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.

Představujeme kandidáty na AES:

Šifra MARS

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 k nim například 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, 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.

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).

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)

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

RIJNDAEL 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.

Šifra RIJNDAEL

Blokovou šifru ***RIJNDAEL*** přihlásili do soutěže známí kryptologové Joan Daemen a Vincent Rijmen. Ačkoliv jejich šifra podporuje i větší bloky, pro AES je délka vstupního a výstupního bloku definována jako 128 bitů. Délka klíče je volitelně 128, 192 a 256 bitů, což je **Nk** (= 4, 6 nebo 8) 32bitových slov.

RIJNDAEL je velmi flexibilní. I když jeho popis uvedeme v bajtech, lze jej elegantně zapsat i v 32bitových slovech. Návrh je přímočarý a za základ jsou použity operace v různých algebraických strukturách. Pracuje se s prvky *Galoisova tělesa* $GF(2^8)$ a s polynomy, jejichž koeficienty jsou prvky z $GF(2^8)$. Příslušné operace s nimi lze provádět buď tabulkově, nebo výpočtem přímo, což je v prvním případě výhodné pro implementaci softwarovou a v druhém případě pro hardwarovou. Bajtově orientovaný návrh také umožňuje optimalizovat programový kód pro různé mikroprocesory. Pro operace zašifrování a odšifrování sice není možné využít úplně totožný hardware (jako tomu bylo u šifry MARS), značnou část jeho prvků však použít lze.

Než přistoupíme k základním operacím, vysvětlíme si nejnutnější pojmy. Prvky v Galoisově tělese $GF(2^8)$ mají osm bitů $(b_7, ..., b_0)$, nereprezentují však bajty, nýbrž polynomy $(b_7x^7 + ... + b_1x^1 + b_0)$. Násobení těchto prvků je proto zavedeno nikoli jako násobení bajtů, ale jako násobení jim odpovídajících polynomů, a to modulo $m(x) = x^8 + x^4 + x^3 + x^1 + 1$.

Takže například ´57´ (v apostrofech píšeme běžné hexadecimální vyjádření bitů b_7, ..., b_0) krát ´83´ je rovno ´C1´, neboť

$(x^6 + x^4 + x^2 + x^1 + 1) * (x^7 + x^1 + 1) = = (x^7 + x^6 + 1) \bmod m(x)$.

Postup při zašifrování

RIJNDAEL pracuje v rundách. Jejich počet **Nr** = 10, 12 a 14 je určen podle toho, jak dlouhý je šifrovací klíč, a odpovídá hodnotám **Nk** = 4, 6 a 8. Pro delší klíč se tedy použije více rund. Před operací zašifrování (nebo v jejím průběhu, tzv. “on-the-fly”) se vypočítá 4 + Nr*4 rundovních klíčů (32bitových slov). První čtyři se “naxorují” na otevřený text (tzv. “whitening”). Potom proběhne **Nr** rund a v každé z nich se použijí 4 rundovní klíče. Na počátku se 16 bajtů otevřeného textu naplní postupně po sloupcích (tj. shora dolů a zleva doprava) do matice bajtů **A** = $(a_{ij})_{i=0..3,\ j=0..3}$ a na ně se ve stejném pořadí postupně “naxoruje” 16 bajtů tvořících první čtyři rundovní klíče.

Poté proběhne **Nr** rund podle pseudokódu na obr. 1, kde “State” znamená stav matice **A**. Připomeňme, že prvky matice **A** jsou sice bajty, ale při násobení jsou chápány jako prvky $GF(2^8)$. “Sčítání” těchto prvků (při operaci *MixColumn*) je běžná operace XOR. Výsledný šifrový text se opět vybírá po sloupcích z matice **A**.

Hlavní transformace

Všechny rundy jsou stejné, až na poslední, kde je malá změna – neprovádí se operace mixování MixColumn. Nyní k jednotlivým operacím z obrázku 1:

ByteSub je bajtová substituce (a→b), kterou aplikujeme na každý bajt $\mathbf{a}_{i,j}$ matice **A**. Nejprve vypočteme multiplikativní inverzi prvku **a**, tj. **c** = $\mathbf{a}^{-1}$ mod m(x), a poté bajt **c** transformujeme na **b** substitucí **S** podle obr. 1. Substituci nemusíme počítat podle tohoto vzorce, ale můžeme si ji uložit jako pevnou tabulku.

ShiftRow vykoná v matici **A** cyklickou rotaci jejích prvků v jednotlivých řádcích doleva, a to tak, že první řádek ponechá beze změny, druhý rotuje o jednu pozici, třetí o dvě a čtvrtý o tři pozice.

MixColumn zesložití prvky v rámci každého sloupce matice **A**. Vstupem této transformace jsou všechny prvky daného sloupce (na obrázku je označen **a**) a výstupem jejich nové hodnoty (**b**). Tak bude například b_0 = ´02´*a_0 ⊕ ´03´*a_1 ⊕ ´01´*a_2 ⊕ ´01´*a_3.

Nakonec se operací *AddRoundKey* na prvky matice **A** (opět po sloupcích) “naxorují” po řadě jednotlivé bajty čtyř rundovních klíčů, které jsou na řadě. A to je celé.

Odšifrování probíhá trochu jinak než zašifrování, ale využívá jeho stavební prvky (popis je uveden v hlavním dokumentu popisujícím šifru; viz infotipy). Zbývá popsat výpočet rundovních klíčů ze šifrovacího klíče.

Zpracování klíče

Šifrovací klíč **key** (viz obr. 2) o **Nk** 32bitových slovech (4, 6 nebo 8) se naplní na počátek pomocného pole 32bitových slov **W**[0 ... Nk-1]. Toto pole se poté expanduje tak, že každé nové W je

vypočítáno jako W[i] = W[i - Nk] ⊕ temp, kde **temp** je W[i - 1] nebo jeho modifikace – viz obrázek 2. Při modifikaci se využívá operace cyklického posuvu bajtů slova **temp** o jeden doprava (*RotByte*), dále nám známé substituce bajtů *SubByte*, a to aplikované na každý bajt proměnné **temp**, a pole konstant **Const**[].

Implementace a rychlost

Dnešní implementace šifry RIJNDAEL v jazyce C na referenčním PC s Pentiem Pro 200MHz dosahují rychlosti šifrování cca 70/60/50 Mb/s při délkách klíče 128/192/256 bitů. Rychlost šifrování měřená počtem cyklů na jeden 128bitový blok je 363/432/500 cyklů (pro tytéž délky klíče); jde tedy zhruba o 3 – 5 cyklů na jeden bit. Na osmibitovém procesoru Intel 8051 trvá zašifrování jednoho bloku cca 3000 – 5000 cyklů (1 cyklus = 12 period oscilátoru) a na čipu Motorola 68HC08 (1 cyklus = 1 perioda oscilátoru) je to cca 8000 – 12 000 cyklů. Spotřeba paměti RAM je pouhých 52 bajtů (!), neboť u obou těchto implementací byly rundovní klíče počítány on-the-fly. Délka kódu je v obou případech do 1 KB. Odšifrování trvá vždy cca o 30 % déle než zašifrování.

Bezpečnost

Oba autoři dokazují skvělé vlastnosti stavebních bloků schématu i odolnost vůči lineární a diferenciální kryptoanalýze. Protože schéma pro zašifrování i odšifrování (v hardwaru) se liší, není tu riziko slabých klíčů. Ekvivalenci klíčů (což je případ, kdy různé šifrovací klíče dávají stejné sady rundovních klíčů) brání podle autorů nelineární expanze.

Závěr

U šifry RIJNDAEL je ceněn její průzračný návrh, založený na různých algebraických operacích. Šifra je flexibilní při realizaci na různých typech procesorů s velmi malými nároky na paměť i velikost kódu, a přitom vykazuje ještě dostatečnou rychlost. Je vhodná i pro paralelní zpracování a je odolná vůči fyzickým typům útoků. Z mého pohledu jsou však navržené stavební prvky i jejich kompozice poměrně nové a osobně bych byl překvapen, kdyby RIJNDAEL zvítězil.

Vlastimil Klíma (vklima@decros.cz)

Infotipy:

Zdrojové kódy:

ftp://ftp.funet.fi/pub/crypt/

cryptography/symmetric/rijndael/

Úplný popis:

http://csrc.nist.gov/encryption/aes/aes_home.htm