Nepopiratelnost digitálních podpisů

Tomáš Rosa

eBanka, a.s.
Na Příkopě 19, 117 19 Praha 1, e-mail: trosa@ebanka.cz

Klíčová slova: nepopiratelnost, digitální podpis, elektronický podpis, kolize, RSA, DSA, ECDSA.

Abstrakt. Příspěvek se věnuje problematice digitálního podpisu jakožto základnímu stavebnímu prvku služeb nepopiratelnosti podle norem ISO/IEC 13888 a ISO/IEC 10181-4. Zatímco otázkám konstrukce robustních důkazů o nepopiratelnosti vybraných událostí je v uvedených normách věnována relativně velká pozornost, tak otázce nepopiratelnosti samotného dílčího aktu *provedení operace podpisu* v použitém podpisovém schématu už tak silná pozornost věnována není. Ještě menší pozornost pak bývá věnována tomu, jestli charakteristiky, které se pod pojmem nepopiratelnost studují v oblasti kryptologie, jsou právě tím souborem vlastností digitálních podpisů, který je bezpečnostními architekty implicitně předpokládán ve standardech uvedeného typu. Naším cílem je představit zde vybrané kryptologické aspekty digitálních podpisů, u kterých je reálný předpoklad úzké souvislosti s konstrukcí služeb nepopiratelnosti vyšší úrovně (nepopiratelnost odeslání či doručení zprávy, atp.).

1. Služba nepopiratelnosti

Představme si soudní proces, v němž se určitý subjekt snaží zprostit závazků vyplývajících z nějaké smlouvy. V obecnějším případě můžeme uvažovat o tom, že se daný subjekt snaží být zproštěn důsledků toho, že se v minulosti stala či naopak nestala nějaká událost (například subjekt ukradl citlivá data, neprovedl včas nějakou akci, atp.). Základním nástrojem, který má takový subjekt k dispozici, je pomocí logických důkazů přesvědčit soud, že domnělá událost se ve skutečnosti vůbec nestala, případně že se odehrála jinak, než soud předpokládá. Pokud se mu podaří takové důkazy shromáždit, může doufat v příznivý výsledek celého procesu. Zdůrazněme ovšem, že v prvé řadě soud musí uvěřit v logickou platnost přednesených důkazů a až poté se bude pravděpodobně zabývat jejich důsledky pro samotný proces. Se zajišťováním a ověřováním důkazů v „běžném“ světě už mají příslušné orgány tisícileté zkušenosti, takže toto téma obvykle nebývá nijak zvlášť zdůrazňováno. Ovšem v případě, že takové zajišťování probíhá v prostředí moderních informačních systémů, je radno se mít na pozoru. Bohužel málokterý informační systém je totiž dnes postaven tak, aby se v něm proces zajišťování a ověřování důkazů nehrozil zvrhnout v gejzír justičních trapasů, kdy byv sotva předložen, je příslušný důkaz okamžitě roztrhán znalci příslušného oboru.

Mechanizmus, jehož správná implementace v daném informačním systému by měla umožnit mimo jiné důvěryhodné vytváření a ověřování důkazů, se nazývá *služba nepopiratelnosti*. Definice uvedená v normě [1], která se touto problematikou podrobně zabývá, říká: *Cílem služby nepopiratelnosti je vytvářet, shromažďovat, udržovat, zajistit dostupnost a ověřovat důkazy týkající se údajné události nebo činnosti, aby bylo možné řešit spory o tom, zda se událost nebo činnost vyskytla či nikoliv.* Nepopiratelnost je též považována za jednu ze čtyř základních služeb kryptografických schémat [4]. I přes celé toto snažení však musíme konstatovat, že penetrace tohoto fenoménu do běžných informačních systémů je zatím mizivá. Na „vině“ je v tomto případě nejspíš absence tlaku ze strany uživatelů informačních systémů, kteří si patrně z jakéhosi respektu vůči elektronickým systémům zatím příliš netroufají zpochybňovat důkazy založené na „tvrzeních strojů“. Toto přechodné období klidu však sotva vydrží věčně a je zde reálná hrozba, že první vlna útoků na nepopiratelnost elektronických důkazů zastihne informační společnost krajně nepřipravenou a to hned ze dvou důvodů: Prvním je slabý zájem současných bezpečnostních architektů o službu nepopiratelnosti jako takovou, druhým je pak jistá nevyzrálost služby samotné. Například způsobu, jakým je nepopiratelnost prezentována v pramenech [1] a [2], zjevně chybí hlubší propojení se současným poznáním v oblasti kryptologie. Na druhé straně kryptologům zase často chybí jasné vodítko, které by jim pomohlo rozlišovat, jaké vlastnosti kryptografických schémat jsou pro službu nepopiratelnosti zásadní a jaké mají jen marginální význam.

Jedním ze základních nástrojů uvažovaných v [1] a [2] pro konstrukci služby nepopiratelnosti je digitální podpis [4]. To je jistě velmi racionální krok, avšak jeho zapracování do citovaných pramenů trpí určitým nedostatkem a to, že při vytváření a ověřování důkazů je věnována poměrně malá pozornost problematice nepopiratelnosti samotného *aktu vytvoření digitálního podpisu*. Cílem tohoto příspěvku je upozornit na problémy, které zde z kryptologického hlediska mohou nastat, a nastínit možné způsoby jejich řešení. Technické útoky založené na přímé kompromitaci privátního klíče, falešných údajích v certifikátu veřejného klíče, substituci veřejného klíče certifikační autority apod. jsou v praxi poměrně dobře známy a je jim věnován určitý prostor i v [2]. Jimi se zde proto podrobněji zabývat nebudeme.

Před dalším výkladem upozorněme na nutnost pečlivého rozlišování pojmů *digitální* a *elektronický* podpis [12]. Připomeňme, že první z nich je pojem kryptologický, zatímco druhý je pojem v podstatě legislativní. Na první pohled se možná zdá být smysluplnější, aby se prameny [1] a [2] odvolávaly na kryptologii nikoliv přímo, ale právě prostřednictvím norem typu [12], avšak při hlubším promyšlení této konstrukce je zřejmé, že tento krok je sotva schůdný. Základní překážkou je zde různorodost obou pramenů (mezinárodní norma vs. lokální legislativní pramen). Zařazení norem [1] a [2] je tak ve vztahu k [12] lépe spatřovat na úrovni technické vyhlášky [10], kde jejich společným cílem může být naplnění proklamačních požadavků stanovených pro konkrétní druhy elektronického podpisu zákonem [12]. Dodejme, že pojem nepopiratelnost se v [12] sice nediskutuje přímo, avšak je touto normou, coby služba spojená s určitým druhem elektronického podpisu, striktně implicitně předpokládána. Proto vše, co je v tomto příspěvku uvedeno, se též automaticky úzce dotýká oblasti elektronického podepisování. Tento fakt budeme dále v textu považovat již za zřejmý a nebudeme jej zvlášť zdůrazňovat.

2. Útoky na nepopiratelnost digitálních podpisů

Označme *Priv* a *Pub* soukromý, respektive veřejný klíč uživatele (signatáře), tvořící klíčový pár (*Pub*, *Priv*). Dále zaveďme podepisovací a ověřovací algoritmus v podpisovém schématu s dodatkem jako Sig_{Priv}: $\boldsymbol{M} \rightarrow \boldsymbol{S}$: $m \rightarrow s$, respektive Ver_{Pub}: $\boldsymbol{M}$ x $\boldsymbol{S} \rightarrow \boldsymbol{RES}$: $(m, s) \rightarrow v$, kde $\boldsymbol{M}$ je množina zpráv, $\boldsymbol{S}$ množina podpisových řetězců, $\boldsymbol{RES}$ = {ANO, NE} a v = ANO znamená, že podpis s zprávy m je algoritmem Ver_{Pub} uznán jako platný [4]. Jádro problematiky (ne)popiratelnosti digitálních podpisů spočívá ve vztahu mezi algoritmy *Sig* a *Ver*, přesněji řečeno v tom, co zatím umíme o tomto vztahu matematicky dokázat. Ukázat, že podpis vytvořený algoritmem *Sig* bude později operací *Ver* uznán jako platný, je obvykle triviální záležitost. Tento typ důkazu však pro dosažení nepopiratelnosti rozhodně nestačí. Zde totiž potřebujeme prokázat, že pokud $Ver_{Pub}(m, s)$ = ANO, potom s muselo nutně vzniknout aplikací algoritmu Sig_{Priv} jako $Sig_{Priv}(m) = s$. Zde jsme ovšem u všech prakticky používaných podpisových schémat (RSA, (EC)DSA, ElGamal [4]) nuceni používat důkazy krajně heuristické povahy. Jednoduše řečeno musíme spoléhat na to, že u daného schématu dosud není znám jiný způsob, kterým by bylo možné diskutovanou hodnotu s získat, než je vzorná aplikace podepisovacího algoritmu dle uvedeného předpisu. Objevení takového alternativního způsobu pak pochopitelně přímo a podstatně oslabuje nepopiratelnost podpisů vytvořených daným schématem. Říkáme, že v takovém případě je dotyčný subjekt schopen u soudu předložit *alternativní vysvětlení*, kterým vyvrací původní domněnku, že vědomě opatřil předloženou zprávu svým digitálním podpisem. Primárním cílem kryptologů je zamezit všem známým konstrukcím takových vysvětlení a tím dané podpisové schéma učinit právně důvěryhodným. V následujících příkladech uvidíme, že to není vždy triviální úkol a že ho dosud v mnoha případech nelze považovat za zcela vyřešený.

2.1 Kolize hašovacích funkcí

Vliv vlastností hašovacích funkcí (přehled viz [4]) na nepopiratelnost digitálních podpisů je v kryptologii znám velmi dobře a patrně proto je i jedním z nejčastěji studovaných témat ve vztahu k nepopiratelnosti vůbec. Drtivá většina podpisových schémat totiž nezpracovává podepisované či ověřované zprávy přímo, nýbrž v podobě jejich otisku $h(m)$, kde m je příslušná zpráva a h zvolená hašovací funkce. Jedním z hlavních důvodů pro toto uspořádání je, že výstup hašovací funkce má vždy pevnou délku (například MD5 má 128 b, SHA-1 nabízí 160 b), zatímco jejich vstup může mít délku prakticky libovolnou. Tím vzniká jistá normalizace délky zpracovávaných zpráv, což výrazně usnadňuje návrh celého podpisového schématu. Z pohledu nepopiratelnosti ovšem musíme toto zjednodušení kompenzovat řadou netriviálních požadavků na použitou hašovací funkci. Nejdůležitějším z nich je *bezkoliznost*: Musí být výpočetně neschůdné najít dvě různé zprávy m_1, m_2 spolu s hodnotou $z \in Im(h)$ takové, aby $h(m_1) = h(m_2) = z$. Pokud by funkce nebyla bezkolizní, pak by útočník na nepopiratelnost mohl snadno založit své *alternativní vysvětlení* na argumentu, že předložený podpis ve skutečnosti náleží ke zprávě m_2, nikoliv k m_1 (či obráceně). Útočníkem by zde přitom mohl být jak sám signatář, který si před podpisem nějaké riskantní smlouvy m_1 dopředu připraví zadní vrátka v podobě nějakého „neškodného“ dokumentu m_2, tak i jiná osoba, která na signatáři pod záminkou podpisu „neškodného“ dokumentu m_2 vyláká podpis smlouvy m_1.

O tom, že kolize hašovacích funkcí jsou reálnou hrozbou, svědčí například závažné prolomení funkce MD4, kterým Dobbertin na sklonku roku 1995 ukončil jinak slibnou kariéru této funkce [4]. Ani její následovník MD5 přitom nezůstává bez poskvrny, ačkoliv zde se ještě nepovedlo spojit určité dílčí výsledky do útoku na celou hašovací funkci[1]. Rovněž je důležité upozornit, že bezpečnost hašovací funkce nemusí zasáhnout jen náhle objevená konstrukční slabina. Podobně jako u jiných kryptografických transformací najdeme i u hašovací funkce

[1] Pozn. autora: Měsíc před konáním konference došlo v kryptoanalýze k významné události: **Funkce MD5 byla prolomena analyticky. Oznámil to v srpnu t.r. tým čínských kryptologů na Rump Session konference CRYPTO 2004 [11].** Událost potvrzuje, že kolize hašovacích funkcí jsou stále vážnou hrozbou a že tuto oblast je nutné bedlivě sledovat.

jistý technický parametr, který určuje složitost obecného útoku hrubou silou, jenž je z principu vždy možný. Tím parametrem je zde délka výstupního kódu, která zde hraje obdobnou roli jako délka klíče u šifrovacích algoritmů. Má-li výstup hašovací funkce h délku n bitů, potom existuje algoritmus, který s vysokou pravděpodobností nalezne dvě kolidující zprávy se složitostí $\boldsymbol{O}(2^{n/2})$ operací výpočtu $h(x)$. Bližší popis použití tohoto univerzálního algoritmu založeného na takzvaném *narozeninovém paradoxu* lze najít v [4] a [6]. V článku [6] je vyložena paralelizovaná varianta uzpůsobená van Oorschotem a Wienerem přímo pro funkci MD5, která jasně ukazuje, že tato funkce by s ohledem na délku svého výstupu již neměla být považována za bezpečnou pro nepopiratelné podepisování. I několik let poté se však MD5 těší stále hojné oblibě, což spustilo iniciativu [5], která se snaží demonstrovat schůdnost hledání kolizí MD5 pomocí distribuovaného výpočtu v prostředí internetu. Autoři projektu odhadují, že úspěch v podobě nalezené kolize se dostaví zhruba do dvou let. To je zároveň doba, do které je ve všech důležitých aplikacích žádoucí nahradit MD5 vhodnou funkcí s nejméně 160 bitovým výstupem. Nabízí se například SHA-1 nebo další funkce z rodiny SHA (SHA-256, 384 nebo 512).

2.2 Vnitřní kolize

Zajímavý a prakticky nebezpečný příklad takové kolize u známého schématu DSA popsal Vaudenay v práci [9]. *Alternativní vysvětlení* založené na tomto druhu útoku je ve svém důsledku obdobné s případem z §2.1. I zde lze najít dvě různé správy m_1 a m_2 společně s podpisem s takové, že $Ver_{Pub}(m_1, s) = Ver_{Pub}(m_2, s) = \text{ANO}$. Jádro útoku spočívá v dílčí transformaci podepisovacího algoritmu DSA, při které probíhá výpočet části hodnoty podpisu v algebře mod q, kde q je prvočíslo splňující $2^{159} < q < 2^{160}$. V rámci tohoto výpočtu se přitom operuje s hodnotou $h(m)$, kde h je hašovací funkce SHA-1 a m je podepisovaná zpráva. Platí $0 \le h(m) < 2^{160}$, přičemž h se na tomto intervalu chová jako náhodná funkce. Díky tomu lze snadno najít prvočíslo q a zprávy m_1, m_2 splňující kongruenci $h(m_1) \equiv h(m_2) \pmod{q}$. Obě zprávy tak budou v rámci podepisovací transformace nerozlišitelné, což zde znamená, že povedou ke stejné hodnotě podpisu. Útočník si tak opět může později vybrat, kterou zprávu prohlásí za příslušející k danému podpisu. Dodejme, že celý útok musí být připraven už ve fázi generování klíče, přičemž útočit může jak sám signatář, tak i entita generující jeho klíče (certifikační autorita, atp.). Jak již bylo uvedeno, útok je poměrně nebezpečný a fakt, že je v řadě aplikací dosud přehlížen, evokuje představu dobře ukryté časované bomby.

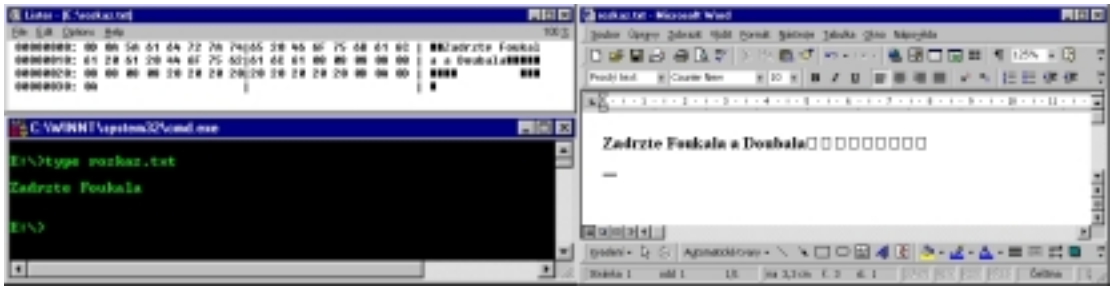
2.3 Kolize veřejných klíčů

Tuto problematiku poprvé nastínili Massias, Serret-Avila a Quisquater v práci [3]. Podrobné rozpracování pro schémata RSA a (EC)DSA, včetně formálního ukotvení pojmu klíčová kolize (k-kolize), pak bylo publikováno v [7] a [8]. V případě (EC)DSA bylo v [8] ukázáno, že pro tato schémata existuje triviálně schůdný postup, kterým lze pro libovolné dvě zprávy m_1, m_2 (stejné nebo různé) najít dva různé veřejné klíče Pub_1 a Pub_2 společně s hodnotou podpisu s splňující $Ver_{Pub_1}(m_1, s) = Ver_{Pub_2}(m_2, s) = \text{ANO}$. K oběma veřejným klíčům lze samozřejmě najít i klíče privátní a to tím způsobem, že ze znalosti jednoho z nich nelze určit druhý. Majitelé příslušných klíčů mohou při hledání k-kolize spolupracovat, nebo může kolizi k danému podpisu sestavit jen jeden z nich. To vše ukazuje na možnosti mnoha druhů různě sofistikovaných útoků. Nejpřímějším využitím je útok na nepopiratelnost s použitím dvojníka. Postup může být zhruba následující: Signatář se při podpisu riskantního dokumentu pojistí tím, že ke svému veřejnému klíči Pub_1 nalezne kolidující klíč Pub_2, který nechá registrovat na nějakou, pro daný případ bezvýznamnou osobu. V případě, že se věci budou vyvíjet kladně, ponechá tuto osobu i s klíčem Pub_2 v utajení. V případě ovšem, že se pro něho stane situace nepříznivou, přijde před soud s *alternativním vysvětlením*, že podpis pod inkriminovaným dokumentem není jeho, nýbrž osoby s klíčem Pub_2. Určitě bychom v praxi nalezli i takové případy, kdy je výhodnost nějakého podniku známa v čase t splňujícím $t_{\text{sig}} < t < t_{\text{id}}$, kde t_{sig} je okamžik podpisu nějakého klíčového dokumentu a t_{id} je okamžik uveřejnění identity signatáře. V takovém případě nebude ani nutné, aby signatář chodil s *alternativním vysvětlením* před soud. Jednoduše anonymně počká, jak situace dopadne, a podle toho odhalí buď svou vlastní identitu, nebo identitu svého dvojníka, aniž by celý systém zaregistroval cokoliv podezřelého. Konkrétně můžeme takovou situaci očekávat při podávání nabídek do výběrových řízení, kde se pak může pod jednou nabídkou ve skutečnosti skrývat několik (dva i více) různých subjektů, kteří se k ní později (ne)přihlásí až podle okolností známých v čase t.

2.4 Sémantické kolize, kódování zpráv

Tato problematika vychází z na první pohled triviální skutečnosti: Při vytváření a ověřování podpisu se vždy pracuje s binární reprezentací dotčené zprávy. Teprve v okamžiku její vizualizace (či jiné lidské interpretace) je zpráva prostřednictvím vhodného dekodéru převedena do symbolů srozumitelných pro člověka. Obdobná situace nastává i v případě, že příjemcem zprávy není člověk, ale nějaký stroj – například odpalovací rampa, řízená střela, zabezpečovací ústředna atp. Ve všech těchto případech vstupuje do hry nějaké dekódovací zobrazení $\varphi: \boldsymbol{M} \to \boldsymbol{L}$, kde $\boldsymbol{L}$ je jazyk příjemce. V současných aplikacích se stále poměrně často setkáme se stavem, kdy příslušné φ není pevně spojeno s podepisovanou zprávou. V takovém případě hrozí nebezpečí nalezení sémantické kolize

v podobě dvou různých dekodérů φ_1 a φ_2 takových, že $w_1 = \varphi_1(m)$ a $w_2 = \varphi_2(m)$ jsou dvě smysluplné, avšak sémanticky odlišné interpretace téže binární zprávy m. *Alternativní vysvětlení* je pak postaveno na tvrzení, že podepsaná zpráva se má interpretovat pomocí φ_2, nikoliv φ_1 (či obráceně). Příklad rozdílné interpretace jedné a téže binární zprávy je uveden na obrázku 1.



Obrázek 1: Příklad sémantické kolize dvou různých dekodérů

3. Základní protiopatření

Cílem této části je představit vybrané postupy návrhu a implementace podpisových schémat, které mají za cíl minimalizovat riziko útoků na nepopiratelnost v obecné rovině. Kromě nich je pak pochopitelně nutné vždy přihlédnout ještě ke konkrétním vlastnostem konkrétního použitého schématu, jejichž podrobný rozbor však již jde daleko za rámec tohoto příspěvku. Soustřeďme se proto na postupy ryze obecné. Prvním z nich je procedura generování klíčů, kde je nutné si uvědomit, že útočníkem na nepopiratelnost může být často sám legitimní signatář, čili oprávněný majitel privátního klíče. Řada procedur tento fakt dosud přehlíží a stará se zejména o slabiny, které mohou vzniknout, pokud bude klíč generovat nějaká třetí strana. Útoky představené v §2.2 a §2.3 však jasně ukazují, že ani sám budoucí oprávněný majitel privátního klíče by neměl mít proceduru generování klíčů zcela pod svou kontrolou. Možností, jak toho dosáhnout při zachování soukromí, existuje v kryptologii celá řada, avšak jejich použití většinou naráží na netriviální technické obstrukce. Technicky schůdných a přizpůsobivých postupů je zatím poměrně málo, jeden z nich byl pro (EC)DSA navržen v [8], kde je popsán protokol generování klíčů odstraňující útoky z §2.2 a §2.3. Další důležité opatření se týká identifikátorů použité hašovací funkce a dekodéru pro interpretaci podepisované zprávy. Ukázali jsme si, že oba tyto prvky jsou pro dosažení nepopiratelnosti podpisu zásadní, a proto by útočník neměl mít přílišnou (nejlépe žádnou) volnost v jejich volbě. V opačném případě by si tím mohl před nebo po vytvoření podpisu připravit půdu pro svůj útok. Proto je nutné uvedené identifikátory spojit s podepisovanou zprávou takovým způsobem, aby nebylo možné později některý vyměnit při zachování platnosti původního podpisu. V případě hašovacích funkcí lze tento problém považovat za uspokojivě vyřešený u (EC)DSA (používá implicitní vazbu na SHA-1) a RSA (identifikátor je součástí formátování podepisované zprávy). V případě identifikace dekodéru však na kryptologické úrovni dosud žádné uspokojivé řešení nenajdeme.

Závěr

Na úsvitu éry digitálních podpisů byla hlavní pozornost kryptologů soustředěna na jejich nepadělatelnost. Rozmáhající se praktické nasazení těchto schémat však ukazuje, že existuje obecnější a v jistém směru i silnější požadavek a tím je nepopiratelnost. Lze triviálně ukázat, že dosažení nepopiratelnosti implikuje nepadělatelnost, neboť v opačném případě by nevyvratitelný argument říkající, že podpis je padělek, snadno vedl k objektivnímu popření předloženého podpisu. Proto je dnes vhodnější bezpečnost digitálních podpisů nazírat právě přes nepopiratelnost, jakožto souhrn požadavků nutných pro konstrukci důvěryhodných služeb elektronického podepisování a vytváření nepopiratelných důkazů o vzniku a průběhu událostí v informačních systémech. Ukázali jsme si, že ne všechny problémy v této oblasti lze považovat za uspokojivě vyřešené, a zároveň jsme naznačili cesty, kterými by se jejich řešení na úrovni podpisových schémat mohlo ubírat.

Literatura

[1] ČSN ISO/IEC 13888-1, 2, 3, 4: *Nepopiratelnost*, ČSNI, květen 2001.
[2] ČSN ISO/IEC 10181-4: *Struktura nepopiratelnosti*, ČSNI, červen 1999.
[3] Massias, H., Serret Avila, X., and Quisquater, J.-J.: *Timestamps: Main issues on their use and implementation*, In Proc. of IEEE 8th International Workshop on Enabling Technologies, pp. 178-183, June 1999.
[4] Menezes, A.-J., van Oorschot, P.-C., and Vanstone, S.-A.: *Handbook of Applied Cryptography*, CRC Press, 1996.
[5] Attacking MD5 – distributed computing project, http://www.md5crk.com/
[6] Rosa, T.: *Paralelní hledání kolizí hašovacích funkcí*, CHIP 8/2001, str. 131-133, ftp://ftp.decros.cz/pub/Archiv/Publications/2001/chip-2001-08-131-133.pdf, 2001.
[7] Rosa, T.: *O klíčových kolizích v podpisových schématech*, workshop VKB 2002, pp. 14-26, Brno, 2002.

[8] Rosa, T.: *On Key-collisions in (EC)DSA Schemes*, CRYPTO 2002 Rump Session, IACR ePrint archive 2002/129, Santa Barbara, USA, August 2002.
[9] Vaudenay, S.: *Hidden Collisions on DSS*, in Proc. of CRYPTO '96, pp. 83-88, Springer-Verlag, 1996.
[10] Vyhláška č. 366/2001 Sb., aktuální platné znění viz http://www.micr.cz.
[11] Wang, X., Feng, D., Lai, X., and Yu, H.: *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, CRYPTO 2004 Rump Session, IACR ePrint archive 2004/199, Santa Barbara, USA, August 2004.
[12] Zákon č. 227/2000 Sb., o elektronickém podpisu, platné znění viz http://www.micr.cz.