1 AES ŠTANDARD – BLOKOVÁ ŠIFRA RIJNDAEL

1.1 ÚVOD

Americký úrad (National Institute of Standards and Technology – NIST) vyhlásil v roku 1997 verejnú súťaž na výber nového symetrického blokového algoritmu na ochranu citlivých údajov [1]. Cieľom bolo vybrať blokový šifrovací algoritmus, ktorý by nahradil dnes už málo bezpečný algoritmus DES (Data Encryption Standard) [2]. V roku 1998 NIST ohlásil predbežný výber 15 kandidátov a vyzval kryptografickú komunitu na ich analýzu. Táto analýza zahrňovala analýzu bezpečnostných ako aj rýchlostných a implementačných parametrov jednotlivých kandidátov. Na základe výsledkov tejto predbežnej analýzy NIST vybral v lete 1999 5 finalistov – algoritmy MARS, RC6, Rijndael, Serpent a Twofish. Na základe ďalšej verejnej analýzy NIST 2. októbra 2000 ohlásil víťaza celého výberového konania – algoritmus Rijndael. Dôvody výberu a porovnanie s parametrami ostatných finalistov NIST zverejnil vo verejne dostupnej správe [3].

NIST v novembri 2001 zverejnil nový štandard (Advanced Encryption Standard - AES) ako oficiálny dokument FIPS PUB 197 [5]. Je teda možné konštatovať, že výber AES prebiehal za kvalitatívne úplne odlišných podmienok ako výber jeho predchodcu – DESu. Dôvody výberu jednotlivých blokov algoritmu DES neboli dodnes zverejnené a predstavujú tak stále určitý zdroj nedôvery k algoritmu DES. Očakáva sa, že AES by sa mal stať najpoužívanejším symetrickým blokovým šifrovacím algoritmom.

1.2 ALGORITMUS RIJNDAEL

Algoritmus Rijndael [4] prihlásili do súťaže dvaja Belgičania – Joan Daemen a Vincent Rijmen. Materiál [4] obsahuje úplný opis šifry, základné matematické operácie použité v šifre, dôvody výberu jednotlivých parametrov ako aj možné spôsoby optimalizácie na rôznych platformách (8-bitové, 32-bitové procesory). Stručný opis algoritmu Rijndael je možné nájsť aj v [6]. V rámci cvičenia opíšeme základné operácie, vlastnosti a štruktúru algoritmu pre šifrovanie a expanziu kľúča, ďalšie podrobnosti je možné nájsť v [4],[5].

1.2.1 ZÁKLADNÉ OPERÁCIE V TELESE $GF\left(2^8\right)$

Galoisove teleso (Galois Field) $GF\left(2^8\right)$ je ***konečné teleso*** (pojem zavedený v algebre a diskrétnej matematike), ktoré má 256 prvkov. Konečné teleso je množina

prvkov, ktoré nazývame čísla alebo prvky poľa[1] $A \in GF\left(2^8\right)$, spolu s definíciou dvoch operácii nazývaných "***sčítanie*** +" a "***násobenie*** $\bullet$", pričom tieto operácie splňujú základné požiadavky akými sú ***komutatívny, distributívny a asociatívny zákon***, ***existencia*** prvku 0 a 1, ktoré splňujú podmienky $A+0=A$, $1\bullet A=A$, existenciu ***inverzného prvku*** $A\bullet A^{-1}=1$ pre $A\neq 0$.
Prvky $A \in GF\left(2^8\right)$ je možné reprezentovať viacerými spôsobmi. V [4] bola využitá reprezentácia pomocou polynómov. 256 prvkov $GF\left(2^8\right)$ je možné reprezentovať v jednom bajte. Nech prvky $A, B \in GF\left(2^8\right)$ sú reprezentované v tvare

$$\mathbf{a}=\left(a_7,a_6,\ldots,a_0\right), \qquad A \leftrightarrow a\left(X\right)=\sum_{i=0}^{7} a_i X^i \tag{1.1}$$

$$\mathbf{b}=\left(b_7,b_6,\ldots,b_0\right), \qquad B \leftrightarrow b\left(X\right)=\sum_{i=0}^{7} b_i X^i \tag{1.2}$$

Aj keď ďalej opísané matematické operácie v $GF\left(2^8\right)$ sú neštandardné, sú technicky ľahko realizovateľné ako programovo tak aj pomocou špecializovaných technických prostriedkov (napr. hradlových polí, obvodov ASIC, ...).

1.2.1.1 SČÍTANIE

Sčítanie je v $GF\left(2^8\right)$ veľmi jednoduché a realizuje sa sčítaním po zložkách v modulo-2 aritmetike podľa vzťahov

$$\begin{aligned} 0+0&=0 & 0.0&=0 \\ 0+1&=1 & 1.0&=0 \\ 1+0&=1 & 0.1&=0 \\ 1+1&=0 & 1.1&=1 \end{aligned} \tag{1.3}$$

Príklad 1
Ukážte, že v $GF\left(2^8\right)$ *platí*

0x57+0x83=0xD4

pričom zápis 0x.. *znamená hexadecimálny zápis bajtu.*

1.2.1.2 NÁSOBENIE

Násobenie v $GF\left(2^8\right)$ je definované zložitejším predpisom. Pre polynomiálnu reprezentáciu (1.1)-(1.2) je násobenie realizované ako násobenie polynómov $a\left(X\right)$ a $b\left(X\right)$ modulo ireducibilný binárny polynóm stupňa 8. V [4] je **ireducibilný polynóm**

[1] Veľkými písmenami budeme v tomto texte označovať prvky z telesa $A \in GF\left(2^8\right)$, t.j. prvky, ktoré je potrebné reprezentovať viacbitovými hodnotami. Malými písmenami budeme označovať prvky z telesa $a \in GF\left(2\right)$, t.j. prvky ktoré môžu nadobúdať hodnoty 0 a 1. Platí teda $A \leftrightarrow \mathbf{a}=\left(a_7,a_6,\ldots,a_0\right)$, pričom **a** reprezentuje vektor bitov.

stupňa 8 **nad telesom** $GF(2)$ (t.j. má len koeficienty 0 a 1) označený ako $m(X)$ a je definovaný v tvare

$$\mathbf{m} = (1,0,0,0,1,1,0,1,1), \qquad m(X) = X^8 + X^4 + X^3 + X + 1 \tag{1.4}$$

Násobenie dvoch prvkov $A, B \in GF(2^8)$ je definované takto:

$$A \bullet B \leftrightarrow (\mathbf{a} \bullet \mathbf{b})_{GF(2^8)} = a(X)b(X) \bmod m(X) \tag{1.5}$$

Príklad 2
Ukážte, že v $GF(2^8)$ *s polynómom* $m(X)$ *(1.4) platí* 0x57•0x83=0xC1.

Je zrejmé, že počítanie súčinu $GF(2^8)$ pomocou vzťahu (1.5) je relatívne komplikované a napr. pre softvérovú realizáciu neefektívne. V prílohe je uvedený postup, ktorý umožňuje pomocou dvoch tabuliek realizovať násobenie v $GF(2^8)$ veľmi jednoduchým spôsobom. Tento spôsob realizácie je typicky využívaný v softvérových implementáciách, ktoré majú k dispozícií dostatočnú pamäť.

V implementáciách algoritmu AES pomocou 8-bitových mikroprocesorov, ktoré sú často využívané v inteligentných kartách (smart cards) je výhodné[2] použiť funkciu **b=xtime(a)**. Táto funkcia realizuje výpočet

$$\mathbf{b} = (2 \bullet \mathbf{a})_{GF(2^8)} = Xb(X) \bmod m(X) \tag{1.6}$$

Vzťah (1.6) realizuje násobenie prvku **a** konštantným prvkom $(2)_{GF(2^8)}$. Tento výpočet je možné realizovať pomocou inštrukcie rotácie vľavo a podmieneného XOR súčtu s prvkom 0x11B[3] v prípade, že po rotovaní je výsledok väčší ako 255.

1.2.1.3 POLYNÓMY S KOEFICIENTMI Z TELESA $GF(2^8)$

Podobne ako boli definované binárne polynómy vo vzťahoch (1.1)-(1.2), je možné definovať aj polynómy nad telesom $GF(2^8)$ (t.j. s koeficientmi ktoré sú prvkami z telesa $GF(2^8)$). Takto je možné priradiť 4-bajtovému vektoru polynóm menšieho stupňa ako 4. Takto definované polynómy je možné sčítavať tak, že sa zodpovedajúce koeficienty sčítajú v $GF(2^8)$, čo je možné realizovať sčítaním jednotlivých bitov modulo-2.

Násobenie je opäť podstatne komplikovanejšie. Uvažujme dva polynómy $A(X)$, $B(X)$ definované nad telesom $GF(2^8)$

[2] Algoritmus Rijndael využíva len násobenia dvoch prvkov v $GF(2^8)$, pričom jeden z nich je vždy konštantný a má pomerne malú hodnotu. Táto skutočnosť umožňuje využiť funkciu xtime() pomocou jednoduchého rozkladu ako napr. $\mathbf{b} = xtime(9 \bullet \mathbf{a}) = xtime((8+1) \bullet \mathbf{a}) = xtime(xtime(xtime(\mathbf{a}))) + \mathbf{a}$.

[3] Samozrejme v prípade reálnej implementácie pomocou procesora je možné s výhodou využívať rotáciu do príznakového registra Carry a podmienené vykonanie XOR operácie s 8-bitovým prvkom 0x1B.

$$A(X) = A_3X^3 + A_2X^2 + A_1X + A_0 \tag{1.7}$$

$$B(X) = B_3X^3 + B_2X^2 + B_1X + B_0 \tag{1.8}$$

Pre ich súčin platí

$$C(X) = A(X) \bullet B(X) = C_6X^6 + C_5X^5 + C_4X^4 + C_3X^3 + C_2X^2 + C_1X + C_0 \tag{1.9}$$

pričom

$$\begin{aligned}
C_0 &= A_0 \bullet B_0 \\
C_1 &= A_1 \bullet B_0 \oplus A_0 \bullet B_1 \\
C_2 &= A_2 \bullet B_0 \oplus A_1 \bullet B_1 \oplus A_0 \bullet B_2 \\
C_3 &= A_3 \bullet B_0 \oplus A_2 \bullet B_1 \oplus A_1 \bullet B_2 \oplus A_0 \bullet B_3 \\
C_4 &= A_3 \bullet B_1 \oplus A_2 \bullet B_2 \oplus A_1 \bullet B_3 \\
C_5 &= A_3 \bullet B_2 \oplus A_2 \bullet B_3 \\
C_6 &= A_3 \bullet B_3
\end{aligned} \tag{1.10}$$

Výsledok súčinu $C(X)$ vo všeobecnom prípade nie je možné reprezentovať pomocou 4 bajtov. Redukciou polynómu $C(X)$ vo vzťahu (1.9) pomocou modulo operácie s polynómom stupňa 4 je možné výsledok reprezentovať pomocou 4 bajtov. Algoritmus Rijndael používa polynóm $M(X) = X^4 + 1$ pre ktorý platí

$$X^j \bmod \left(X^4 + 1\right) = X^{j \bmod 4} \tag{1.11}$$

Modulárny súčin $A(X)$ a $B(X)$ označený ako $D(X) = A(X) \otimes B(X)$ je rovný

$$D(X) = A(X) \bullet B(X) \bmod M(X) = D_3X^3 + D_2X^2 + D_1X + D_0 \tag{1.12}$$

pričom

$$\begin{aligned}
D_0 &= A_0 \bullet B_0 \oplus A_3 \bullet B_1 \oplus A_2 \bullet B_2 \oplus A_1 \bullet B_3 \\
D_1 &= A_1 \bullet B_0 \oplus A_0 \bullet B_1 \oplus A_3 \bullet B_2 \oplus A_2 \bullet B_3 \\
D_2 &= A_2 \bullet B_0 \oplus A_1 \bullet B_1 \oplus A_0 \bullet B_2 \oplus A_3 \bullet B_3 \\
D_3 &= A_3 \bullet B_0 \oplus A_2 \bullet B_1 \oplus A_1 \bullet B_2 \oplus A_0 \bullet B_3
\end{aligned} \tag{1.13}$$

Operáciu násobenia (1.12) je možné zapísať aj v maticovom tvare

$$\begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} A_0 & A_3 & A_2 & A_1 \\ A_1 & A_0 & A_3 & A_2 \\ A_2 & A_1 & A_0 & A_3 \\ A_3 & A_2 & A_1 & A_0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix} = \mathbf{A} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix} \tag{1.14}$$

pričom algoritmus Rijndael využíva pri šifrovaní násobenie vhodne vybranou konštantnou[4] maticou. V prípade dešifrovania je použitá jej inverzia.

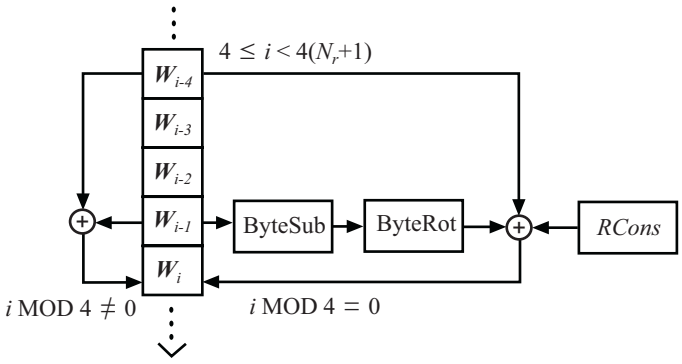
1.2.2 ŠTRUKTÚRA ŠIFRY

Šifra Rijndael je iteračná bloková šifra (t.j. používa N_r opakujúcich sa rúnd) s voliteľnou veľkosťou bloku[5] N_b a voliteľnou dĺžkou kľúča N_k. Hodnoty N_b a N_k môžu byť nastavené nezávisle na hodnoty 128, 192 a 256 bitov, pričom súvislosť jednotlivých parametrov ukazuje nasledujúca tabuľka.

Počet rúnd N_r ako funkcia dĺžky bloku N_b (ako násobok 32 bitov) a dĺžky kľúča N_k (ako násobok 32 bitov)

N_r	$N_b = 4$	$N_b = 6$	$N_b = 8$
$N_k = 4$	10	12	14
$N_k = 6$	12	12	14
$N_k = 8$	14	14	14

Jednotlivé bloky šifry pracujú s údajmi (medzivýsledkami), nazývanými ***Stav***. Stav je možné reprezentovať ako obdĺžnikovú maticu, ktorá má 4 riadky a N_b stĺpcov. Pred začiatkom šifrovania (prípadne aj počas samotného šifrovania tzv. metódou „on the fly" ktorej princíp je znázornený na Obr.1) sa z $32 \times N_k$-bitového kľúča vypočítajú 32-bitové tzv. rundové kľúče (RoundKeys), ktorých je $N_b + N_b N_r$.



Obr.1 Princíp „on the fly" výpočtu rundových kľúčov pričom ByteSub a ByteRot sú špeciálne operácie opísané v ďalšej časti a RCons sú preddefinované konštanty.

[4] Prvky tejto matice boli vyberané tak, aby umožnili efektívnu technickú realizáciu a preto majú len malé hodnoty – $(1)_{GF(2^8)}$, $(2)_{GF(2^8)}$, $(3)_{GF(2^8)}$.

[5] Norma AES využíva len dĺžku bloku $N_b = 4$, t.j. 128 bitov. Dĺžky blokov 196 a 256 bitov nie sú podporované.

Prvých N_b rundových kľúčov sa „naXORruje" s otvoreným textom ($32 \times N_b$ vstupných bitov, t.j. $4 \times N_b$ bajtov) a uložia sa do premennej ***Stav***, ktorá je tvorená maticou **A** s rozmermi $4 \times N_b$ (matica **A** sa naplňuje po stĺpcoch, t.j. zhora dole a zľava doprava). Potom sa vykoná N_r rúnd podľa nasledujúceho pseudokódu v jazyku C:

```
Round ( State, RoundKey ) {
        ByteSub ( State );
        ShiftRow ( State );
        MixColumn ( State );                    // nevykonava sa v poslednej runde!
        AddRoundKey ( State, RoundKey );
}
```

pričom ***Stav*** je reprezentovaný maticou

$$\mathbf{A} = \begin{bmatrix} A_{00} & A_{01} & A_{02} & \dots & \dots & A_{0N_b-1} \\ A_{10} & A_{11} & A_{12} & \dots & \dots & A_{1N_b-1} \\ A_{21} & A_{22} & A_{23} & \dots & \dots & A_{2N_b-1} \\ A_{31} & A_{32} & A_{33} & \dots & \dots & A_{3N_b-1} \end{bmatrix} \tag{1.15}$$

1.2.2.1 OPERÁCIA BYTESUB

Transformácia ***ByteSub*** je nelineárna bajtová substitúcia, realizovaná nezávisle na všetkých bajtoch matice (1.15). Substitučná tabuľka (tzv. S-box) je invertibilná transformácia skladajúca sa z dvoch transformácií:

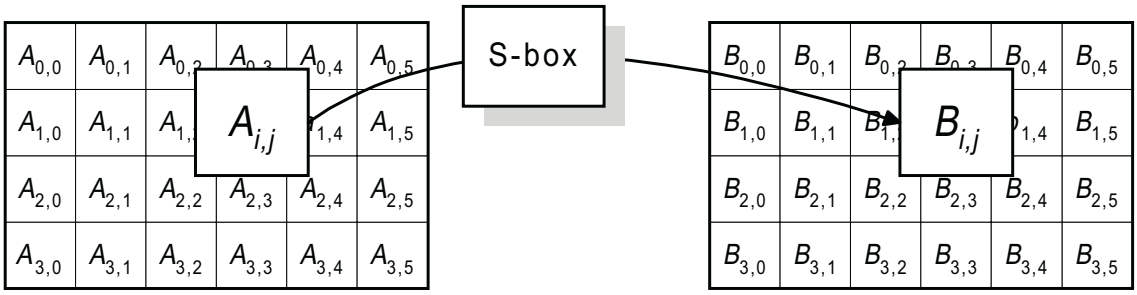
1. pre hodnotu $A_{ij} \in GF\left(2^8\right)$, $A_{ij} \neq 0$ určíme multiplikatívne inverzné číslo $X = A_{ij}^{-1}$, pre ktoré platí $A_{ij} \bullet X = 1$, pričom je použitá reprezentácia (1.4). Hodnota $A_{ij} = 0$ je mapovaná na hodnotu 0.
2. Hodnota $X \leftrightarrow \left(x_7, x_6, \dots, x_0\right)$ je transformovaná affinou transformáciou (nad telesom $GF\left(2\right)$) podľa vzťahu

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{1.16}$$

Použitie opísaného S-boxu na všetky bajty premennej stav je označené ako

```
ByteSub ( State )
```

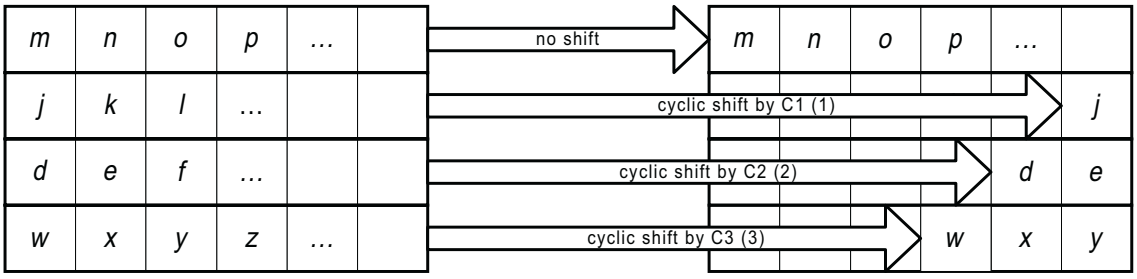
a je znázornené na Obr.2.



Obr.2 Operácia ByteSub

1.2.3 OPERÁCIA SHIFTROW

Operácia ***ShiftRow*** realizuje cyklický posuv jednotlivých riadkov matice ***Stav***, pričom riadok 0 je neposunutý, riadok 1 sa posúva o C1 bajtov, riadok 2 o C2 bajtov a riadok 3 o C3 bajtov, čo je znázornené na Obr.3



Obr.3 Operácia ShiftRow

a označované ako

ShiftRow (State)

Posuny C1,C2,C3 závisia na hodnote N_b , čo je dokumentované v nasledujúcej tabuľke.

Hodnoty posunov pre rôzne hodnoty N_b

N_b	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

1.2.3.1 OPERÁCIA MIXCOLUMN

Operácia ***MixColumn*** spracováva jednotlivé stĺpce matice ***Stav*** (interpretované ako koeficienty polynómu nad telesom $GF\left(2^8\right)$) pomocou súčinu $\otimes$ (definovanom vzťahmi (1.12)-(1.14)) s polynómom

$$C\left(X\right) = 0\text{x}03X^3 + 0\text{x}01X^2 + 0\text{x}01X + 0\text{x}02 \tag{1.17}$$

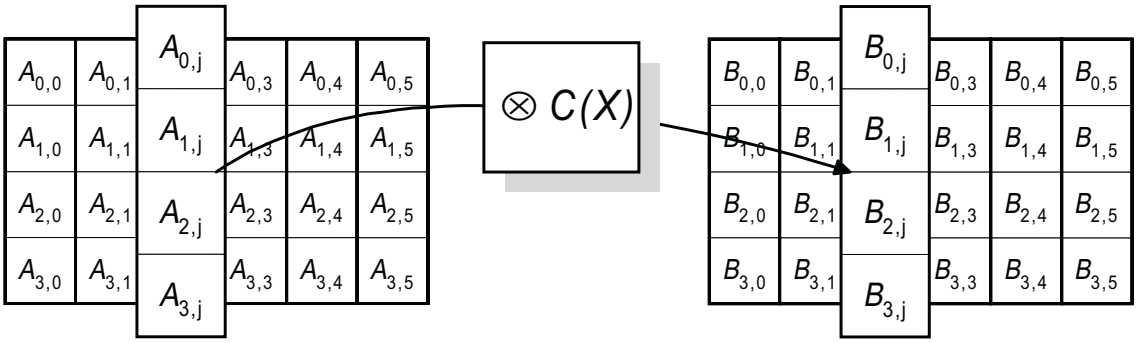
čo je možné reprezentovať maticovým zápisom

$$\begin{bmatrix} B_{0j} \\ B_{1j} \\ B_{2j} \\ B_{3j} \end{bmatrix} = \begin{bmatrix} 0\text{x}02 & 0\text{x}03 & 0\text{x}01 & 0\text{x}01 \\ 0\text{x}01 & 0\text{x}02 & 0\text{x}03 & 0\text{x}01 \\ 0\text{x}01 & 0\text{x}01 & 0\text{x}02 & 0\text{x}03 \\ 0\text{x}03 & 0\text{x}01 & 0\text{x}01 & 0\text{x}02 \end{bmatrix} \begin{bmatrix} A_{0j} \\ A_{1j} \\ A_{2j} \\ A_{3j} \end{bmatrix} \tag{1.18}$$

Operácia

```
MixColumn ( State )
```

je znázornená na Obr.4.



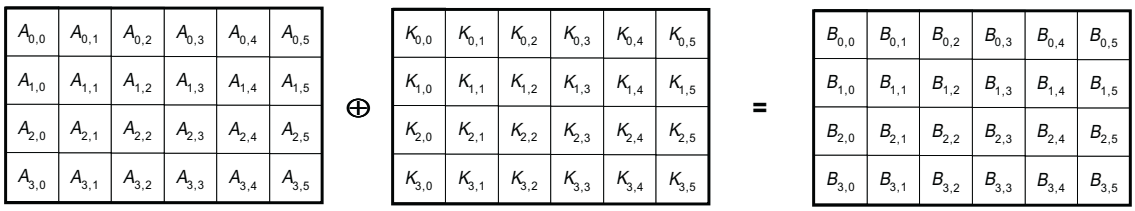
Obr.4 Operácia MixColumn

1.2.3.2 OPERÁCIA ADDROUNDKEY

Táto operácia XORuje príslušný rundový kľúč a jednotlivé bajty matice ***Stav***, čo je znázornené na Obr.5 a označovaná ako

```
AddRoundKey ( State,  RoundKey )
```

pričom dĺžka rundového kľúča **K** je zhodná s dĺžkou bloku.



Obr.5 Operácia AddRoundKey

1.3 ZHRNUTIE

V rámci cvičenia sme prebrali základné bloky a matematický aparát šifry Rijndael. Ďalšie podrobnosti (napr. inverzná šifra, expanzia kľúča na rundové kľúče, optimalizácia pre rôzne architektúry, ...) je možné nájsť v [4],[5]. Veľmi pekná demonštrácia jednotlivých blokov šifry AES v Matlabe je v [7]. Na záver je možné konštatovať, že použité stavebné bloky šifry Rijndael sú relatívne nové a výrazne odlišné od blokov šifry DES.

LITERATÚRA

[1] Advanced Encryption Standard, http//www.nist.gov/aes

[2] Data Encryption Standard - FIBS PUB 46-3, Federal Information Processing Standards Publications, Reaffirmed 1999 October 25. U.S. Department of Commerce/National Institute of Standards and Technology. Dostupné v elektronickej forme **– FIPS46-3.pdf**.

[3] J. Nechvastal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, E. Roback, "Report on the Development of the Advanced Encryption Standard (AES)", U.S. Department of Commerce/National Institute of Standards and Technology, October 2, 2000, pp.1-116, available at [1]. Dostupné v elektronickej forme **– AES_r2report.pdf**.

[4] J. Daemen, V. Rijmen, "The Rijndael Block Cipher", AES Proposal, Version 2, September 1999, http://www.nist.gov/aes. Dostupné v elektronickej forme **– Rijndael.pdf**.

[5] Advanced Encryption Standard (AES) - FIBS PUB 197, Federal Information Processing Standards Publications, November 26, 2001 U.S. Department of Commerce/National Institute of Standards and Technology. Dostupné v elektronickej forme **– FIPS-197.pdf**.

[6] V. Klíma, "Šifrovací standard AES: Šifra Rijndael", CHIP 11/1999, s.64-65.

[7] J.J. Buchholz, "Matlab implementation of the Advanced Encryption Standard", http://buchholz.hs-bremen.de/aes/aes.htm. Dostupné v elektronickej forme - **aes.pdf**, **aes.zip**.

PRÍLOHA

Optimalizované zdrojové kódy v jazyku C využívajú (aj) nasledujúce tabuľky:

```
word8 Logtable[256] = {
  0,   0,  25,   1,  50,   2,  26, 198,  75, 199,  27, 104,  51, 238, 223,   3,
100,   4, 224,  14,  52, 141, 129, 239,  76, 113,   8, 200, 248, 105,  28, 193,
125, 194,  29, 181, 249, 185,  39, 106,  77, 228, 166, 114, 154, 201,   9, 120,
101,  47, 138,   5,  33,  15, 225,  36,  18, 240, 130,  69,  53, 147, 218, 142,
150, 143, 219, 189,  54, 208, 206, 148,  19,  92, 210, 241,  64,  70, 131,  56,
102, 221, 253,  48, 191,   6, 139,  98, 179,  37, 226, 152,  34, 136, 145,  16,
126, 110,  72, 195, 163, 182,  30,  66,  58, 107,  40,  84, 250, 133,  61, 186,
 43, 121,  10,  21, 155, 159,  94, 202,  78, 212, 172, 229, 243, 115, 167,  87,
175,  88, 168,  80, 244, 234, 214, 116,  79, 174, 233, 213, 231, 230, 173, 232,
 44, 215, 117, 122, 235,  22,  11, 245,  89, 203,  95, 176, 156, 169,  81, 160,
127,  12, 246, 111,  23, 196,  73, 236, 216,  67,  31,  45, 164, 118, 123, 183,
204, 187,  62,  90, 251,  96, 177, 134,  59,  82, 161, 108, 170,  85,  41, 157,
151, 178, 135, 144,  97, 190, 220, 252, 188, 149, 207, 205,  55,  63,  91, 209,
 83,  57, 132,  60,  65, 162, 109,  71,  20,  42, 158,  93,  86, 242, 211, 171,
 68,  17, 146, 217,  35,  32,  46, 137, 180, 124, 184,  38, 119, 153, 227, 165,
103,  74, 237, 222, 197,  49, 254,  24,  13,  99, 140, 128, 192, 247, 112,   7
};

word8 Alogtable[256] = {
  1,   3,   5,  15,  17,  51,  85, 255,  26,  46, 114, 150, 161, 248,  19,  53,
 95, 225,  56,  72, 216, 115, 149, 164, 247,   2,   6,  10,  30,  34, 102, 170,
229,  52,  92, 228,  55,  89, 235,  38, 106, 190, 217, 112, 144, 171, 230,  49,
 83, 245,   4,  12,  20,  60,  68, 204,  79, 209, 104, 184, 211, 110, 178, 205,
 76, 212, 103, 169, 224,  59,  77, 215,  98, 166, 241,   8,  24,  40, 120, 136,
131, 158, 185, 208, 107, 189, 220, 127, 129, 152, 179, 206,  73, 219, 118, 154,
181, 196,  87, 249,  16,  48,  80, 240,  11,  29,  39, 105, 187, 214,  97, 163,
254,  25,  43, 125, 135, 146, 173, 236,  47, 113, 147, 174, 233,  32,  96, 160,
251,  22,  58,  78, 210, 109, 183, 194,  93, 231,  50,  86, 250,  21,  63,  65,
195,  94, 226,  61,  71, 201,  64, 192,  91, 237,  44, 116, 156, 191, 218, 117,
159, 186, 213, 100, 172, 239,  42, 126, 130, 157, 188, 223, 122, 142, 137, 128,
155, 182, 193,  88, 232,  35, 101, 175, 234,  37, 111, 177, 200,  67, 197,  84,
252,  31,  33,  99, 165, 244,   7,   9,  27,  45, 119, 153, 176, 203,  70, 202,
 69, 207,  74, 222, 121, 139, 134, 145, 168, 227,  62,  66, 198,  81, 243,  14,
 18,  54,  90, 238,  41, 123, 141, 140, 143, 138, 133, 148, 167, 242,  13,  23,
 57,  75, 221, 124, 132, 151, 162, 253,  28,  36, 108, 180, 199,  82, 246,   1
};
```

```
word8 S[256] = {
 99, 124, 119, 123, 242, 107, 111, 197,  48,   1, 103,  43, 254, 215, 171, 118,
202, 130, 201, 125, 250,  89,  71, 240, 173, 212, 162, 175, 156, 164, 114, 192,
183, 253, 147,  38,  54,  63, 247, 204,  52, 165, 229, 241, 113, 216,  49,  21,
  4, 199,  35, 195,  24, 150,   5, 154,   7,  18, 128, 226, 235,  39, 178, 117,
  9, 131,  44,  26,  27, 110,  90, 160,  82,  59, 214, 179,  41, 227,  47, 132,
 83, 209,   0, 237,  32, 252, 177,  91, 106, 203, 190,  57,  74,  76,  88, 207,
208, 239, 170, 251,  67,  77,  51, 133,  69, 249,   2, 127,  80,  60, 159, 168,
 81, 163,  64, 143, 146, 157,  56, 245, 188, 182, 218,  33,  16, 255, 243, 210,
205,  12,  19, 236,  95, 151,  68,  23, 196, 167, 126,  61, 100,  93,  25, 115,
 96, 129,  79, 220,  34,  42, 144, 136,  70, 238, 184,  20, 222,  94,  11, 219,
224,  50,  58,  10,  73,   6,  36,  92, 194, 211, 172,  98, 145, 149, 228, 121,
231, 200,  55, 109, 141, 213,  78, 169, 108,  86, 244, 234, 101, 122, 174,   8,
186, 120,  37,  46,  28, 166, 180, 198, 232, 221, 116,  31,  75, 189, 139, 138,
112,  62, 181, 102,  72,   3, 246,  14,  97,  53,  87, 185, 134, 193,  29, 158,
225, 248, 152,  17, 105, 217, 142, 148, 155,  30, 135, 233, 206,  85,  40, 223,
140, 161, 137,  13, 191, 230,  66, 104,  65, 153,  45,  15, 176,  84, 187,  22
};

word8 Si[256] = {
 82,   9, 106, 213,  48,  54, 165,  56, 191,  64, 163, 158, 129, 243, 215, 251,
124, 227,  57, 130, 155,  47, 255, 135,  52, 142,  67,  68, 196, 222, 233, 203,
 84, 123, 148,  50, 166, 194,  35,  61, 238,  76, 149,  11,  66, 250, 195,  78,
  8,  46, 161, 102,  40, 217,  36, 178, 118,  91, 162,  73, 109, 139, 209,  37,
114, 248, 246, 100, 134, 104, 152,  22, 212, 164,  92, 204,  93, 101, 182, 146,
108, 112,  72,  80, 253, 237, 185, 218,  94,  21,  70,  87, 167, 141, 157, 132,
144, 216, 171,   0, 140, 188, 211,  10, 247, 228,  88,   5, 184, 179,  69,   6,
208,  44,  30, 143, 202,  63,  15,   2, 193, 175, 189,   3,   1,  19, 138, 107,
 58, 145,  17,  65,  79, 103, 220, 234, 151, 242, 207, 206, 240, 180, 230, 115,
150, 172, 116,  34, 231, 173,  53, 133, 226, 249,  55, 232,  28, 117, 223, 110,
 71, 241,  26, 113,  29,  41, 197, 137, 111, 183,  98,  14, 170,  24, 190,  27,
252,  86,  62,  75, 198, 210, 121,  32, 154, 219, 192, 254, 120, 205,  90, 244,
 31, 221, 168,  51, 136,   7, 199,  49, 177,  18,  16,  89,  39, 128, 236,  95,
 96,  81, 127, 169,  25, 181,  74,  13,  45, 229, 122, 159, 147, 201, 156, 239,
160, 224,  59,  77, 174,  42, 245, 176, 200, 235, 187,  60, 131,  83, 153,  97,
 23,  43,   4, 126, 186, 119, 214,  38, 225, 105,  20,  99,  85,  33,  12, 125
};
```

a pre násobenie v $GF\left(2^8\right)$ využívajú nasledujúci kód v jazyku C:

```
word8 mul(word8 a, word8 b) {
  /* multiply two elements of GF(2^m)
   * needed for MixColumn and InvMixColumn
   */
        if (a && b)
                return Alogtable[(Logtable[a] + Logtable[b])%255];
        else
                return 0;
}
```

Príklad 3

Vysvetlite, ako je možné uvedené tabuľky využiť pri softvérovej realizácii algoritmu Rijndael. Na akom princípe sú zostrojené uvedené tabuľky?