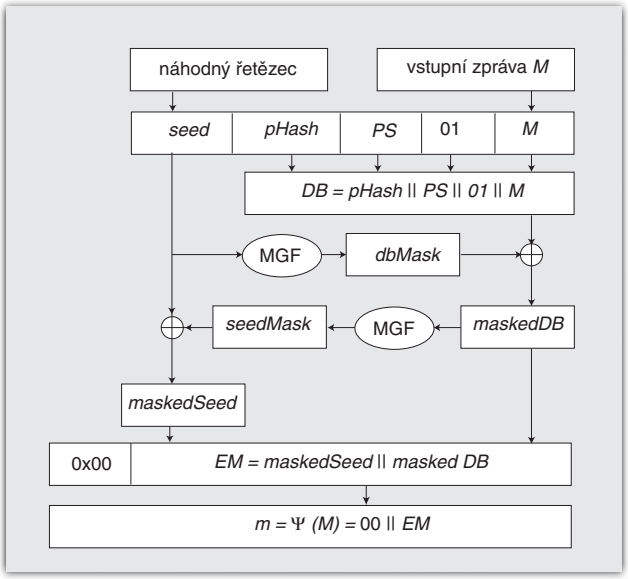
Kryptologie pro praxi – formátování a bezpečnost

V tomto pokračování navážeme na obecné seznámení s asymetrickými schématy [4] a blíže rozvedeme problematiku správného formátování šifrovaných či podepisovaných zpráv. Jak jsme si už naznačili, představuje tato oblast v podstatě Achillovu patu všech současných asymetrických kryptosystémů. Do jisté míry lze tuto problematiku chápat jako paralelu k módům symetrických šifer [5].

Nejprve se věnujme šifrování. Mějme dánu instanci nějaké asymetrické šifry (použitá značení viz [4]), která určuje transformace E_V a D_P takové, že definiční obor funkce E_V představuje celočíselný interval $<0,N-1>$, případně $(0,N-1>$, kde N je celé kladné číslo. Tuto podmínku splňuje jak RSA, tak i ElGamal, tedy oba nejčastěji používané systémy. Naším cílem je nyní definovat kódovací a dekódovací transformace Ψ a Ψ^{-1}, které se budou používat při šifrování a odšifrování zpráv. Budeme je proto nazývat šifrovým kódováním, ačkoliv toto kódování samo o sobě šifrou není(!) – viz [4]. Šifrování dané zprávy M pak probíhá jako $m=\Psi(M)$, $C=E_V(m)$, kde C je výsledný šifrový text určený k odeslání (uložení). Odšifrování pak provedeme jako $m=D_P(C)$, $M=\Psi^{-1}(m)$. Patrně největší kus práce v oblasti sjednocení formátovacích postupů učinila společnost RSA, která navrhla a stále aktualizuje popis čtyř praktických metod (dvě pro šifrování, dvě pro podpis) ve svém, de facto, mezinárodním standardu PKCS#1 [6].

Nejrozšířenější šifrové formátování je v [6] nazýváno komplikovaným názvem EME-PKCS1-v1_5, častěji se však setkáme s různými zkrácenými formami tohoto názvu. Podstatnou informací, která by měla být v každém jméně zachována, je, že formátování pochází z doby, kdy platila verze 1.5 standardu [6]. Tehdy zde bylo uvedeno právě toto jediné šifrové kódování a patrně by při tom i zůstalo nebýt objevu několika jeho zásadních slabin (přehled viz [6]). To bylo důvodem k uvedení nové verze standardu s novým šifrovým kódováním, které se označuje jako EME-OAEP. I přes některé výhrady (viz také [1]) je tato metoda považována za prakticky bezpečnou. Formátování z verze 1.5 ovšem zcela zapuzeno nebylo. Důvodem je nutnost udržet kompatibilitu s ohromným množstvím existujících aplikací navržených a vyvinutých dříve (zejména před rokem 1998). Naštěstí lze i u verze 1.5 udržet jakousi úroveň bezpečnosti za podmínky, že budou velmi pečlivě dodržena implementační doporučení uvedená v [6]. Jedná se však o balancování na velmi úzké hraně, což dokazuje mimo jiné i zatím poslední objev slabiny v exponovaných protokolech SSL/TLS [3], která byla zavíněna právě tím, že tyto protokoly jsou jednak nuceny udržovat zpětnou kompatibilitu s verzí 1.5 a jednak si tvůrci aplikací s nějakými implementačními problémy příliš nelámali hlavu.



Obr. 1 Datově-procesní diagram robustního kódování EME-OAEP [6]

Vzhledem k masovému rozšíření a jednoduchosti popisu si zde ukážeme hlavní rysy formátování EME-PKCS1-v1_5. Pro popis poněkud komplikovanější metody EME-OAEP, kterou nám v hrubých rysech ilustruje *obr. 1,* odkazujeme na [6]. Označme B délku binárního zápisu N (viz výše) v bajtech, čili $2^{8(B-1)} \le N < 2^{8B}$. Analogicky označme L délku šifrované zprávy M. Potom kódování zprávy M probíhá podle předpisu $m=\Psi(M)=00\|02\|PS\|00\|M$, kde operace $\|$ znamená zřetězení bajtových řetězců a PS představuje náhodný (nutno generovat pro každý výpočet znovu!) řetězec nenulových bajtů délky $B-L-3$. Pro převod mezi celými čísly a bajtovými řetězci je použita běžná konvence, kdy první bajt zleva je nejvýznamnější. Poznamenejme, že levostranná nula v řetězci m nemusí mít vždy plnou délku 8 bitů, což závisí na konkrétní bitové délce čísla N. Výpočet inverzní hodnoty $\Psi^{-1}(m)$ probíhá zřejmým způsobem, kdy se hodnota M jednoduše přečte z konce řetězce m. Na uvedeném popisu lze ilustrovat i základní slabinu této metody, spočívající v komplikovaném ošetření chybových stavů. Představme si automat, který je nucen rutinně přijímat šifrové texty C, odšifrovat je (předpokládejme nejčastější metodu RSA) a výsledek předat dalším částem systému ke zpracování. Tento popis se mj. přesně hodí pro určitý modul běžného internetového serveru, který používá bezpečnostní protokoly SSL/TLS. Problém nastává v okamžiku, kdy odšifrovací transformace $m=D_P(C)$ vrátí takovou hodnotu m, která nespadá do oboru hodnot funkce Ψ (například řetězec m nezačíná $00\|02$), čili nemohla reálně při odeslání zprávy vzniknout. V takovém případě nelze výpočet $\Psi^{-1}(m)$ smysluplně provést. Jako velmi samozřejmé řešení celé situace se jeví prostě informovat odesílající stranu o chybě ve formátování a celou operaci tím ukončit. Jenže to by právě byla osudová chyba! Pokud by totiž útočník mohl pro libovolné C takto jednoduše zjišťovat, zda m prošlo kontrolou formátování či nikoliv, potom by byl schopen vyluštit libovolný zachycený šifrový text. Toto dosti šokující tvrzení je důsledkem objevu postranních kanálů (přehled viz [2]). Proto správné řešení i tak triviální otázky, jako je reakce na chybové stavy, je podstatně komplikovanější, než by se zdálo a vyžaduje si hlubší kryptologický rozbor konkrétní aplikace (více viz [6], [3]). Pro návrh nových aplikací proto nelze než důrazně doporučit orientaci na EME-OAEP. I zde je ovšem nutno dávat na postranní kanály dobrý pozor [6], [1].

Pro realizaci podpisového schématu metodou ze [4], což je například pro RSA velmi obvyklé, nabízí PKCS#1 na výběr opět dvě varianty podpisového formátování a to sice EMSA-PKCS1-v1_5 a EMSA-PSS. První z nich je přitom nejen svým názvem blízké výše popsané metodě pro šifrové kódování. Stejně jako jeho souputník, tak i EMSA-PKCS1-v1_5 je založeno na jednoduchém řetězcově orientovaném doplňování a mělo by být postupně nahrazováno komplikovanějším EMSA-PSS. Podstatný rozdíl je zde ovšem ten, že tato náhrada je motivována zatím pouze velmi teoretickými slabinami, které nemají žádný praktický dopad na bezpečnost. Otázka přechodu zde proto není tak žhavá, i když designéři zbrusu nových a kompatibilitou nezatížených aplikací by asi váhat neměli. Za zmínku zde stojí fakt, že EMSA-PSS nepřináší jen výhody, ale že se zde objevuje zranitelnost, která u předchozího formátování nebyla. Jedná se o potenciální skrytý kanál, který může

teoreticky i prakticky sloužit k záměrnému vytvoření podprahového postranního kanálu [2]. Z obecného hlediska se ovšem nejedná o zásadní nedostatek, nýbrž o vlastnost, kterou musí mít návrhář na zřeteli a vyrovnat se s ní.

Ukažme si nyní stručně metodu EMSA-PKCS1-v1_5. S využitím zavedeného značení můžeme kódovací předpis psát jako $\Psi(M)$=00||01||*FF*…*FF*||00||ID_h||$h(M)$, kde ID_h je specifický identifikátor použité hašovací funkce (viz [6]) a řetězec $h(M)$ je hašový kód podepisované zprávy *M* vypočtený zvolenou hašovací funkcí *h*. Doplňovací řetězec *FF*…*FF* má takovou délku, aby délka celého sestaveného řetězce byla právě *B* bajtů, přičemž levostranná nula může být opět neúplná. Podpis *S* zprávy *M* se postupně vypočítá jako y=$\Psi(M)$, $S=D_P(y)$. Při ověřování podpisu se nejprve vypočítá y=$E_v(S)$. Poté se ověří, že řetězec *y* patří do množiny hodnot transformace Ψ (mj. s ohledem na ID_h). Nakonec se vypočte $h(M)$ a porovná se s hodnotou na konci řetězce *y*. Pokud všechny kontroly vyjdou (podrobně viz [6]), je podpis prohlášen za platný. V ostatních případech je odmítnut jako neplatný. Připomeňme, že narozdíl od šifrového kódování je zde doplňovací řetězec tvořen konstantními bajty *FF*. V tomto případě absence náhodnosti nejenže prakticky nevadí, ale je současně i jistou prevencí zmíněného podprahového kanálu. Pro popis robustnější metody EMSA-PSS odkazujeme opět na [6].

Všechny zde představené metody původně vznikly zejména pro spojení s mechanizmem RSA, avšak jsou natolik obecné, že je lze kombinovat i s jinými asymetrickými mechanizmy. Proto jsme je zde coby samostatný stavební blok oddělili od RSA s tím, že o konkrétní kombinovatelnosti s jinými metodami (ElGamal, atp.) vždy ještě příslušně pojednáme.

Vlastimil Klíma, Tomáš Rosa,
v.klima@volny.cz, t_rosa@volny.cz

LITERATURA

[1] Klíma, V., Rosa, T.: Further Results and Considerations on Side Channel Attacks on RSA, in Proc. of CHES '02, San Francisco Bay, CA - USA, pp. 245 - 260, Springer-Verlag, 2002

[2] Klíma, V., Rosa, T.: Vybrané aspekty moderní kryptoanalýzy, Sdělovací technika č. 3, str. 3-7, 2003

[3] Klíma, V., Pokorný, O., Rosa, T.: Attacking RSA-based Sessions in SSL/TLS, in Proc. of CHES '03, Cologne, Germany, September 7-11, 2003

[4] Kryptologie pro praxi (3), Sdělovací technika 2003, č. 8, str. 22

[5] Kryptologie pro praxi (4), Sdělovací technika 2003, č. 9, str. 16

[6] PKCS#1: RSA Cryptography Standard, http://www.rsasecurity.com/rsalabs/pkcs/index.html