Pro a proti RSA s e=3

Symbol e v titulu samozřejmě znamená veřejný exponent schématu RSA (viz *ST 3/2004* v [3]). V praxi je dnes jeho běžnou hodnotou prvočíslo 65 537. Avšak například v oblasti elektronických platebních karet najdeme často i nastavení e=3. Někoho to nechává ledově chladným, jiní volají po okamžité nápravě, neboť někde slyšeli, že „RSA s trojkou“ je prolomené.

Argument pro *e*=3

Důležitým důvodem pro použití trojky může být zrychlení šifrovací transformace a transformace ověření digitálního podpisu. Tyto transformace mají tvar x^e mod N, kde x je vstupní hodnota, N je modul RSA a e již zmíněný veřejný exponent. V algoritmu square-and-multiply, který se pro enumerace takových výrazů v praxi používá, budeme pro e=3 potřebovat právě jednu operaci druhé mocniny a násobení. To je pro RSA nejmenší možná složitost. Pro e=65537 potřebujeme 16krát druhou mocninu a jedno násobení.

Tento exponent však bohužel nezrychluje jen užitečné operace. Existuje celá řada útoků na chybné implementace RSA (přehledově viz [1]), jejichž praktická schůdnost silně závisí na hodnotě e. Čím menší jeho hodnota je, tím snazší je útok. Za takzvané malé hodnoty jsou přitom považována čísla řádu jednotek až desítek.

Exponované útoky

Nebude-li řečeno jinak, budeme dále předpokládat e=3. Mezi klasické demonstrační útoky patří využití slabiny ve formátování šifrovaných zpráv. Předpokládejme, že není použito žádné speciální formátování. To znamená, že zpráva m je jen převedena na celé číslo $0 \le m < N$ a rovnou zašifrována na šifrový text $c=m^3$ mod N. Dále nechť odesilatel zašle třem různým adresátům stejnou zprávu m. Adresáti ať používají nezávislé nesoudělné moduly N_1, N_2 a N_3. Odesilatel tedy vytvoří tři šifrové texty c_1, c_2 a c_3, kde $c_i=m^3$ mod N_i. Útočník, který všechny tři šifrové texty zachytí, z nich může s využitím Čínské věty o zbytku schůdnou cestou vypočítat celé číslo C, splňující $C=m^3$ mod B, kde $B=N_1*N_2*N_3$. Vzhledem k tomu, že $0 \le m^3 < B$, platí uvedená rovnost přímo na celých číslech, tedy $C=m^3$. Odtud už lze hledanou zprávu triviálně najít přes aproximaci reálné třetí odmocniny. Útok lze zobecnit pro libovolnou hodnotu e, avšak praktický úspěch v luštění můžeme očekávat jen pro malé hodnoty. Například pro exponent 65 537 bychom potřebovali šifrové texty jedné a téže zprávy pro 65 537 adresátů, což je vyjma přístupů přes cílenou modifikaci programového kódu nepravděpodobné.

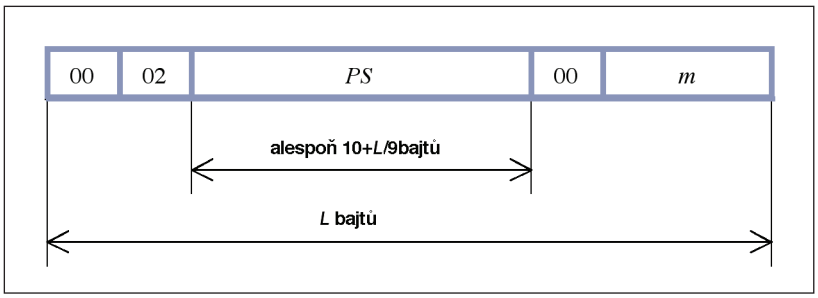
Za celou rodinou dalších, pokročilejších útoků stojí významný algoritmus [2]. Mějme polynom

$$p(x)=x^k + a_{k-1}x^{k-1} + \ldots + a_0,$$

jehož koeficienty jsou celá čísla. Dále buď N celé kladné číslo. Takzvaný Coppersmithův algoritmus (viz odkaz [1] pro jeho formálně přehlednější obdobu od Howgrave-Grahama) dokáže schůdnou cestou najít všechna x_0 splňující $|x_0| < N^{1/k}$ a $p(x_0)$ mod N=0. Jinými slovy, umíme najít všechny malé kořeny $p(x)$ modulo N. Základní aplikace pro chybně implementované RSA je přímočará: Předpokládejme, že odesilatel opět zapomněl šifrovanou zprávu správně zformátovat a útočník zachytil hodnotu $c=m^3$ mod N pro nějakou neznámou zprávu m. Všimněme si, že m je kořenem polynomu $p(x)=(x^3-c)$ mod N, neboť:

$p(m)=(m^3-c)$ mod $N=(c-c)$ mod N=0. Pokud hledaná zpráva splňuje $m<N^{1/3}$, může ji útočník výše uvedeným algoritmem snadno najít. Pro modul 1024 bitů to znamená, že lze úspěšně luštit každou neformátovanou zprávu kratší než 341 bitů, což s přehledem pokrývá přenosy běžných klíčů symetrických šifer. Se zvyšující se hodnotou e přitom praktická hrozba tohoto útoku viditelně rychle klesá, což rozhodně není důvodem pro zanedbávání formátování.

V příspěvku [2] je dále popsán útok na zprávy, které sice už nějak formátovány jsou, avšak nevhodným způsobem. Jako příklad si můžeme vzít momentálně nejrozšířenější standard PKCS#1 verze 1.5 (viz *ST 10/2003*), který nám připomíná *obr. 1*. Vidíme, že část zformátované zprávy tvoří řetězec náhodných nenulových bajtů *PS*. Jedná se o účelový doplněk, který má za cíl vycpat mezeru mezi úvodní hlavičkou 00 02 a vlastní zprávou. První, co nás pod dojmem Coppersmithova algoritmu napadá, je nutnost utajit doplněk. V opačném případě bychom mohli malá m úspěšně hledat jako kořeny polynomu: $p(x)=(2*256^{L-2}+PS*256^{a+1}+x)^3 - c$ mod N, kde a je délka zprávy a L je délka modulu (vše v bajtech). Jinou implementační chybou je použít příliš krátký doplněk. Útok na takové schéma je mimochodem jednou z nejelegantnějších aplikací Coppersmithova algoritmu. Předpokládejme, že odesilatel zašifruje jednu a tutéž zprávu dvakrát pro stejného příjemce. To se může snadno stát při opakovaném přenosu stejného symetrického klíče, atp. Vždy přitom svědomitě použije novou tajnou hodnotu *PS*. Pokud však vždy $PS<N^{1/9}$ (obecně $PS<(N^{1/e})^{1/e}$), pak je i přes jeho svědomitost schéma úspěšně luštitelné. To umožní Coppersmithův algoritmus, když dovedně využije ještě další, dříve známé útoky. Prakticky vzato, pro modul N délky 1024 bitů nesmí velikost doplňku klesnout pod 15 bajtů. Lépe je ovšem držet se zde alespoň na 25 bajtech. Obecně pak doporučujeme hranici $10+L/9$. Bohužel existují rozšířené knihovní implementace RSA, které takových doporučení nedbají. Ostatně ani samotný standard PKCS#1 není v této věci zcela správný. To vše jsou zapomenuté chybičky, které se při e=3 mohou stát fatálními. S rostoucí hodnotou e schůdnost útoku opět rychle klesá díky podmínce $PS<(N^{1/e})^{1/e}$.



Obr. 1 Formátování zprávy podle "PKCS#1 ver. 1.5"

Závěr

Nastavení e=3 může významně urychlit šifrovací, respektive ověřovací transformaci. Na druhé straně bohužel i významně usnadňuje využití implementačních slabin (viz též útoky na podpis, *ST 12/2006*). Samo o sobě však dosud známou slabinou není. Věříme-li, že žádné faux pas v aplikaci my sami ani naši partneři (!) nemáme, můžeme „trojku“ bez obav použít.

Vlastimil Klíma, Tomáš Rosa,
v.klima@volny.cz, trosa@ebanka.cz

LITERATURA

[1] Boneh, D.: Twenty Years of Attacks on the RSA Cryptosystem, Notices of AMS, vol. 46, no. 2, pp. 203–213, 1999

[2] Coppersmith, D.: Small Solutions to Polynomial Equations and Low Exponent RSA Vulnerabilities, Journal of Cryptology, Vol. 10, pp. 233–260, 1997

[3] E-archivy http://cryptography.hyperlink.cz, http://crypto.hyperlink.cz