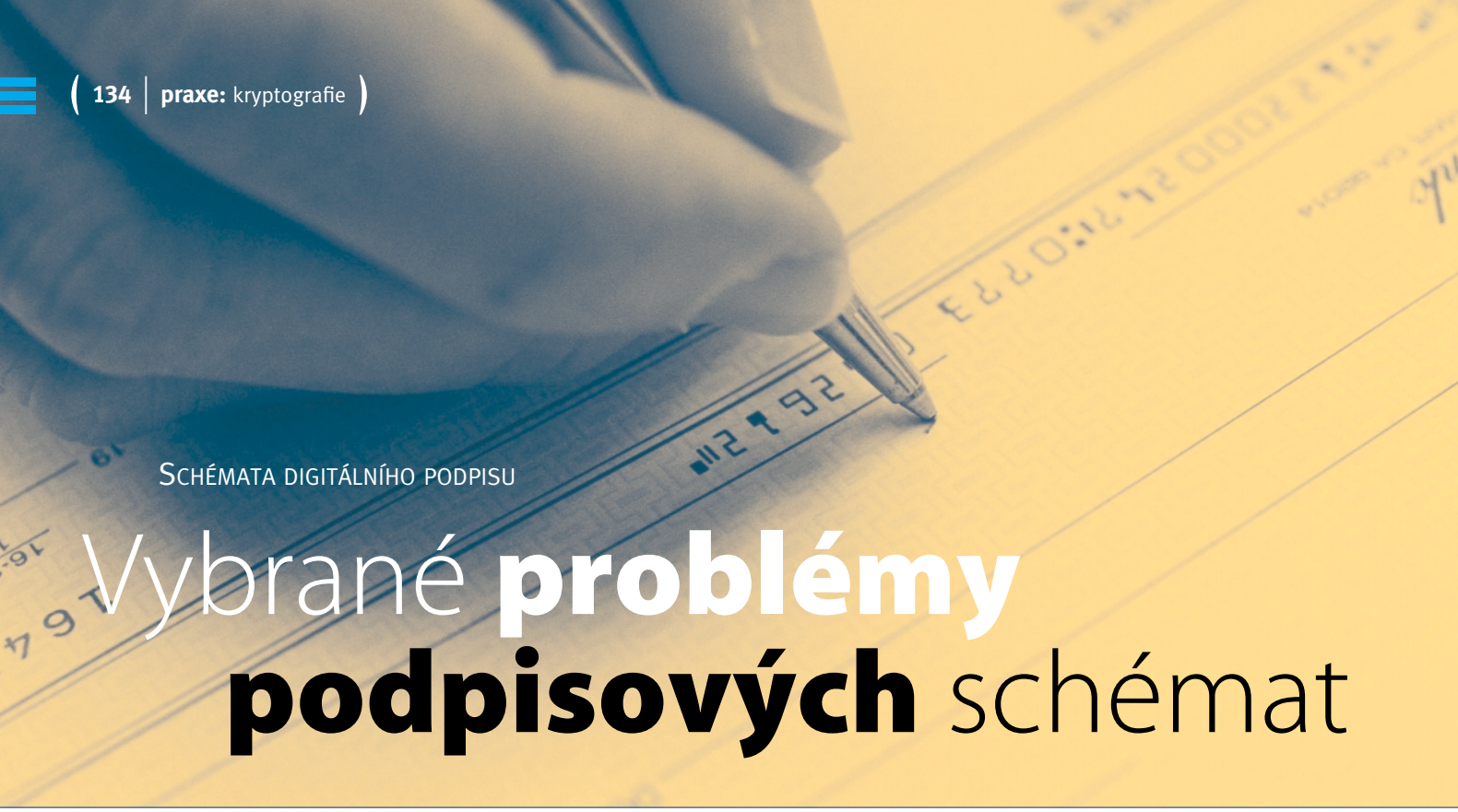

SCHÉMATA DIGITÁLNÍHO PODPISU

Vybrané problémy podpisových schémat

Tento článek je první z řady nepravidelných příspěvků, které mají za cíl navázat na předchozí dvoudílný úvod do problematiky schémat digitálních podpisů tím, že se budou snažit teoretickým, avšak zároveň vždy přístupným způsobem rozebrat některý z vybraných problémů v této oblasti, který bude v danou dobu aktuální.

Pro tento úvodní díl byly vybrány dva (řekněme zahřívací) problémy z oblasti schémat digitálních podpisů, kterými je jednak správná terminologie v názvosloví pro použité transformace, jednak poukázání na zajímavou snahu autority NIST spočívající v zavedení nových hašovacích funkcí SHA-256, -384 a -512.

ŠIFROVAT, ČI DEŠIFROVAT?

Začněme prvním problémem, na který se zaměříme vzhledem k silně zakořeněnému mylnému názoru, který spojuje podepisovací transformaci s operací šifrování. Nejvíce se s touto problematikou setkáváme u systému RSA, na který lze pohlížet jako na podpisové schéma vzniklé převodem asymetrické šifry na schéma digitálního podpisu. S ohledem na tento fakt může být za jistých okolností přípustné (nebo dokonce vhodné) používat místo názvů podepisovací a ověřovací transformace názvy operací „původních“ – tj. dešifrování a šifrování. V takovém případě je však třeba důsledně dodržovat vzájemné přiřazení těchto operací (tj. nezaměňovat šifrování s dešifrováním).

Tolik na úvod a nyní se vraťme k článku [SDP1], konkrétně k místu, kde jsme se zabývali převodem asymetrických šifrovacích schémat (AŠS) na schémata digitálního podpisu (SDP). Zde jsme rozbor této problematiky ukončili tím, že jsme si řekli, že dešifrovací transformaci AŠS budeme používat jako podepisovací transformaci SDP a šifrovací transformaci AŠS jako ověřovací transformaci SDP. Otázku, zdali při operaci podepisování provádíme operaci šifrování nebo dešifrování, jsme tak považovali za vyřízenou. Praktické zkušenosti však ukazují, že touha tvrdit, že při podepisování se šifruje, je v lidech natolik zakořeněna, že je patrně vhodné věnovat této otázce poněkud více prostoru. Z praktického hlediska možná jde o „pouhou“ formalitu, avšak při matematickém modelování kryptografie (které potřebujeme například pro formální důkazy bezpečnosti) je třeba mít v těchto základních otázkách zcela jasno.

Nejprve připomeňme, že obecný model AŠS se soustřeďuje zejména na šifrovací a dešifrovací transformace, které chápe jako zobrazení mezi množinami otevřených a šifrových textů. Klíče použité pro jednotlivá zobrazení zde přitom tyto transformace „pouze“ parametrizují (zcela obecný model uvažuje klíče jako indexy do množiny všech možných šifrovacích/dešifrovacích transformací). To, že u RSA vypadají základní definice obou transformací stejně, což nás nutí k jejich rozlišování podle použitého klíče, není důvodem k tomu, abychom na tomto obecném modelu něco měnili.

Dále si uvědomme, že v požadavcích na AŠS se podle obecného modelu hovoří pouze o složitosti (nemožnosti) dešifrovat náhodně

Vstup: veřejný klíč (n, e), šifrovaná zpráva m ($0 \le m \le n-1$)
- n použitý modul RSA
- e veřejný exponent RSA

Výpočet:
- i. $c = m^e \bmod n$
- ii. výstupem budiž hodnota c

obrázek 1: Šifrovací transformace RSA dle PKCS#1.

vybraný šifrový text s pouhou znalostí šifrovací transformace, kterou byl tento vytvořen. O složitosti opačného problému, tedy o tom, jak by bylo obtížné najít pro náhodně zvolenou otevřenou zprávu jí odpovídající šifrový text s pouhou znalostí dešifrovací transformace, se v tomto modelu nikde nemluví!

Z uvedeného tak dostáváme, že pokud chceme najít u AŠS ekvivalent k podepisovací transformaci u SDP, potom se musíme soustředit výhradně na použití **dešifrovací** transformace. Pouze tak totiž můžeme dokázat platnost základní vlastnosti SDP (viz [SDP1]). Považujeme-li podepisovaná data (z pohledu podepisovací transformace) za šifrový text c a jejich podpis za odpovídající hodnotu otevřeného textu m, potom můžeme tvrdit, že tento podpis nebude možné (pro danou hodnotu dat) nalézt s pouhou znalostí ověřovací transformace. Pomocí této transformace, kterou ztotožníme s operací **šifrování**, nicméně můžeme jednoznačně prokázat, že daný podpis je platný (na úrovni SDP) podle toho, zda platí $c = E_e(m)$ (přesnější způsob užití této transformace viz [SDP1]).

Vstup: privátní klíč (p, q, dP, dQ, qInv), dešifrovaný text c ($0 \le c \le n-1$)

p	prvočíselný faktor použitého modulu RSA
q	prvočíselný faktor použitého modulu RSA (n = p*q)
dP	$e*dP \equiv 1 \pmod{(p-1)}$
dQ	$e*dQ \equiv 1 \pmod{(q-1)}$
qInv	$q*qInv \equiv 1 \pmod{p}$

Výpočet:

i.	$m1 = c^{dP} \bmod p$
ii.	$m2 = c^{dQ} \bmod q$
iii.	$h = qInv*(m_1 - m_2) \bmod p$
iv.	$m = m_2 + h*q$
v.	výstupem budiž hodnota m

obrázek 2: Dešifrovací transformace RSA s využitím Čínské věty o zbytku podle PKCS#1.

Z uvedeného rozboru vyplývá, že pokud už chceme v případě schémat digitálního podpisu vzniklých převodem asymetrických šifrovacích schémat používat původní terminologii, potom musíme vnitřně přijmout fakt, že při podpisu dat je vlastně dešifrujeme.

Opravdu to nejde

Systém RSA má bohužel tu (ne)výhodu, že šifrovací i dešifrovací transformace představují v základní definici stejný vzorec. To by na první pohled možná mohlo někoho svádět k tomu, udělat alespoň u RSA výjimku a nazývat podepisovací

transformaci šifrováním. To, že toto ani v tomto případě není možné, ukážeme s využitím alternativního způsobu definice dešifrovací transformace, kterou uvádí například norma PKCS#1. Na obrázku 1 a 2 jsou uvedeny šifrovací a dešifrovací transformace systému RSA, které vyhovují zmíněné normě.

Vidíme, že v tomto případě je již způsob výpočtu obou transformací různý a využívá rovněž různého formátu uložení klíče. Ačkoliv hlavním cílem této úpravy bylo díky aplikaci Čínské věty o zbytku výrazně urychlit dešifrování, poslouží nám její sekundární vlastnosti výhodně k tomu, abychom ukázali, že transformaci uvedenou na obrázku 2 nelze označit jako šifrování.

Vlastní důkaz je velmi jednoduchý. Pokud by totiž tato transformace (označme ji pro konkrétní hodnotu klíče jako F) byla šifrováním, potom by muselo platit, že pro náhodně zvolený šifrový text c je s pouhou znalostí F výpočetně nemožné nalézt odpovídající otevřený text m (c = F(m)). Vzhledem ke způsobu definice F však toto neplatí. Vytvoříme-li z této transformace novou funkci (nazvěme ji G) tak, že místo hodnot dQ a dP použijeme hodnoty eQ a eP takové, že eQ*dQ ≡ 1 (mod (q-1)) a eP*dP ≡ 1 (mod (p-1)), potom snadno nalezneme hledané m jako m = G(c). Pro všechna m ∈ Z_n totiž platí, že G(F(m)) = m. Ani vlastní výpočet hodnoty funkce G ani její odvození ze znalosti funkce F přitom zcela jistě nejsou výpočetně nemožné. Odtud jasně vidíme, že funkci F ani při nejlepší vůli nemůžeme nazvat šifrováním.

Funkce SHA-xxx

Patrně máme ještě všichni v živé paměti datum 2. října roku 2000, které vstoupí do dějin kryptografie jako den, kdy byl zvolen nový nástupce již dosti ztrouchnivělého systému DES. Máme zde na mysli blokovou symetrickou šifru Rijndael, která byla autoritou NIST (National Institute of Standards and Technology) zvolena novým šifrovacím standardem zvaným AES – Advanced Encryption Standard. Více informací je možné nalézt přímo na webové stránce [AES]. Jako český zdroj pak mohu doporučit články na stránce [CRYPTO].

Standard AES jako takový náleží k technikám využívaným zejména k zajištění služby *důvěrnosti* přenášených dat, takže na oblast schémat digitálního podpisu nemá přímý dopad. Vzhledem k tomu, že dnes se již běžně setkáme s informačními systémy využívajícími kryptografii současně pro několik účelů (zejména pro služby *důvěrnost*, *autentizace subjektu a autentizace původu dat*), přičemž pro každý z těchto je požadována zhruba stejná úroveň bezpečnosti, je jistě vhodné zavést současně s AES ještě další kryptografické techniky umožňující této úrovně dosáhnout. S ohledem na tuto filozofii se autorita NIST rozhodla osvěžit také dosud vydané standardy hašovacích funkcí. Tento krok již začíná být pro oblast schémat digitálních podpisů zajímavý.

Kombinace Rijndael-yyy a SHA-xxx	Složitost luštění hrubou silou	Složitost hledání kolizí
Rijndael-128 a SHA-256	2^{127}	$2^{128.5}$
Rijndael-192 a SHA-384	2^{191}	$2^{192.5}$
Rijndael-256 a SHA-512	2^{255}	$2^{256.5}$

obrázek 3: Přehled doporučených kombinací funkcí SHA-xxx s AES (Rijndael).

Zatím jedinou hašovací funkcí, která je posvěcena autoritou NIST, je funkce SHA-1 definovaná dokumentem FIPS PUB 180-1 (její předchůdkyni SHA-0 neuvádíme). Jak víme, jedná se o hašovací funkci s délkou výstupního bloku 160 bitů, o níž se všeobecně předpokládá, že je jednosměrná (OWHF) a bezkolizní (CRHF – oba termíny viz [SDP2]). Z kryptografického hlediska se jedná o celkem oblíbenou a tudíž široce používanou hašovací funkci. Nabízí se proto otázka, proč zavádět funkce nové. Odpověď se ukrývá ve složitosti útoku hrubou silou na schémata digitálního podpisu, která by měla být zhruba stejná, jako je složitost útoku hrubou silou na algoritmus AES.

Pro lepší srozumitelnost si naznačené srovnání rozebereme podrobněji. Předpokládejme, že navrhujeme informační systém, který bude poskytovat služby *důvěrnost a autentizace původu dat* (chcete-li *zaručený elektronický podpis*). Pro zajištění důvěrnosti bude přitom použit algoritmus AES v blokovém režimu CBC. Zde budou podporovány všechny definované délky klíče tohoto algoritmu: 128b, 192b a 256b.

Základní odhad bezpečnosti takové služby můžeme s ohledem na použití útoku hrubou silou určit jako počet operací šifrování (nebo dešifrování) nutných k nalezení příslušného šifrovacího klíče e tak, že pro známou hodnotu otevřeného (OT) a šifrového textu (ŠT) budeme postupně zkoušet všechny možné klíče, dokud nebude platit ŠT = E_e(OT) (pro přehlednost používáme tuto jednoduchou symboliku). Je reálné předpokládat, že vzhledem k jisté redundanci zpráv v každém informačním systému bude útočník schopen potřebné páry (OT, ŠT) k provedení tohoto útoku nalézt.

Složitost zmíněného útoku, který vede při náhodné volbě klíče s cca 50% pravděpodobností k jeho nalezení, můžeme odhadnout na 2^{k-1} zmíněných operací, kde k je délka klíče. Vidíme, že i pro nejnižší délku klíče 128 bitů dostáváme zatím technologicky nepřekonatelnou složitost.

Podívejme se, jak jsme na tom u druhé ze služeb – u *zaručeného elektronického podpisu*. Předpokládejme, že pro vytvoření této služby je použito schéma digitálního podpisu s dodatkem. Délka klíče použitého asymetrického algoritmu je přitom volena tak, aby složitost základního známého útoku (v případě RSA se jedná o faktorizaci, u DSA jde o úlohu diskrétního logaritmu) odpovídala složitosti základního útoku na AES, která byla odvozena výše. Z předchozích pojednání ovšem víme, že bezpečnost výsledného schématu digitálního podpisu nezáleží jen na kvalitě použitého asymetrického systému. Ukázali jsme si, že zde velmi záleží též na kvalitách použité hašovací funkce.

Obdobně jako v případě symetrických blokových šifer můžeme i pro každou bezkolizní (CRHF) hašovací funkci nalézt základní druh útoku vedoucího k nalezení kolize, který je vždy (jako hledání klíče hrubou silou) teoreticky možný. V článku [SDP2] jsme si ukázali, že nalezení kolize u použité hašovací funkce vede (alespoň teoreticky) k prolomení daného schématu digitálního podpisu. Proto je rezistence vůči základnímu útoku hledání kolize pro danou hašovací funkci vhodným parametrem, podle kterého můžeme srovnat základní úroveň bezpečnosti poskytované touto funkcí a algoritmem AES.

Zmíněný základní způsob pro hledání kolizí u hašovacích funkcí vychází ze zajímavé pravděpodobnostní úvahy, která se označuje jako *narozeninový paradox*. Proto se útoky tohoto typu v originále často označují jako *birthday-attack*. Podrobnější rozbor a odvození narozeninového paradoxu nalezne čtenář v [MOV96]. My se zde omezíme pouze na připomenutí stěžejního tvrzení: Mějme náhodnou diskrétní veličinu X, která nabývá konečně mnoha (m) hodnot s rovnoměrným rozdělením. Potom se v posloupnosti hodnot této proměnné (x_1, x_2, x_3, ..., x_k) o délce k = $(m*2\ln(2))^{1/2}$, zhruba s padesátiprocentní pravděpodobností vyskyt-

nou dvě hodnoty stejné. (Narozeninový paradox získáme, pokud si uvědomíme, že ve skupině o pouhých 23 lidech budou se zhruba 50% pravděpodobností dva lidé se stejným datem – měsíc a den – narození.)

Připomeňme si ještě, že pro náhodné veličiny, jejichž počet možných hodnot odpovídá mocnině dvou ($m = 2^n$), existuje pro délku popsané posloupnosti odhad ve tvaru $k = 2^{n/2+0,5}$. Pomocí tohoto odhadu můžeme nyní jednoduše určit složitost základního útoku na libovolnou hašovací funkci o délce výstupního bloku n bitů, který spočívá v postupném zjišťování výsledků pro náhodně volené vstupní zprávy, a to tak dlouho, dokud v této posloupnosti nenajdeme dvě hodnoty stejné. Pro tyto hodnoty (x_i a x_j) potom platí, že h(zpráva$_i$) = h(zpráva$_j$) a dvojice (zpráva$_i$, zpráva$_j$) je hledaný kolidující pár vstupních zpráv.

Složitost takto pojatého útoku vyjádřená pro funkci SHA-1 vychází na zhruba $2^{80,5}$ zpracovaných zpráv. Vidíme, že to je (zanedbejme nyní paměťové nároky) výrazně méně než počet operací nutných pro útok na AES hrubou silou. Z tohoto pohledu tak můžeme vyvodit, že služba *zaručeného elektronického podpisu* v našem hypotetickém informačním systému poskytuje výrazně nižší úroveň základní bezpečnosti nežli předchozí služba *důvěrnosti*.

Abychom tento schodek vyrovnali, je třeba zavést nové hašovací funkce s většími délkami výstupních bloků, které by (podle narozeninového paradoxu) měly odpovídat dvojnásobkům standardních délek klíčů pro AES. Právě tímto směrem se autorita NIST vydala, když celkem nedávno uveřejnila návrhy hašovacích funkcí nazvaných jako SHA-256, SHA-384 a SHA-512 (jejich definice viz [SHA-xxx]). Číslo za pomlčkou přitom zcela zřejmě udává právě délku výstupního bloku. Předpokládá se, že tyto funkce budou vydány jako oficiální standard zhruba v době, kdy bude vydán standard AES (asi druhé čtvrtletí roku 2001).

Rozbor vlastních funkcí SHA-xxx již přesahuje rámec tohoto článku, takže se jím zde zabývat nebudeme. Pro nás bylo důležité zejména poukázat na nutnost udržení odpovídající úrovně bezpečnosti přes všechny použité mechanismy a uvést konkrétní způsob, který tohoto stavu umožňuje dosáhnout. Tabulka na obrázku 3 shrnuje doporučovaný způsob kombinace funkcí SHA-xxx pro schéma digitálního podpisu s algoritmem AES o příslušné délce klíče.

Závěrem této části ještě tři poznámky: Za prvé je třeba poznamenat, že pojem „odpovídající úroveň bezpečnosti“ je sice zajímavým zaklínadlem, avšak teoretický aparát umožňující přesný popis tohoto fenoménu zatím chybí (a asi ještě dlouho chybět bude). Doporučení plynoucí z popsaných úvah je tak třeba chápat jako nejlepší možné odhady, jejichž dodržením rozhodně nelze nic pokazit. Druhá poznámka se pak týká toho, že NIST není rozhodně jedinou institucí, která nabízí „delší“ hašovací funkce. Z ostatních zmiňme například RIPEMD-320. Na SHA-xxx je však zajímavé to, že pocházejí od stejné autority jako AES, což má jistě určitou váhu.

Konečně třetí poznámka říká, že nevyrovnaná základní bezpečnost u jednotlivých služeb IS ještě neříká, že je některá z těchto služeb přímo napadnutelná. Říká pouze tolik, že mezi bezpečností jednotlivých služeb existují určité disproporce, které mohou být z určitého pohledu na škodu.

Závěr

V tomto příspěvku jsme si ukázali, že podepisovací transformace u podpisových schémat vzniklých převodem schémat šifrovacích musí odpovídat zásadně operaci dešifrování. Proto je třeba uvádět, že při podpisu se data dešifrují, nikoliv zašifrují.

Dále jsme upozornili na navrhované standardy nových hašovacích funkcí. Zde jsme uvedli, že primárním účelem těchto funkcí, jejichž finální uvolnění je plánováno spolu s uvolněním algoritmu AES, není poskytnout prostředek pro odvozování klíče pro algoritmus AES (i když se samozřejmě výborně hodí i k tomuto účelu), ale umožnit vyrovnání základní úrovně bezpečnosti u jednotlivých služeb informačního systému.

Tomáš Rosa | *tomas.rosa@decros.cz*

literatura

[MOV96] Menezes, A. J., van Oorschot, P. C., Vanstone, S. A.: *Handbook of Applied Cryptography*, CRC Press 1996
[SDP1] Rosa, T.: *Podpis pro pokročilé (1)*, CHIP 11/00, str. 174 – 178, dostupné v [CRYPTO]
[SDP2] Rosa, T.: *Podpis pro pokročilé (2)*, CHIP 12/00, str. 172 – 176, dostupné v [CRYPTO]
[CRYPTO] Archiv článků http://www.decros.cz/Security_Division/Crypto_Research/archiv.htm
[AES] Advanced Encryption Standard, http://csrc.nist.gov/encryption/aes/
[SHA-xxx] http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf