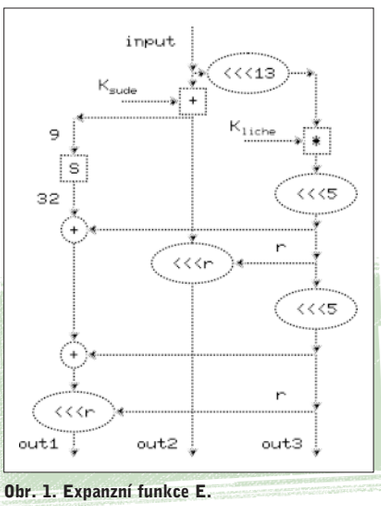
Šifrovací standard AES

Představujeme kandidáty na AES: Šifra MARS

MARS 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 pro nějaká ta desetiletí) a bude mít dalekosáhlý vliv na počítačovou bezpečnost.

Blokovou šifru ***MARS*** přihlásila do soutěže společnost **IBM a** algoritmus navrhl její jedenáctičlenný autorský kolektiv. Připomeňme, že šifra pracuje se 128bitovým vstupem a výstupem a délka jejího klíče je volitelně 16, 24 nebo 32 bajtů. MARS pracuje se slovy o 32 bitech a vychází z osvědčených kryptografických operací, které obohacuje několika novými zajímavými myšlenkami. Patří



Obr. 1. Expanzní funkce E.

k nim např. teze, že střed algoritmu má větší význam než jeho začátek a konec. To sice vypadá dost astrologicky, ale u schémat konkrétních typů to vskutku má své opodstatnění.

Jiným významným rysem je využití tzv. *Feistelova schématu typu 3* tak, že v každé rundě jedno datové slovo ze čtyř (ev. klíčový materiál) ovlivňuje zbývající tři datová slova (viz obr. 2) – to je zásadní rozdíl od častého principu, kdy se právě obdržené nejsložitější slovo okamžitě použije k modifikaci dalšího slova. Tento princip také umožnil návrhářům podpořit důkazy kvality šifry. Její další velmi podstatnou vlastností je skutečnost, že **zašifrování i odšifrování se provádí na stejném hardwaru** – obě činnosti se liší pouze v opačném řazení rundovních klíčů (jako u DES).

Postup při zašifrování

Označíme-li registry (slova) **A** a **B**, pak MARS využívá operací **A+B**, **A-B**, **A⊕B**, **A*B**, to znamená operací sčítání, odčítání, XOR a násobení slov (až na XOR vše v modulu 2^{32}), a dále cyklické rotace bitů slova **A** doleva (resp. doprava), **A<<<B** (resp. **A>>>B**), o počet bitů **r** daný pěti nejnižšími bity obsaženými v registru **B** (**r** = B AND 0x1F).

Při zašifrování se nejprve ze šifrovacího klíče (pole **k**[]) vytvoří rundovní klíče (pole **K**[]). Otevřený text se naplní do čtyř datových registrů (pole **D**[]) a potom proběhnou operace zašifrování podle pseudokódu na obrázku 2: nejprve se na data načtou první čtyři rundovní klíče K[0..3], pak proběhne dopředné mixování (bez účasti klíče), poté kryptografické jádro o 16 rundách (zde se zásadně využije 16*2 rundovních klíčů K[4..35] a funkce E, viz obr. 1), pak následuje zpětné mixování a nakonec překrytí dat rundovními klíči K[36..39] (tzv. „whitening" s operací „-").

Substituční tabulky

Ve schématu se ve fázi dopředného a zpětného mixování používá dvoukilobajtové pole **S**. Je to pevná substituční tabulka, která byla vygenerována tak,

Schéma zašifrování:

Dopředné mixování

```
for i = 0 to 3 do D[i] = D[i] + K[i]
for i = 0 to 7 do
{
  D[1] = D[1] ⊕ S0[ dolní bajt D[0] ]
  D[1] = D[1] + S1[ druhý bajt D[0] ]
  D[2] = D[2] + S0[ třetí bajt D[0] ]
  D[3] = D[3] ⊕ S1[ horní bajt D[0] ]
  D[0] >>> 24
  if i=0 or i=4 then D[0] = D[0] + D[3]
  if i=1 or i=5 then D[0] = D[0] + D[1]
  (D[3], D[2], D[1], D[0]) =  (D[0], D[3], D[2], D[1])
}
```

Klíčované jádro

```
for i = 0 to 15 do
{
  (out1, out2, out3) = E(D[0], K[4+2i], K[5+2i])
  D[0] = D[0] <<< 13
  D[2] = D[2] + out2
  if i<8 then {D[1] = D[1] + out1, D[3] = D[3] ⊕ out3}
         else {D[3] = D[3] + out1, D[1] = D[1] ⊕ out3}
  (D[3], D[2], D[1], D[0]) = (D[0], D[3], D[2], D[1])
}
```

Zpětné mixování

```
for i = 0 to 7 do
{
  if i=2 or i=6 then  D[0] = D[0] - D[3]
  if i=3 or i=7 then  D[0] = D[0] - D[1]
  D[1] = D[1] ⊕ S1[ dolní bajt D[0] ]
  D[2] = D[2] - S0[ horní bajt D[0] ]
  D[3] = D[3] - S1[ třetí bajt D[0] ]
  D[3] = D[3] ⊕ S0[ druhý bajt D[0] ]
  D[0] <<< 24
  (D[3], D[2], D[1], D[0]) = (D[0], D[3], D[2], D[1])
}
for i = 0 to 3 do D[i] = D[i] - K[36+i]
```

Obr. 2. Schéma zašifrování.

aby co nejvíce zabraňovala lineární a diferenciální kryptoanalýze. Popis její tvorby je dosti složitý a je obsažen v základním dokumentu definujícím MARS (viz infotipy). **S** je využíváno buď jako jedna tabulka „9 na 32 bitů" (tj. 2^9 32bitových položek), nebo jako dvě tabulky **S0** a **S1** „8 na 32 bitů" uložené za sebou.

infotipy

Zdrojové kódy v C, ASM:
ftp://ftp.funet.fi/pub/crypt/cryptography/symmetric/MARS/
Popis včetně inovované přípravy klíče:
http://csrc.nist.gov/encryption/aes/aes_home.htm

Zpracování klíče

Autoři akceptovali připomínku vzešlou z veřejné diskuse a změnili původní expanzi klíče. Šifrovací klíč o **n** slovech (AES vyžaduje **n** = 4, 6 a 8, MARS je definován i pro **n** = 4..14) je naplněn do pomocného pole **T** o 16 slovech. Poté se ve čtyřnásobném cyklu obsah pole **T** vždy nejprve lineárně transformuje, načež se promíchá s obsahem tabulky **S**. Část mezivýsledku se pak uloží do pole rundovních klíčů – slov K[0...39] – viz obr. 3. Po ukončení hlavního cyklu se upraví klíče K[5, 7, 9, ..., 35], které se v expanzní funkci **E** používají k násobení. Úprava je opět značně komplikovaná a jejím účelem je zabránit použití slabých klíčů.

Implementace a rychlost

Současné implementace šifry MARS v jazyce C dosahují šifrovací rychlosti 65 až 85 Mb/s (na 200MHz PC) a v hardwaru lze očekávat rychlost asi desetkrát vyšší. Pokud se MARS realizuje v 32bitovém asembleru, pak se projeví výhoda 32bitových operací a šifrování 128bitového bloku spotřebuje cca 375 hodinových cyklů. Na „smart kartách" s osmibitovým procesorem a taktem 20 MHz lze očekávat rychlost šifrování cca 500 Kb/s. Paměťové nároky představují něco přes 160 bajtů RAM (na klíč **K**) a 2 KB ROM (na **S** a na další konstanty).

Příprava klíče:

Vstupem je **n** slov šifrovacího klíče v poli **k**[ ],
výstupem je pole rundovních klíčů **K**[ ],
T[ ] je pomocné pole.

```
T[0..n-1] = k[0..n-1],  T[n] = n,  T[n+1..14] = 0

 for j=0 to 3 do
 {
   for i=0 to 14 do
   T[i] = T[i] ⊕ ((T[i-7 mod 15] ⊖ T[i-2 mod 15]) <<< 3) ⊕ (j+4*i)
   for m = 0 to 3 do
   {
 for i = 0 to 14 do  T[i] = (T[i] + S[dolních 9 bitu T[i-1 mod 15] ] ) <<< 9
   }
   for i=0 to 9 do K[10*j + i] = T[i]
 }
 dále následuje úprava klíčových slov K[5, 7, 9, ..., 35]
```

Obr. 3. Příprava klíče.

Bezpečnost

Návrháři věnovali velkou pozornost důkazům o kvalitě stavebních bloků schématu i lineární a diferenciální kryptoanalýze. Protože však schéma pro zašifrování i odšifrování (v hardwaru) je stejné, hrají zde významnou roli tzv. slabé klíče (dvojím zašifrováním se obdrží původní data). Tvorba rundovních klíčů zde sice nezaručuje, že se náhodně nevytvoří slabé klíče, ale tato pravděpodobnost je zcela mizivá. U rundovních klíčů, kterými se násobí datová slova, je zaručeno, že data nedegenerují.

Závěr

MARS je robustním algoritmem s velmi dobrým a ověřeným kryptografickým zázemím. Připomeňme jen, že IBM tuto veřejnou soutěž již před 25 lety vyhrála s algoritmem DES; MARS sice těží z kryptoanalýzy založené de facto na DES, ale oproti ní je nesrovnatelně bezpečnější.

VLASTIMIL KLÍMA (*VKLIMA@DECROS.CZ*)