Kryptologie pro praxi – Achillova pata DSA

V předchozím dílu jsme si ukázali hlavní zranitelnost šifry RSA vzhledem k hrozbě úniku informace postranním kanálem. Nyní na náš výklad navážeme a ukážeme si, kde je v podobném případě Achillova pata hojně rozšířeného (zejména v USA) podpisového schématu DSA (viz *ST 4/2004*). Ačkoliv se jedná v obou případech o jiný typ schématu, tak můžeme oba díly s výhodou řadit za sebe, neboť základní rysy obou zranitelností jsou velmi podobné. V obou případech existuje v průběhu operace s privátním klíčem jistá proměnná, jejíž hodnota musí zůstat pro útočníka dokonale skryta. Pokud by z modulu unikala byť jen nepatrná část její hodnoty, potom by bylo možné cílenou interakcí s napadeným modulem celé schéma prolomit. V případě DSA je situace ještě dále komplikována jednak tím, že prolomení zde znamená přímo získání hodnoty privátního klíče, jednak v tom, že dotčená proměnná je v rámci příslušné operace generována jako náhodné číslo, což poměrně zvyšuje riziko její kompromitace (vadný zdroj náhodných čísel, silnější parazitní vyzařování, cílené ovlivnění pomocí elektromagnetických šoků, atp.).

Matematická podstata

Připomeňme si nejprve podepisovací transformaci schématu DSA: Označme h hašový kód zprávy M, kterou máme podepsat. Víme, že h v tomto případě počítáme ze vztahu h = SHA-1(M) (hašovací funkce SHA-1 viz *ST 2/2004*). Vlastní podpis je reprezentován dvojicí celých čísel (r, s) vypočtených z rovnic:

$$r = (g^k \bmod p) \bmod q, \quad (1)$$
$$s = (h + xr)k^{-1} \bmod q, \quad (2)$$

kde $kk^{-1} \bmod q = 1$ a k je náhodně vygenerované číslo z intervalu <1, q–1>. Hodnoty (p, q, g) tvoří veřejné parametry schématu a x je celé číslo představující privátní klíč (podrobněji viz *ST 4/2004*). Hodnota k bývá označována jako klíč zprávy nebo také jako tzv. „nonce“ (z anglického výrazu Number-used-ONCE). Už při základním popisu DSA jsme si uvedli, že hodnotu k musíme držet v tajnosti, neboť po jejím prozrazení lze z rovnice (2) triviálně určit hodnotu privátního klíče. Snadno lze také ukázat, že její použití musí být v souladu se zkratkou „nonce“ skutečně jednorázové: Označme (r_1, s_1) podpis vypočtený pro zprávu s hašovým kódem h_2 a (r_2, s_2) podpis pro zprávu s kódem h_2. S využitím (2) můžeme psát:

$$(h_1 + xr_1)s_1^{-1} \equiv k_1 \pmod q,$$
$$(h_2 + xr_2)s_2^{-1} \equiv k_2 \pmod q.$$

Pokud by byla pro oba podpisy použita stejná hodnota k=k_1=k_2, což je stav, který může útočník podle r_1=r_2 snadno detekovat, potom bychom dostali triviální soustavu dvou rovnic o dvou neznámých, jejímž vyřešením bychom získali přímo hledaný privátní klíč $x = (h_2s_1–h_1s_2)(r_1s_2–r_2s_1)^{-1} \bmod q$. Zvažujeme-li rizika tohoto typu útoku, je nutné se u dané implementace zamyslet i nad tím, zda útočník nemůže cíleně měnit kontext generátoru náhodných čísel tak, aby se posloupnost generovaných čísel po nějaké době opakovala. Pak už stačí jen čekat na zachycení správné dvojice podpisů.

Kromě úniku celé hodnoty k, případně jejího opakování, však dnes díky práci [2] umíme využít i situaci, kdy známe třeba jen několik jejích bitů. Formální odvození útoku je pochopitelně daleko za rámcem tohoto článku (i když je v něm řada zajímavých míst a určitě stojí za to přemýšlet o jeho dalším možném vylepšování), avšak my si zde ukážeme, že celý postup lze snadno přepsat do podoby jednoduché „kuchařky“, kterou snadno zvládne implementovat každý zkušenější hacker. Pro jednoduchost předpokládejme, že známe vždy b nejnižších bitů hodnoty k. Známe tedy hodnotu a=$k \bmod 2^b$. Nyní budeme pro každý zachycený podpis (r, s) počítat hodnoty:

$$t = rs^{-1}2^{-b} \bmod q,$$
$$u = [(a–hs^{-1})2^{-b} \bmod q]+q/2^{b+1}$$

Upozorněme, že hodnota u je zde chápána jako racionální číslo. Postupně tak získáme N dvojic $(t_1, u_1), \ldots, (t_N, u_N)$. V [2] je ukázáno, že každá z těchto dvojic jistým způsobem aproximuje privátní klíč x. K vyřešení těchto vztahů a získání hodnoty privátního klíče bude hacker potřebovat vhodnou knihovnu implementující algoritmus pro aproximační řešení problému nejbližšího vektoru (CVP – Closest Vector Problem) na (N+1) dimenzionální racionální mřížce definované bázovou maticí $\boldsymbol{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_N, \boldsymbol{b}_{N+1})$, $\boldsymbol{b}_i \in \boldsymbol{Q}^{N+1}$, $\boldsymbol{b}_1 = (q, 0, \ldots, 0)$, $\boldsymbol{b}_2 = (0, q, 0, \ldots, 0)$, $\ldots$, $\boldsymbol{b}_N = (0, 0, \ldots, q, 0)$, $\boldsymbol{b}_{N+1} = (t_1, t_2, \ldots, t_N, 2^{-b-1})$. Takové implementace lze na Internetu poměrně snadno najít. Kryptoanalytici často vycházejí například z oblíbené knihovny NTL [5]. V ideálním případě pak už stačí jen vhodné funkci předložit výše uvedenou bázi společně s vektorem $\boldsymbol{u}$, ke kterému se má nejbližší vektor mřížky v hledat. V našem případě položíme $\boldsymbol{u}$=$(u_1, \ldots, u_N, 0)$, kde u_i jsou aproximační koeficienty získané výše. Lze ukázat [2], že současné algoritmy pro řešení CVP vrátí za uvedených podmínek pro vhodně zvolené parametry (N, b) vektor, jehož poslední souřadnice s vysokou pravděpodobností splňuje rovnici v_{N+1}=$(x+\gamma)2^{-b-1}$, kde $\gamma \not\equiv 0 \pmod q$. Odtud pak již snadno vypočteme privátní klíč ze vztahu x=$2^{b+1}v_{N+1} \bmod q$.

Volba parametrů (N, b) již vyžaduje hlubší porozumění použité metodě spolu s jistou dávkou experimentování. V krajním případě lze ovšem správné nastavení najít čistě experimentálně na modelovém případě plánovaného útoku. Pro ilustraci uvádíme, že podle [2] v 90 % případů fungovalo nastavení (N=70, b=4), čili stačilo, aby útočník znal 4 nejnižší bity hodnoty k pro 70 podpisů. Pokud bychom znali více bitů, tak podle [1] můžeme očekávat téměř jistý úspěch například pro (N = 27, b = 8). Odtud vidíme, že máme co do činění s velmi efektivní metodou, která si v razanci útoku příliš nezadá s postupem prezentovaným pro RSA v minulém dílu (*ST 4/2005*).

Představenou metodu lze s mírnými obměnami využít pro útok se znalostí libovolných shluků bitů v hodnotě k. Rozdíly jsou pak (podobně jako u RSA) ve složitosti takového útoku, přičemž výše uvedený případ využití postranní informace patří mezi ty výrazně efektivnější postupy.

Praktické důsledky

Základním důsledkem je, že klíč zprávy musí být v případě DSA generován a chráněn opravdu hodně pečlivě, neboť potenciální útočníci mají k dispozici skutečně velmi efektivní nástroj na využití úniku byť jen nepatrné informace o této hodnotě. Zdůrazněme, že velmi záleží na tom, aby generované hodnoty pokrývaly skutečně celý interval <1, q–1>. V praxi může snadno nastat situace, kdy má programátor k dispozici zdroj náhodných binárních vektorů, které jsou „jen o pár bitů“ kratší, než je délka odpovídající uvedenému intervalu (obecně 160 b). Pokud by ho pohodlnost přemohla a on by bez dalších úprav tento zdroj použil pro DSA, bylo by celé schéma rázem prolomitelné. Připomeňme, že chyby tohoto druhu s takto fatálními následky se už skutečně staly (viz *ST 11/2004*).

Jak jsme si už naznačili výše, musíme při analýze bezpečnosti dané implementace uvažovat i s tím, že útočník není nutně odkázán jen na to, aby se snažil vygenerované hodnoty k odposlechnout nebo čekal na chyby programátorů. Místo toho se je může snažit a priori už během generování známým způso-

bem měnit. Touto cestou se ubírá inspirativní práce [1], jejíž autoři prakticky demonstrují možnost prolomení DSA na čipové kartě, která je náchylná k chybovým útokům indukovaným pomocí napěťových impulzů. S využitím této techniky bylo možné během generování hodnoty *k* nastavit určitý počet nejnižších bytů na známou hodnotu danou technickými parametry karty a použitého mikrokódu. Pak už jen stačilo vygenerovat příslušný počet podpisů a popsanou metodou z nich vypočítat hodnotu privátního klíče. Dodejme také, že podpisy získané tímto způsobem byly ve smyslu ověřovacího algoritmu DSA (*ST 4/2004*) normálně platné. Pokud by tedy snad napadená karta spoléhala na ochranu tím, že každý podpis před jeho odesláním ověří, nebylo by jí to zde nic platné. Stojí za poznámku, že v tomto případě jsme svědky vzniku tzv. subliminálního (podprahového) postranního kanálu (*ST 3/2003* a [4]), který je charakteristický právě tím, že bez důkladné analýzy není z přenášených dat patrný. Půjdeme-li dál tímto směrem, můžeme jednoduchou úpravou metody [2] ukázat další praktický důsledek a tím je, že implicitní ověřování platnosti právě vypočtených podpisů nejenže nemůže být chápáno jako postačující ochrana proti chybovým útokům, ale že může při nevhodném použití celkovou zranitelnost vůči těmto útokům dokonce zvýšit [3]. To pochopitelně neznamená, že bychom toto opatření neměli používat vůbec. Znamená to však, že si musíme být jisti, že to není naše jediná obrana a že způsob jejího použití nenapomáhá k vytváření podprahových postranních kanálů.

Závěr

Prakticky každý kryptografický algoritmus má kromě vlastního tajného (ať už sdíleného symetrického nebo soukromého asymetrického) klíče ještě nějaké jiné jasně viditelné slabé místo, které útočník může využít k jeho napadení. Před sto lety Auguste Kerckhoff von Nieuwendhoff postuloval tehdy revoluční myšlenku, že musíme předpokládat, že protivník získá (utajovaný) popis šifry. Dnes však tento více než sto let starý princip z pochopitelných důvodů nemůže postačovat k plné reflexi záludností současných informačních technologií. Ty jsou tak rozmanité, že útočník může probíhajícímu algoritmu někdy doslova koukat pod ruce a získávat tak cenné informace pro své nekalé úmysly. Některé informace jsou přitom k tomuto účelu výrazně vhodnější než jiné, takže nám zde během výpočtu dočasně vzniká cosi, co by snad šlo nazvat „klíče druhého druhu“, tj. něco, co musíme s ohledem na současnou technologii útoků chránit stejně dobře jako vlastní klíče. V *ST 4/2005* jsme si ukázali příklad klíčů druhého druhu pro RSA, nyní jsme obdobný rozbor provedli pro podpisové schéma DSA.

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

LITERATURA

[1] Naccache, D., Nguyen, P.-Q., Tunstall, M., and Whelan, C.: Experimenting with Faults, Lattices and the DSA, in Proc. of Public Key Cryptography – PKC'05, pp. 16-28, Springer-Verlag, 2005

[2] Nguyen, P.-Q., and Shparlinski, I.-E.: The Insecurity of the Digital Signature Algorithm with Partially Known Nonces, Journal of Cryptology, Vol. 15, No. 3, pp. 151-176, Springer-Verlag, 2002

[3] Rosa, T.: Lattice-based Fault Attacks on DSA - Another Possible Strategy, přijato na konferenci Security and Protection of Information 2005, 3. - 5. května 2005, Brno

[4] Rosa,T.: Modern Cryptology – Standards Are Not Enough, doktorská disertační práce, 2001-2004, dostupné v [6]

[5] Shoup, V.: Number Theory C++ Library (NTL), http://www.shoup.net/ntl/

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