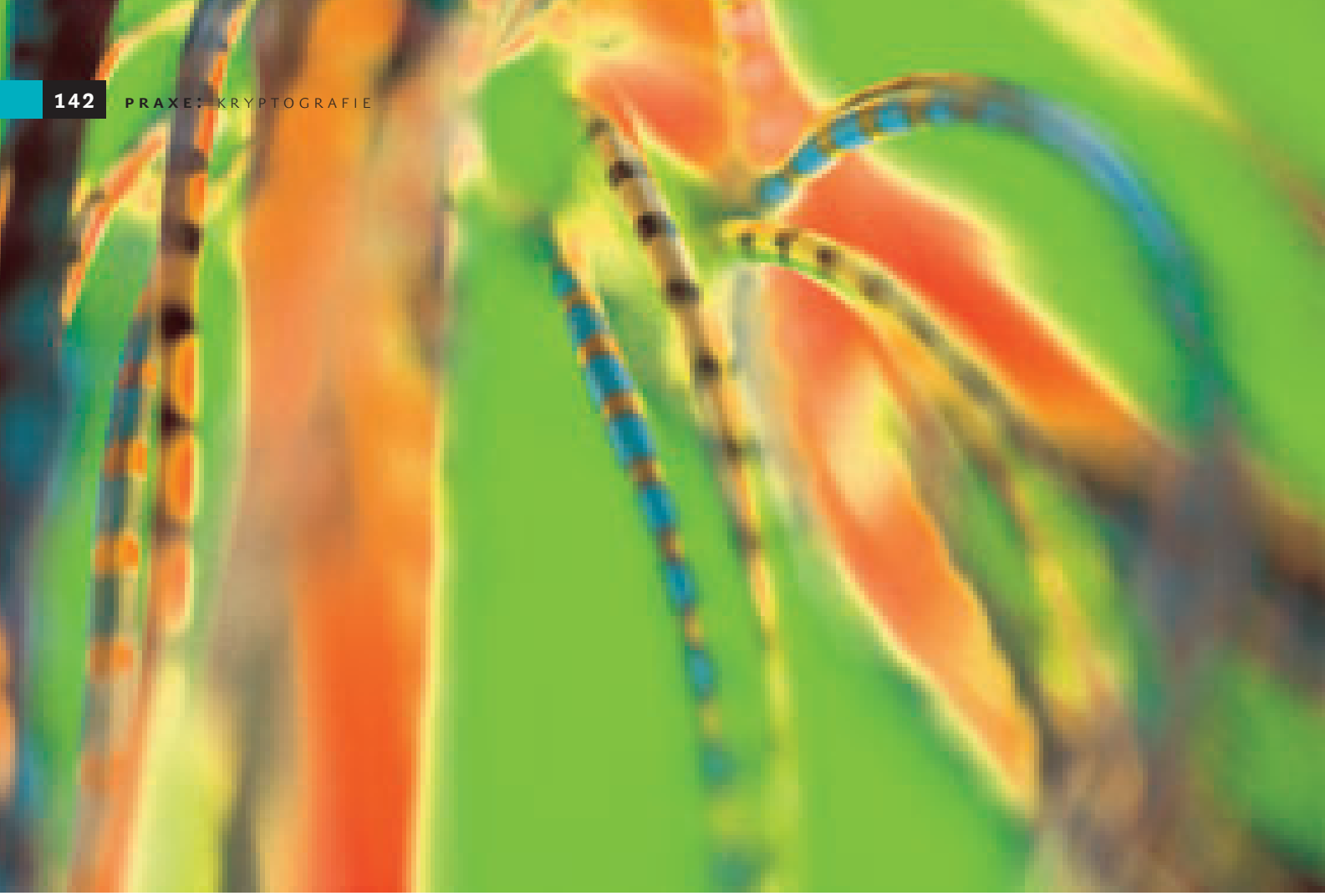


AES

Nová šifra nastupuje

Po čtyřech a půl letech od vyhlášení soutěže na výběr nové šifry vstupuje v platnost AES – od letošního 26. května může být tento standard používán k ochraně neutajovaných senzitivních dat ve státní správě USA a očekává se, že se stane nejrozšířenější komerční šifrou na světě. Přinášíme několik užitečných informací jak pro manažery, tak pro ty, kdo budou AES implementovat.

Šifra pro třetí tisíciletí – i takto bombasticky znějícím přídomkem už byl ozdoben nový standard. Zde zůstaneme při zemi a – spokojeni s představou, že vydrží alespoň pár desetiletí – blíže se s ním seznámíme. Nejprve základní údaje: **AES** znamená *Advanced Encryption Standard* čili **pokročilý šifrovací standard** (nepoužívejte slovo „kryptovací", je to špatně!). Nahrazuje DES, který byl jeho předchůdcem od roku 1977. Novou šifru schválil 26. 11. 2001 americký Národní úřad pro standardizaci (NIST) v publikaci FIPS PUB 197 jako federální standard USA s účinností od 26. 5. 2002.

Bezpečnost AES

Očekává se, že AES bude mít životnost minimálně 20 až 30 let. Protože mu ale nehrozí útok hrubou silou (vyzkoušení všech možných klíčů), což bylo u DES možné (viz DES-Cracker v [4]), není třeba se obávat, že by platnost AES musela být po uplynutí 30 let z bezpečnostních důvodů ukončena kvůli krátkým klíčům. Možná si teď říkáte, že se určitě nějaká metoda na luštění najde. Jistě, taková možnost tu jistě je. Ale výběr trval čtyři a půl roku právě proto, aby se všechny potenciální teoretické slabiny mohly vyloučit. A k bezpečnosti si řekneme ještě více.

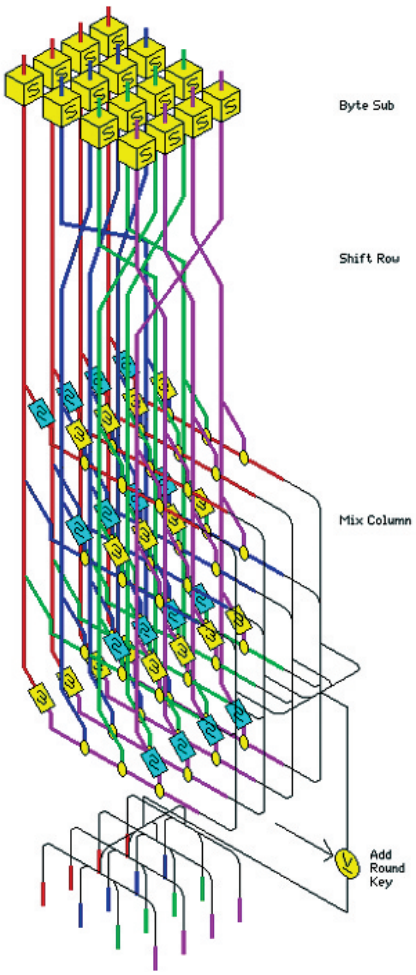
AES stojí na starých základech

Kdo se už blíže seznámil s popisem AES, bude možná překvapen: AES se na první pohled diametrálně liší od DES, ale opak je pravda! AES ve skutečnosti vychází z těch teoretických principů, které byly použity u DES a za celých 25 let existence DES nikdy nebyly zpochybněny. I když toto tvrzení by vydalo na hlubší kryptologickou přednášku, je fakt, že podstata AES je postavena na poměrně starých kryptologických základech, které odolaly všem výhradám nejen v uplynulých čtyřech a půl letech testování AES, ale ve skutečnosti nejméně posledních 30 let.

Druhou bezpečnostní zárukou AES jsou délky klíčů. AES podporuje tři délky klíčů, a to 128, 192 a 256 bitů. Současné lidské technologie ani veškeré pozemské zdroje nestačí na to, aby mohly útočit hrubou silou na 128bitový klíč, a žádné vhodné ani nejsou ve výhledu. (A i kdyby byly, vždy je možné přejít na klíč delší.)

Odolá AES „nekřemíkovým" počítačům?

Pokud by došlo k pokroku například v oblasti kvantových počítačů nebo počítačů na bázi DNA, bude to jistě známo dostatečně dlouho (odhadujeme 10 až 15 let) před tím, než by taková technologie mohla být prakticky použitelná na lámání dlouhých klíčů AES. Jinými slovy, z dosud uvedeného vidíme, že bezpeč-



Obr. 1. Základní skupina operací (jedna runda) algoritmu AES (viz [9])

nost AES je zajištěna velmi zodpovědně. Víc neumíme...

Na druhé straně však nemůžeme tvrdit, že AES nelze rozluštit (a přesto to určitě někdy uslyšíte). Kdo by to říkal, bude lhát, protože z informačně teoretického hlediska to vyloučit nelze. Jestli ale nějaké šifře byla v dosavadní historii kryptologie věnována mimořádně velká pozornost, pak to byly právě AES a DES.

Pokud bych si tedy měl vybrat, kterou šifrou ochránit svá nejcennější data, použil bych buď TripleDES, nebo AES. Protože TripleDES je pomalejší (trojnásobné použití DES s třemi obecně různými klíči, blíže viz [8]), nemůže aspirovat na nový standard; jinak však, podobně jako AES, využívá výhod DES a odstraňuje hlavní nevýhodu krátkého klíče DES. Používá totiž trojnásobně dlouhý klíč (168 bitů) a vliv ostatních nevýhod DES (slabé klíče a vlastnost komplementárnosti) se dá vhodnou implementací eliminovat.

Implementace

Manažeři, kteří se budou rozhodovat mezi TripleDES nebo AES, si musí být vědomi, že **AES zpracovává vstupní blok o délce 128 bitů**, což je dvojnásobek bloku, který dosud používaly všechny šifrovací standardy (včetně DES, TripleDES a mnoha dalších). Ve starších produktech tedy bude nahrazení délky bloku zpracovávaných dat a změna velikosti klíče dost často znamenat velký zásah do zdrojových kódů a někdy i uživatelského rozhraní (třeba výběr délky klíče, délka klíčové fráze apod.). TripleDES je proto při nahrazování DES v tomto směru trochu ve výhodě.

AES se nepochybně stane nejpoužívanější šifrou v historii.

V nových produktech bude zcela určitě lepší přejít na AES. Pokud se pro jeho implementaci rozhodnete, nemusíte vše programovat zcela od počátku. V [5] naleznete odkazy na stránky obsahující zdrojové kódy AES v různých programovacích jazycích a pro různé platformy (nejčastěji osmi- a 32bitové). Tyto kódy sice za programátora neřeší vše, protože tak jako tak musí dojít k jejich přizpůsobení pro dané podmínky, je ale z čeho vycházet a vývojáři jistě tuto možnost ocení – také proto, že tak dostanou k dispozici testovací vektory.

Optimalizace

Neobejdete-li se bez optimalizace, ať už paměťové nebo časové, budete se muset hlouběji ponořit do zdrojových kódů a popisu AES. Naštěstí je AES ideální pro realizaci různými způsoby s ohledem na kompaktnost nebo rychlost, neboť k tomu lze využít různé zákonitosti, které nejsou z jeho „pedagogického" popisu na první pohled zřejmé.
V tomto článku se pochopitelně nemůžeme věnovat všem možným konkrétním problémům, protože předem neznáme příslušné specifické podmínky, ale snad vám v něčem pomůže alespoň naše osobní zkušenost.

Před časem jsme stáli před úkolem optimalizovat rychlost AES na procesoru MPC850 rodiny PowerPC (Motorola, 32bitová sběrnice, BIG ENDIAN). Zvolenou metodu i postup jsme podrobně popsali v prezentaci přednesené na konferenci Vojenská kryptografie 2001 (viz [3]). Odráží stav uprostřed práce na výzkumně-vývojovém projektu CSP II MicroCzech, v jehož rámci bylo ve spolupráci s Národním bezpečnostním úřadem ČR vyvíjeno šifrovací zařízení pro ochranu utajovaných informací. Přestože od té doby ještě došlo k dalším zlepšením (zejména v rychlosti, neboť naměřená rychlost zahrnovala nejen vlastní proces šifrování, ale i další kryptologická opatření), naleznete zde všechny hlavní myšlenky optimalizace. Jsou použitelné nejen pro vybraný procesor, ale i v jiných prostředích, a zejména popsaný postup ukazuje obecnou metodu zrychlování.

Základní problém, který musí programátor vyřešit, je vidět z obrázku 1 a z části programového kódu na obrázku 2 (viz též [9] a [3]). Jde o skupinu operací (tzv. rundu), které se ve schématu opakují mnohokrát po sobě (10x, 12x nebo 14krát podle délky klíče). Ze vstupu musí být extrahovány jednotlivé bajty, i kdyby se jinak pracovalo s 32bitovými slovy. Každý tento dílčí bajt pak ovlivňuje čtyři bajty na výstupu. Naštěstí se jeho vliv dá tabelovat, takže se tyto jednotlivé operace dají uchovávat v různě velkých

Obrázku 1 odpovídá následující pseudokód. Je zde vidět, že jednotlivá 32bitová vstupní slova bi[x] se musí rozbít na bajty, které procházejí určitými tabulkami, jejichž výstupy formují příslušné slovo výstupu (bo[y]), k němuž je navíc ještě "přixorován" klíčový materiál. Tabulky v sobě sdružují jak substituční boxy, tak tělesové násobení. Blíže viz [3].

```
void f_nround_c(u4byte* bo, u4byte* bi, u4byte** k)
{
bo[0] = ft_tab[0][byte(bi[0],0)] ^ ft_tab[1][byte(bi[1],1)] ^
ft_tab[2][byte(bi[2],2)] ^ ft_tab[3][byte(bi[3],3)] ^ (*k)[0] ;
bo[1] = ft_tab[0][byte(bi[1],0)] ^ ft_tab[1][byte(bi[2],1)] ^
ft_tab[2][byte(bi[3],2)] ^ ft_tab[3][byte(bi[0],3)] ^ (*k)[1] ;
bo[2] = ft_tab[0][byte(bi[2],0)] ^ ft_tab[1][byte(bi[3],1)] ^
ft_tab[2][byte(bi[0],2)] ^ ft_tab[3][byte(bi[1],3)] ^ (*k)[2] ;
bo[3] = ft_tab[0][byte(bi[3],0)] ^ ft_tab[1][byte(bi[0],1)] ^
ft_tab[2][byte(bi[1],2)] ^ ft_tab[3][byte(bi[2],3)] ^ (*k)[3] ;
*k = *k + 4;
}
```

Obr. 2. Pseudokód jedné rundy zašifrování

tabulkách, nebo naopak realizovat delším kódem při úspoře paměti tabulek. Ostatní je programátorovi jistě celkem jasné.

Trocha kryptologie nikoho nezabije...

V popisu se hovoří o násobení v různých algebraických strukturách (Galoisova tělesa s 2^8 a 2^{32} prvky, viz očíslované obdélníčky na obr. 1), ale na vše, kromě substitučních boxů, vystačíte s operací XOR a bitovými posuny. Jinými slovy, substituční boxy (S) na obrázku 1 reprezentují tzv. nelineální část schématu a zbytek je jeho lineární (přesněji řečeno afinní) část. Totéž se týká přípravy tzv. rundovních klíčů. Odtud ta podobnost s algoritmem DES, kde jedinou nelineární částí také byly tzv. substituční boxy. AES je ale má kvalitnější, neboť mají lepší nelineární vlastnosti a jsou dvakrát širší. Další finesy naleznou programátoři ve zmíněné práci i ve vlastním popisu AES [6].

Klíče pro AES

Souběžně s AES pracoval NIST také na nových standardech hašovacích funkcí, a to SHA-256, SHA-384 a SHA-512, které mají delší výstupní kódy (délky hašovacích kódů jsou přímo v názvech funkcí). Ostatně, podrobněji jsme o nich v Chipu už psali, viz [10]. Hašovací funkce jsou často používány k vytváření klíčů pro různé šifry, protože dovedou kryptograficky kvalitně zpracovat klíčovou frázi (*passphrase*) nebo přístupové heslo (*password*) o různých délkách. Přitom na jejich výstupu obdržíme perfektně náhodně vyhlížející bitový řetězec, mající předem pevně danou délku.

Častým nedorozuměním je domněnka, že pokud potřebujeme klíče v délkách 128, 192 a 256 bitů pro AES, musíme je generovat po řadě funkcemi SHA-256, 384 a 512 a z výstupu vzít vždy jen polovinu. Je dobré vědět, že to tak nemusíme dělat za každých okolností (i když to je z bezpečnostního hlediska zcela v pořádku). Stačí si uvědomit, co znamená zkrácení výstupního kódu nějaké hašovací funkce. Například pokud vezmeme polovinu 512bitového výstupu funkce SHA-512, vytváříme ve skutečnosti novou hašovací funkci (mohli bychom ji označit jako „SHA-512/2"), jejíž užitné vlastnosti se nijak zvlášť nebudou lišit od SHA-256. Pak ale můžeme použít rovnou SHA-256 s plnou délkou kódu!

Ve většině případů můžeme všechny klíče pro AES bez obav generovat jen funkcí SHA-256 a její výsledek eventuálně oříznout na potřebnou délku. Jediné bezpečnostní riziko by vzniklo tehdy, pokud bychom z jedné klíčové fráze generovali všechny tři délky klíčů pro AES pomocí jediné hašovací funkce. Potom by nám vyzrazení klíče v jednom případě dalo část klíče i pro druhý a třetí případ využití. Tomu lze zabránit generováním klíčů různých délek jedinou hašovací funkcí, ale pokaždé s využitím jiné tzv. „soli". Hodnoty soli mohou být náhodné, nebo to mohou být konstanty, ale pro různé druhy použití různé. Například můžeme mít pravidlo, že klíče délek x = 128, 192 a 256 bitů budou z klíčové fráze derivovány jako SHA-256(passphrase || const(x)), kde const(128), const(192) a const(256) jsou různé konstanty definující sůl, která se připojuje za klíčovou frázi. Můžete ovšem využít i techniku HMAC nebo standard PKCS#5 apod.

Z hlediska odlišností různých implementací je šifra AES velmi pružná.

AES je založen na stejných principech jako DES, v otázce bezpečnosti je však „o třídu výš".

Jaký modus vybrat?

Definice AES je standardní definicí blokové šifry. Je to tedy zobrazení, které při daném klíči převádí 128bitový vstupní blok otevřeného textu na 128bitový výstupní blok šifrového textu. Tento druh šifrování se nazývá „elektronická kódová kniha" (ECB, *Electronic Code Book*) a k šifrování souborů ap. se příliš nehodí. Stačí si totiž uvědomit, že stejné bloky otevřeného textu produkují stejné bloky šifrového textu, a tak by třeba platební příkaz na 1000 Kč mohl být snadno padělán na částku 1 000 000 Kč, a to jednoduchým přidáním jednoho bloku šifrového textu do výsledné šifrové zprávy, viz obrázek 3. Operaci na obrázku 3 přitom lze provést bez jakékoliv znalosti šifrovacího klíče!

.....	3tdszj34	j7čžuths	bgžc4rš7	rg43č7řz	.....
.....		převeďte 1	0 0 0	,- Kč	.....

.....	3tdszj34	j7čžuths	bgžc4rš7	bgžc4rš7	rg43č7řz	.....
.....		převeďte 1	0 0 0	0 0 0	,- Kč	.....

Obr. 3. Bloková šifra v modu elektronické kódové knihy nemusí být vždy bezpečná...

K šifrování se spíše používají jiné režimy neboli „mody", jak říkají kryptologové; kromě CFB a OFB jde zejména o modus CBC (více o modech jsme psali v [11]). V souvislosti s přijetím AES hodlá NIST standardizovat i tyto mody a definovat navíc jeden nový – tzv. čítačový modus. Kdy se to stane, se dozvíte na stránkách NIST [6]. Stojí za upozornění, že zde existují velmi čerstvé útoky proti modu CBC, o kterých budeme čtenáře Chipu informovat v nejbližší době, včetně doporučovaných protiopatření. Pro ty, kteří problém modu CBC nutně potřebují znát už nyní, uvádíme předběžně alespoň to, že souvisí s postranními kanály (viz [2]) a je popsán v práci [12], kterou se nám podařilo v předstihu získat.

Malé shrnutí

Na závěr si připomeňme, že AES je veřejně dostupný standard, za jehož použití se neplatí žádné licenční poplatky. Nese razítko amerického standardizačního úřadu NIST a od 26. 5. 2002 je možné jej v americké státní správě používat k ochraně citlivých neutajovaných informací. Očekává se, že se i ve světě stane převládajícím symetrickým algoritmem, jako tomu bylo u algoritmu DES před 25 lety. Letošní 26. květen se tak ve světě šifrování stává historickým datem.

■ ■ ■ Vlastimil Klíma, *autor@chip.cz*

Literatura:

[1] Archiv článků *http://www.decros.cz/bezpecnost/_kryptografie.html*, kde jsou citované články dostupné v elektronické podobě

[2] Klíma V., Rosa T.: seriál článků o postranních kanálech, Chip 2001, 2002

[3] Klíma V., Rydlo P.: Optimalizace rychlosti algoritmu AES v instrukčním souboru procesoru PowerPC, Vojenská kryptografie IV, Sborník příspěvků, Brno, 2001

[4] Kladivo na DES, Chip 11/98, str. 74 - 75

[5] domácí stránka AES od autorů (obsahuje různé implementace): *http://www.esat.kuleuven.ac.be/~rijmen/rijndael/*

[6] domácí stránka AES od NIST (obsahuje oficiální popis a další informace): *http://csrc.nist.gov/encryption/aes/*

[7] Stručný popis AES: Chip 11/99, str. 64 - 65

[8] O TripleDES: Chip 6/00, str. 56 - 59

[9] Popis AES s obrázky: *http://home.ecn.ab.ca/~jsavard/crypto/co040801.htm*

[10] O hašovacích funkcích SHA: Chip 8/01, str. 138 - 139

[11] O modech blokových šifer: Chip 7/00, str. 50 - 53

[12] Vaudenay, S.: CBC Padding: Security Flaws in SSL, IPSEC, WTLS, ... to appear in Eurocrypt 2002