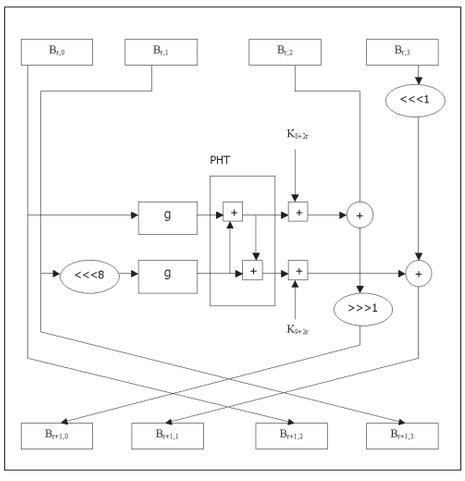
Šifrovací standard AES

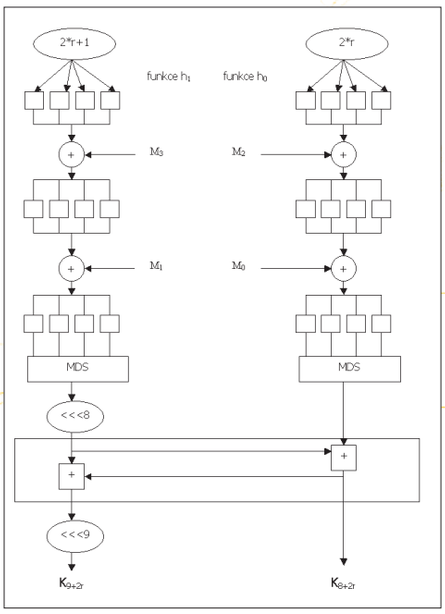
Představujeme kandidáty na AES: Šifra TWOFISH

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.

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 ⊕ 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 ⊕ a substituční boxy. Výhodné je, že umožňuje výpočet rundovních klíčů za chodu („on-the-fly").



Obr. 1. Rundovní funkce – základ TWOFISH.



Obr. 2. Výpočet rundovních klíčů.

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).

popis

Zdrojové kódy:
ftp://ftp.funet.fi/pub/crypt/cryptography/symmetric/TWOFISH/
Úplný popis:
http://csrc.nist.gov/encryption/aes/aes_home.htm

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.

Substituce Q_0 : x->y využívá tabulek

```
t0 = [ 8 1 7 D 6 F 3 2 0 B 5 9 E C A 4 ]
t1 = [ E C B 8 1 2 3 5 F 4 A 6 7 0 9 D ]
t2 = [ B A 5 E 6 D 9 0 C 8 F 3 2 4 7 1 ]
t3 = [ D 7 F 4 1 2 6 E 9 B 3 0 8 5 C A ]
```

a substituce Q_1 : x->y využívá tabulek

```
t0 = [ 2 8 B D F 7 6 E 3 1 9 4 0 A C 5 ]
t1 = [ 1 E 2 B 4 C 3 7 6 D A 5 F 9 0 8 ]
t2 = [ 4 C 7 5 1 6 9 A 0 E D 8 2 B 3 F ]
t3 = [ B 9 5 1 C 3 D E 6 4 7 F 2 0 8 A ]
```

Obě používají stejný výpočet x->y s rozdílnými tabulkami takto

```
a0 = x >> 4
b0 = x mod 16

a1 = a0 ⊕ b0
b1 = a0 ⊕ (b0 >>> 1) ⊕ (8*a0 mod 16)
a2 = t0[a1]
b2 = t1[b1]
a3 = a2 ⊕ b2
b3 = a2 ⊕ (b2 >>> 1) ⊕ (8*a2 mod 16)
a4 = t2[a3]
b4 = t3[b3]

y = 16 * b4 + a4
```

Pozn.: w>>r znamená posun slova w doprava o r bitů, přičemž zleva se doplní nuly.

Obr. 3. Definice substitucí Q_0 a Q_1.

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*)