloženého DVD

Zložka s programy: Applications\

Pokyny k obsluhe aplikácií:
pokyny.txt

Spustenie implementacie servera (Solution) v MS Visual Studio:
SigningServer.sln

Spustenie implementacie klienta (Solution) v MS Visual Studio:
ClientApp.sln

Zložka aplikácie server:
ServerApp\
spustenie aplikácie: bin\Debug\ verifycationServer.exe

Zložka aplikácie klient:
ClientApp\
spustenie aplikácie: bin\Debug\ ClientApplication.exe