SERPENT je jedním z pěti kandidátů na Advanced Encryption Standard (AES). O celém výběrovém řízení se podrobněji dozvíte v úvodu k této sérii stručných popisů všech finalistů, a to v článku "Bitva o trůn vrcholí" v Chipu 10/99; zde se už věnujeme přímo technickému popisu šifry. Připomeňme jen, že AES se stane šifrovacím standardem pro příští století (nebo alespoň nějaká ta desetiletí) a bude mít dalekosáhlý vliv na počítačovou bezpečnost.

Představujeme kandidáty na AES:

Šifra SERPENT

Blokovou šifru ***SERPENT*** přihlásili do soutěže Ross Anderson (UK), Eli Biham (Izrael) a Lars Knudsen (Norsko), známá esa světové kryptologie. Jako u všech kandidátů na AES je délka vstupního a výstupního bloku 128 bitů a podporované délky klíče 128, 192 a 256 bitů. SERPENT používá pevné substituční tabulky (osm S-boxů zobrazujících 4 bity na 4 bity) a pracuje v rundách podobně jako DES, má však dvojnásobný počet rund (32). Se 128bitovým blokem a 256bitovým klíčem je přibližně stejně rychlý jako DES, je ale bezpečnější než TripleDES.

Návrh šifry je dost konzervativní. Autoři nechtěli použít žádné nové prvky (datově závislé rotace, násobení nebo sčítání místo operace $\oplus$ apod.), a proto výhodně aplikovali osvědčené principy tak, aby se šifra dala dobře hardwarově i softwarově implementovat. Zejména, jak uvidíme dále, je kladen důraz na možnost paralelního zpracování jednotlivých bitů a možnost výpočtu rundovních klíčů za chodu ("on-the-fly"). Díky tomu, že návrh je bitově orientovaný, umožňuje optimalizovat programový kód pro různé mikroprocesory. Odšifrování (je popsáno v hlavním dokumentu; viz infotipy) zde však probíhá trochu jinak než zašifrování, takže nelze použít stejný hardware jako u šifry MARS.

Postup při zašifrování

Před operací zašifrování nebo odšifrování anebo v jejím průběhu se vypočítá 33 rundovních klíčů. Jsou to 128bitové hodnoty K[i], i = 0..32, z nichž každou chápeme jako zřetězení čtyř 32bitových slov k_{4*i+0} , k_{4*i+1} , k_{4*i+2} , k_{4*i+3} (jejich výpočet popíšeme dále). Otevřený text se naplní do 128bitového bloku B[0] a v každé z 32 rund (i = 0..31) se z B[i] vypočte B[i+1]. Výsledný šifrový text je uložen v B[32].

Runda **i** se skládá z následujících kroků (viz též obrázek 1). Na vstupní 128bitový blok B[i] se "naxoruje" rundovní klíč K[i]. Obě proměnné jsou chápány jako čtyři 32bitová slova, takže také výsledek B[i] $\oplus$ K[i] je možné chápat jako čtyři 32bitová slova. Nyní se tato slova seřadí tak, že jejich bity

vytvářejí “čtyřřad” (viz obr. 2), takže na prvním místě stojí za sebou první bity slov, na druhém místě druhé bity atd. Nyní aplikujeme substituční box S_i postupně na všech 32 popsaných čtveřic bitů (v i-té rundě použijeme jeden S-box S_i). Protože rund je 32 a S-boxů osm, používají se S-boxy “dokola”, tedy $S_i = S_{i \bmod 8}$.

Výstupem S-boxu jsou čtyři bity, které zase uložíme do čtyřřadu tak, jak byl seřazen vstup. Čtyři nově vzniklá 32bitová slova (tvořící ony čtyři řady) si označme x_0, x_1, x_2, x_3. Dosavadní operace pak můžeme zapsat ve tvaru $(x_0, x_1, x_2, x_3) = S_i(B[i] \oplus K[i])$. S novými hodnotami slov **x** nyní provedeme lineární transformaci L podle pseudokódu na obrázku 1 a obdržíme nové hodnoty 32bitových slov $x = L(x_0, x_1, x_2, x_3)$. Ty už tvoří výstup z i-té rundy $B[i+1] = (x_0, x_1, x_2, x_3)$. Ještě poznamenejme, že u 32. rundy je lineární transformace nahrazena operací XOR s rundovním klíčem K[32].

Příprava klíčů

Rundovní klíče se vytvářejí poměrně jednoduše. Pokud šifrovací klíč (128, 192 nebo 256 bitů) nemá délku 256 bitů, doplní se na ni bitem 1 a dále nulovými bity. Ten se naplní po řadě do osmi 32bitových proměnných w_{-8}, w_{-7}, ..., w_{-1} a ty se dále expandují až do w_{131} podle vzorce $w_i = (w_{i-8} \oplus w_{i-5} \oplus w_{i-3} \oplus w_{i-1} \oplus \phi \oplus i) <<< 11$, $i = 0..131$;

zde **w <<< r** znamená rotaci slova **w** o **r** bitů doleva a ϕ je hexadecimální konstanta *0x9e37799b* (autoři tvrdí, že tento vzorec vylučuje vznik slabých klíčů).

Nyní na čtveřici slov (w_0, w_1, w_2, w_3) aplikujeme S-box (jako první použijeme S_3) stejným způsobem jako na obrázku 2 – vzniklá slova jsou už jednotlivá slova rundovního klíče $K[0] = (k_0, k_1, k_2, k_3)$. Další klíč $K[1] = (k_4, k_5, k_6, k_7)$ získáme aplikací boxu S_2 na (w_4, w_5, w_6, w_7) atd. (indexy u S-boxů se snižují o 1, modulo 8), až vytvoříme poslední rundovní klíč $K[32] = (k_{128}, k_{129}, k_{130}, k_{131})$. Ještě dodejme, že S-boxy jsou konstantní a byly vygenerovány tak, aby schéma co nejvíce odolávalo diferenciální a lineární kryptoanalýze (blíže viz základní dokument v infotipech).

Implementace a rychlost

Jak je zřejmé z definice zpracování klíče, rychlost šifry nezávisí na jeho délce. Dále je vidět, že rundovní klíče lze počítat za chodu. Zašifrování jednoho 128bitového bloku dat spotřebuje cca 1830 – 1940 instrukcí (je to pochopitelně závislé na typu procesoru). Navíc, díky bitově orientovanému návrhu, například 1940 instrukcí na Pentiu vyžaduje jen 1738 hodinových cyklů. Podstatné je, že na referenčním PC s 200MHz Pentiem Pro (při implementaci v jazyce C) autoři odhadují rychlost šifrování na 14,7 Mb/s. Na osmibitovém procesoru (například 3,5MHz 6805, používaném v čipových kartách) záleží na možnosti optimalizovat kód na úkor paměti. Tak například s využitím 1 KB paměti je možné dosáhnout rychlosti jen 12,8 Kb/s, zatímco s 2 KB paměti je to už 40,7 Kb/s.

Bezpečnost

Na základě pravděpodobností, vypočítaných pro potřeby diferenciální kryptoanalýzy, dospěli autoři k závěru, že 16rundovní SERPENT je stejně bezpečný jako TripleDES. Z bezpečnostních příčin však

ještě zdvojnásobili počet rund na 32, což je z hlediska dlouhodobého používání šifry jistě velmi odpovědné. Pokud jde o odolnost vůči lineární a diferenciální kryptoanalýze a metodě příbuzných klíčů, je takový dotaz trochu jako přihrávka na smeč – jeden z autorů šifry SERPENT je totiž spoluobjevitelem dvou z těchto kryptoanalytických metod...

Závěr

SERPENT je konzervativní a silně bezpečnostně orientovaná šifra. To je bohužel zaplaceno její nejnižší rychlostí v porovnání s ostatními kandidáty na AES. Vhodná tedy bude zejména pro paralelní zpracování. Procesy šifrování a odšifrování jsou odolné vůči fyzickým typům útoků.

Vlastimil Klíma (*v.klima@decros.cz*)

TWOFISH je jedním z pěti kandidátů na Advanced Encryption Standard (AES). O celém výběrovém řízení se podrobněji dozvíte v úvodu k této sérii stručných popisů všech finalistů, a to v článku “Bitva o trůn vrcholí” v Chipu 10/99; zde se už věnujeme přímo technickému popisu šifry. Připomeňme jen, že AES se stane šifrovacím standardem pro příští století (nebo alespoň nějaká ta desetiletí) a bude mít dalekosáhlý vliv na počítačovou bezpečnost.

Představujeme kandidáty na AES: Šifra TWOFISH

Blokovou šifru ***TWOFISH*** přihlásil do soutěže kolektiv šesti Američanů, z nichž čtyři patří do firmy **Conterpane Systems** Bruce Schneiera. TWOFISH má délku vstupního a výstupního bloku 128 bitů a podporuje délky klíče 128, 192 a 256 bitů. V dokumentaci se tvrdí, že používá klíčově závislé substituční tabulky (S-boxy) 8 bitů na 8 bitů. Pro přesnost musíme dodat, že ve skutečnosti vznikají kompozicí klíče a pevných substitucí 4 bity na 4 bity (tzv. tabulky t_0, t_1, t_2, t_3). Nám už známé maskování klíčem (whitening) operací $\oplus$ je použito jak na vstupu, tak na výstupu. Šifra má Feistelovo schéma s 16 rundami. Její návrh využívá různorodé operace, jako násobení prvků v Galoisově tělese GF(2^8), aritmetické sčítání, operaci $\oplus$ a substituční boxy. Výhodné je, že umožňuje výpočet rundovních klíčů za chodu (“on-the-fly”).

Postup při zašifrování

Před operací zašifrování nebo odšifrování anebo v jejím průběhu se vypočítá 40 rundovních klíčů (postup popíšeme dále). Jsou to 32bitové hodnoty K[i], i = 0..39, z nichž první čtyři se “xorují” na otevřený text a další čtyři na výsledek 16. rundy, tj. těsně před výstupem šifrového textu. V každé ze 16

rund (r = 0..15) se použijí vždy dva rundovní klíče, K[8+2r] a K[9+2r]. Vstupní 128bitový blok do rundy **r** označme jako čtyři 32bitová slova $B_{r,0}$, $B_{r,1}$, $B_{r,2}$, $B_{r,3}$. Ta se transformují na $B_{r+1,0}$, $B_{r+1,1}$, $B_{r+1,2}$, $B_{r+1,3}$ postupem podle obrázku 1.

Hlavní úlohu zde hraje funkce **g**, následovaná "pseudohadamardovou" transformací (PHT) a maskováním výstupu rundovními klíči (přičítání, v obrázku označené +, probíhá v modulu 2^{32}). Jinak zde **w <<< r** znamená rotaci slova **w** o **r** bitů doleva a **w >>> r** doprava.

Vstupem funkce **g** je 32bitové slovo neboli čtyři bajty – označme je například (x_0, x_1, x_2, x_3). Každý z nich pak prochází jemu odpovídajícím S-boxem (SBX_0 až SBX_3) a transformuje se na bajt $y_i = SBX_i(x_i)$. Výsledná čtveřice (y_0, y_1, y_2, y_3) je pak dále zpracována na čtveřici bajtů (z_0, z_1, z_2, z_3) pomocí matice MDS.

Matice MDS je typu 4 x 4 a jejími řádky jsou po řadě (hexadecimálně) konstanty (01, EF, 5B, 5B), (5B, EF, EF, 01), (EF, 5B, 01, EF) a (EF, 01, EF, 5B). Násobení prvků matice s proměnnými y_i , např. ve výrazu pro $z_0 = 01*y_0 \oplus EF*y_1 \oplus 5B*y_2 \oplus 5B*y_3$, přitom neznamená násobení bajtů, ale prvků Galoisova tělesa $GF(2^8)$ v modulu $m(x) = x^8 + x^6 + x^5 + x^3 + 1$ (definicí tohoto násobení jsme se podrobněji zabývali v článku o šifře RIJNDAEL v minulém čísle).

Touto operací, která dává už výstup funkce **g**, vlastně dojde k promíchání všech jejích 32 vstupních bitů.

Z obrázku 1 je také vidět, že jsou zde použity dvě paralelně pracující funkce **g**, které jsou opět propojeny pseudohadamardovou transformací PHT. Jedná se o zobrazení {a,b} -> { (a+b) mod 2^{32}, (a+2*b) mod 2^{32} }, které způsobuje promíchávání bitů mezi oběma paralelními větvemi. Následuje ještě maskování rundovními klíči a cyklické rotace, ale tím už je rundovní funkce úplná. Odšifrování probíhá trochu jinak než zašifrování (je popsáno v hlavním dokumentu; viz infotipy), ale hlavní hardwarové prvky lze použít i pro ně.

Příprava klíčů

Zbývá tvorba rundovních klíčů z klíče šifrovacího. Vysvětlíme ji na 128bitovém klíči – pro další dvě délky je tvorba principiálně stejná, jen mírně složitější. Jsou--li bajty šifrovacího klíče m_0, ... , m_{15}, pak definujeme 32bitová slova $M_i = (m_{4*i+0}, m_{4*i+1}, m_{4*i+2}, m_{4*i+3})$ pro i = 0, 1, 2 a 3. Dále pak pomocí nové konstantní matice RS 4 x 8 definujeme 32bitová slova $S_0 = RS*(M_0, M_1)$ a $S_1 = RS*(M_2, M_3)$, přičemž i zde se násobí prvky Galoisova tělesa, tentokrát v modulu $m(x) = x^8 + x^6 + x^3 + x^2 + 1$.

Nyní využijeme dva pevné substituční boxy Q_0 a Q_1 8 na 8 bitů, které jsou buď nadefinovány rovnou, nebo se dají "on-the-fly" počítat z menších předdefinovaných substitučních boxů (t_0 až t_4) 4 na 4 bity. K definici S-boxů využijeme klíčová slova S_0 a S_1, která na okamžik označíme jako L_0 a L_1. Definujeme $y = SBX_i(x)$ takto:

$$y = SBX_0(x) = Q_1 [Q_0 [Q_0 [x] \oplus L_{1,0}] \oplus L_{0,0}],$$

$$y = SBX_1(x) = Q_0 [Q_0 [Q_1 [x] \oplus L_{1,1}] \oplus L_{0,1}],$$

$$y = SBX_2(x) = Q_1 [Q_1 [Q_0 [x] \oplus L_{1,2}] \oplus L_{0,2}],$$

$y = SBX_3(x) = Q_0 [Q_1 [Q_1 [x] \oplus L_{1,3}] \oplus L_{0,3}]$.

S využitím podobných struktur se vypočítávají i rundovní klíče (viz obr. 2). Pokud v definici S-boxů použijeme místo slov L_0 a L_1 klíčová slova M_0 a M_2 a výsledek S-boxů ještě vynásobíme maticí MDS, obdržíme definici funkce h_0. Pokud v definici S-boxů použijeme místo slov L_0 a L_1 klíčová slova M_1 a M_3 a výsledek S-boxů vynásobíme maticí MDS, obdržíme definici funkce h_1. Kompozicí h_0 a h_1 s dalšími prvky (PHT, cyklická rotace) dostáváme definici funkce, která nám vypočítává vždy dvojici rundovních klíčů K_{8+2r} a K_{9+2r} pro r = 0..15 (obr. 2). Vstupem do funkce h_0 jsou v tomto případě čtyři stejné bajty s hodnotou 2*r (r je číslo rundy) a vstupem do funkce h_1 jsou čtyři stejné bajty s hodnotou 2*r +1.

Zbývá definovat konstantní boxy Q_0 a Q_1. Ty jsou založeny na substitucích 4 x 4 bity (t_0 a t_1). Jedno nastavení tabulek (t_0 a t_1) dává substituci Q_0 a druhé nastavení substituci Q_1. Je-li **x** vstup příslušného Q, pak jeho výstupem je hodnota **y** počítaná podle vztahů na obrázku 3. A to už je vše.

Implementace a rychlost

Plně optimalizovaná TWOFISH šifruje na referenčním 200MHz Pentiu Pro jeden blok (128 bitů) za 285 hodinových cyklů (po přípravě klíče trvající 12 700 hodinových cyklů). To dává rychlost šifrování 90 Mb/s. Při zkrácení přípravy klíče na 1250 hodinových cyklů je jeden blok možné zašifrovat za 860 hodinových cyklů. Na čipové kartě s procesorem 6805 je po přípravě klíče, trvající 1750 hodinových cyklů, možné šifrovat jeden blok (128 bitů) za 29 100 hodinových cyklů. Díky tomu, že rundovní klíče lze počítat za chodu a schéma urychlovat přípravou větších tabulek, je zde řada možností, jak schéma optimalizovat na různých procesorech s různě velkou pamětí i rychlostními nároky na přípravu klíče.

Bezpečnost

Nejúspěšnější útok, nalezený autory, je útok na pětirundovou šifru s $2^{22.5}$ volenými otevřenými texty a 2^{51} operacemi. Na základě toho autoři zvýšili počet rund na 16, což je z hlediska dlouhodobého používání šifry určitě užitečné. Autoři také potvrzují odolnost vůči všem známým útokům, zejména lineární a diferenciální kryptoanalýze.

Závěr

TWOFISH je nejen rychlá, ale i bezpečnostně orientovaná šifra. To ji staví na jedno z předních míst i mezi finalisty. Návrh umožňuje různé typy optimalizací mezi rychlostí a velikostí potřebné paměti na různých typech procesorů. Šifrování a odšifrování jsou také odolné vůči některým typům fyzických útoků. TWOFISH je proto velmi vážným kandidátem na AES.

Vlastimil Klíma (v.klima@decros.cz)