Advanced Encryption Standard

Andrew Kozlík

KA MFF UK

Advanced Encryption Standard (AES)

- AES je nástupce DES.
- Byl vybrán v rámci veřejné soutěže v letech 1997 až 2000.
- Vítězem byla šifra Rijndael od Joana Daemena a Vincenta Rijmena.
- $\mathcal{P} = \mathcal{C} = \{0,1\}^{128}$
- $\mathcal{K} = \{0,1\}^{128}$ nebo $\{0,1\}^{192}$ nebo $\{0,1\}^{256}$
- V závislosti na délce klíče rozlišujeme AES-128, AES-192 a AES-256.
- Většina operací se provádí nad tělesem $\mathrm{GF}(2^8)$, neboli $\mathbb{F}_{2^8}$, čili $\mathbb{Z}_2[x]/p(x)$.
  - Jeho prvky reprezentujeme jako polynomy nad $\mathbb{Z}_2$.
  - Operace provádíme modulo ireducibilní polynom $p(x) = x^8 + x^4 + x^3 + x + 1$.

Reprezentace prvků tělesa $\mathrm{GF}(2^8)$

- Prvky tělesa $\mathrm{GF}(2^8)$ jsou polynomy nad $\mathbb{Z}_2$ stupně nejvýše 7.
- Kromě polynomiálního zápisu využíváme také binární nebo hexadecimální zápis pro prvky $\mathrm{GF}(2^8)$.
- Prvek $b_7x^7+b_6x^6+b_5x^5+b_4x^4+b_3x^3+b_2x^2+b_1x^1+b_0x^0$ zapisujeme binárně jako $(b_7b_6b_5b_4b_3b_2b_1b_0)_2$.
- Hexadecimální zápis vznikne z binárního zápisu tak, že každou čtveřici bitů nahradíme jediným symbolem podle následující tabulky:

0000 $\mapsto$ 0	0100 $\mapsto$ 4	1000 $\mapsto$ 8	1100 $\mapsto$ c
0001 $\mapsto$ 1	0101 $\mapsto$ 5	1001 $\mapsto$ 9	1101 $\mapsto$ d
0010 $\mapsto$ 2	0110 $\mapsto$ 6	1010 $\mapsto$ a	1110 $\mapsto$ e
0011 $\mapsto$ 3	0111 $\mapsto$ 7	1011 $\mapsto$ b	1111 $\mapsto$ f

Například $x^6 + x^3 + x^2 = (01001100)_2 = (4c)_{16}$.

Sčítání v $\mathrm{GF}(2^8)$

- Při sčítání se nijak neprojevuje, že počítáme modulo polynom p.
- Jde o běžné sčítání polynomů nad tělesem $\mathbb{Z}_2$, což se v binárním zápisu projevuje jako operace XOR.
- Například:

$$\begin{aligned}(x^6+x^4+x^2+x+1) + (x^7+x+1) &= x^7+x^6+x^4+x^2\\ (01010111)_2 \oplus (10000011)_2 &= (11010100)_2\\ (57)_{16} \oplus (83)_{16} &= (d4)_{16}\end{aligned}$$

Násobení v $\mathrm{GF}(2^8)$

- Nejprve spočítáme součin polynomů nad tělesem $\mathbb{Z}_2$.
- Potom spočítáme zbytek po dělení polynomem $p(x) = x^8 + x^4 + x^3 + x + 1$.
- Například:

$$\begin{aligned}(x^6+x^4+x^2+x+1)(x^7+x+1) &= x^{13}+x^{11}+x^9+x^8+x^6+x^5+x^4+x^3+1 \\ &= (x^5+x^3)p(x) + (\underline{\underline{x^7+x^6+1}}),\end{aligned}$$

čili

$$(57)_{16} \cdot (83)_{16} = (11000001)_2 = (c1)_{16}.$$

Výpočet inverzního prvku v $\mathrm{GF}(2^8)$

- Chceme určit inverzní prvek k polynomu $f \in \mathrm{GF}(2^8)$ modulo ireducibilní polynom p.
- Spočítáme Bézoutovy koeficienty u a v splňující $1 = \mathrm{NSD}(p, f) = up + vf$.
- Potom $vf \equiv 1 \pmod{p}$, přičemž u nepotřebujeme znát.
- Postupujeme pomocí rozšířeného Eukleidova algoritmu.
- Pro výpočet inverzního prvku si lze algoritmus zjednodušit:
  1. $a_0 := p$, $a_1 := f$, $v_0 := 0$, $v_1 := 1$, $i := 1$.
  2. **while** $a_i \neq 1$ **do**
     $a_{i+1} := a_{i-1} \bmod a_i$
     $q_i \quad := a_{i-1} \operatorname{div} a_i$
     $v_{i+1} := v_{i-1} - q_i v_i$
     $i \quad := i + 1$
  3. **return** $v_i \bmod p$

Příklad výpočtu inverzního prvku v $\mathrm{GF}(2^8)$

- Algoritmus:
  1. $a_0 := p$, $a_1 := f$, $v_0 := 0$, $v_1 := 1$, $i := 1$.
  2. **while** $a_i \neq 1$ **do**
     $a_{i+1} := a_{i-1} \bmod a_i$
     $q_i \quad := a_{i-1} \operatorname{div} a_i$
     $v_{i+1} := v_{i-1} - q_i v_i$
     $i \quad := i + 1$
  3. **return** $v_i \bmod p$
- Spočteme inverzní prvek k $(5c)_{16} = x^6 + x^4 + x^3 + x^2$.

i	a_i	q_i	v_i
0	$x^8 + x^4 + x^3 + x + 1$	—	0
1	$x^6 + x^4 + x^3 + x^2$		1
2			
3			
4			

Příklad výpočtu inverzního prvku v $\mathrm{GF}(2^8)$

- Algoritmus:
  1. $a_0 := p$, $a_1 := f$, $v_0 := 0$, $v_1 := 1$, $i := 1$.
  2. **while** $a_i \neq 1$ **do**
     $a_{i+1} := a_{i-1} \bmod a_i$
     $q_i \quad := a_{i-1} \operatorname{div} a_i$
     $v_{i+1} := v_{i-1} - q_i v_i$
     $i \quad := i + 1$
  3. **return** $v_i \bmod p$
- Spočteme inverzní prvek k $(5c)_{16} = x^6 + x^4 + x^3 + x^2$.

i	a_i	q_i	v_i
0	$x^8 + x^4 + x^3 + x + 1$	—	0
1	$x^6 + x^4 + x^3 + x^2$	$x^2 + 1$	1
2	$x^5 + x^4 + x^2 + x + 1$		$x^2 + 1$
3			
4			

Příklad výpočtu inverzního prvku v $\mathrm{GF}(2^8)$

- Algoritmus:
  1. $a_0 := p$, $a_1 := f$, $v_0 := 0$, $v_1 := 1$, $i := 1$.
  2. **while** $a_i \neq 1$ **do**
     $a_{i+1} := a_{i-1} \bmod a_i$
     $q_i \quad := a_{i-1} \operatorname{div} a_i$
     $v_{i+1} := v_{i-1} - q_i v_i$
     $i \quad := i + 1$
  3. **return** $v_i \bmod p$
- Spočteme inverzní prvek k $(5c)_{16} = x^6 + x^4 + x^3 + x^2$.

i	a_i	q_i	v_i
0	$x^8 + x^4 + x^3 + x + 1$	—	0
1	$x^6 + x^4 + x^3 + x^2$	$x^2 + 1$	1
2	$x^5 + x^4 + x^2 + x + 1$	$x + 1$	$x^2 + 1$
3	$x^2 + 1$		$x^3 + x^2 + x$
4			

Příklad výpočtu inverzního prvku v $\mathrm{GF}(2^8)$

- Algoritmus:
  1. $a_0 := p$, $a_1 := f$, $v_0 := 0$, $v_1 := 1$, $i := 1$.
  2. **while** $a_i \neq 1$ **do**
     $a_{i+1} := a_{i-1} \bmod a_i$
     $q_i \quad := a_{i-1} \operatorname{div} a_i$
     $v_{i+1} := v_{i-1} - q_i v_i$
     $i \quad := i + 1$
  3. **return** $v_i \bmod p$
- Spočteme inverzní prvek k $(5c)_{16} = x^6 + x^4 + x^3 + x^2$.

i	a_i	q_i	v_i
0	$x^8 + x^4 + x^3 + x + 1$	—	0
1	$x^6 + x^4 + x^3 + x^2$	$x^2 + 1$	1
2	$x^5 + x^4 + x^2 + x + 1$	$x + 1$	$x^2 + 1$
3	$x^2 + 1$	$x^3 + x^2 + x$	$x^3 + x^2 + x$
4	1		$x^6 + x^4 + 1$

- Inverzní prvek k $(5c)_{16}$ je $x^6 + x^4 + 1 = (51)_{16}$.

Konstrukce S-boxu šifry AES

- Šifra má jediný bijektivní S-box $\mathrm{GF}(2^8) \to \mathrm{GF}(2^8)$.
  1. Ze vstupního prvku se spočítá inverzní prvek $(b_7 \dots b_0)_2$ v tělese $\mathrm{GF}(2^8)$, přičemž prvku 0 se přiřazuje 0.
  2. Aplikuje se afinní zobrazení nad tělesem $\mathbb{Z}_2$:

$$\begin{pmatrix} b'_7 \\ b'_6 \\ b'_5 \\ b'_4 \\ b'_3 \\ b'_2 \\ b'_1 \\ b'_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

- Výstupem S-boxu je prvek $(b'_7 \dots b'_0)_2$.
- Například:
  $(5c)_{16} \overset{1.}{\mapsto} (51)_{16} = (01010001)_2 \overset{2.}{\mapsto} (01001010)_2 = (4a)_{16}$

S-box šifry AES

Obraz prvku $(xy)_{16}$ je popsán následující tabulkou:

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
x	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Objasnění konstrukce S-boxu

- Multiplikativní inverze v tělese $\mathrm{GF}(2^8)$ dává S-boxu dobrou odolnost vůči diferenciální a lineární kryptoanalýze.

- Afinní zobrazení zvyšuje algebraickou složitost S-boxu v $\mathrm{GF}(2^8)$:

$$\text{S-box}(x) = 63 \oplus 8\mathtt{f}\, x^{127} \oplus \mathtt{b5}\, x^{191} \oplus x^{223} \oplus \mathtt{f4}\, x^{239} \oplus 25\, x^{247} \oplus \mathtt{f9}\, x^{251} \oplus 09\, x^{253} \oplus 05\, x^{254}$$

  Bez afinního zobrazení by $\text{S-box}(x) = x^{-1} = x^{254}$ (podle Eulerovy věty).

- Konstanta v afinním zobrazení zbavuje S-box:
  1. pevných bodů ($\text{S-box}(a) = a$),
  2. komplementárních pevných bodů ($\text{S-box}(a) = \bar{a}$).

Šifrovací algoritmus

- Vnitřní stav šifry je reprezentován jako matice typu 4×4 nad tělesem $\mathrm{GF}(2^8)$.
- 16 bajtů otevřeného textu $in_0, \ldots, in_{15}$ se zapíše do sloupců matice vnitřního stavu:

$$\begin{pmatrix} in_0 & in_4 & in_8 & in_{12} \\ in_1 & in_5 & in_9 & in_{13} \\ in_2 & in_6 & in_{10} & in_{14} \\ in_3 & in_7 & in_{11} & in_{15} \end{pmatrix}$$

- Provede se expanze klíče a v závislosti na délce klíče se určí počet rund šifry N_r.

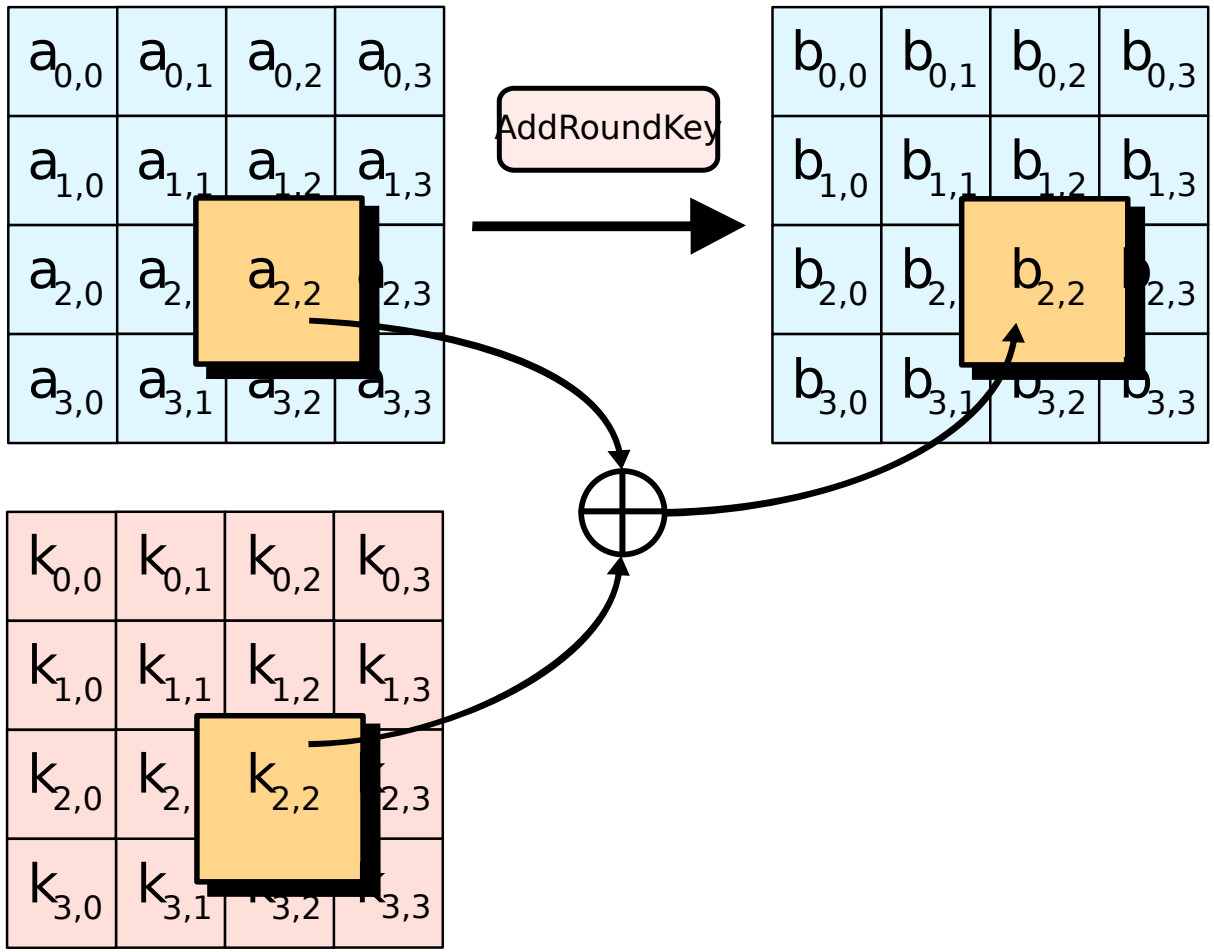
Počet bitů klíče	Počet rund N_r
128	10
192	12
256	14

Šifrovací algoritmus

- Na vnitřním stavu se provede posloupnost operací:
  1. AddRoundKey
  2. **for** $i = 1, \ldots, N_r - 1$ **do**
     SubBytes
     ShiftRows
     MixColumns
     AddRoundKey
  3. SubBytes
     ShiftRows
     AddRoundKey

AddRoundKey

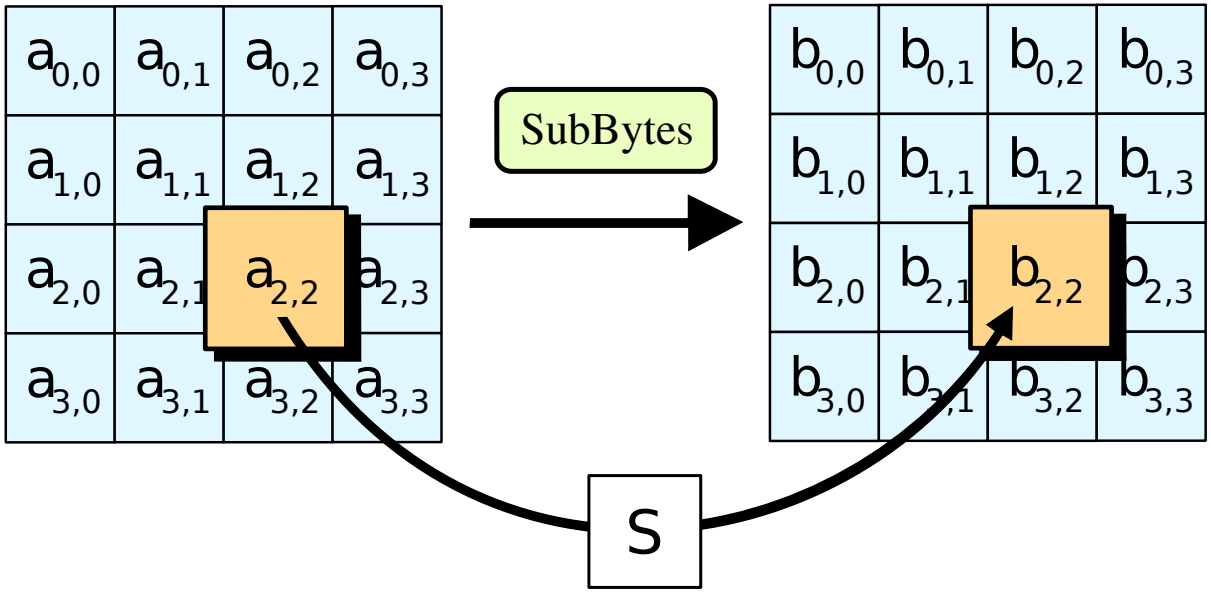
- K vnitřnímu stavu se operací XOR přičte rundovní klíč.
- Jinými slovy, jedná se o součet matic nad tělesem $\mathrm{GF}(2^8)$.



Zdroj: Wikipedia

SubBytes

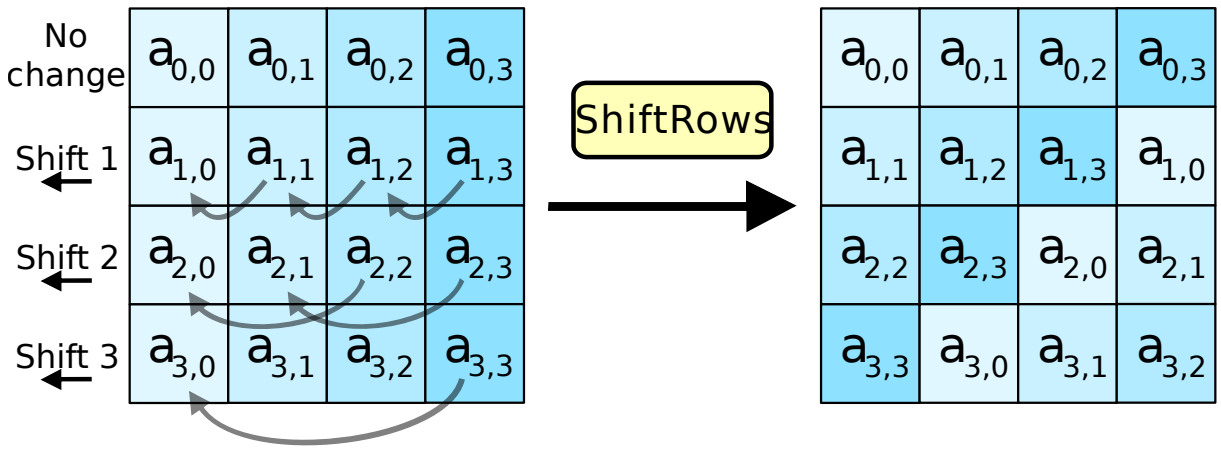
- Každá složka matice vnitřního stavu projde S-boxem.



Zdroj: Wikipedia

ShiftRows

- První řádek matice vnitřního stavu zůstává nezměněn.
- Ostatní řádky se posunou po složkách cyklicky doleva, a to
  - druhý řádek o jednu pozici,
  - třetí řádek o dvě pozice a
  - čtvrtý řádek o tři pozice.
- Jedná se tedy o permutaci na úrovni bajtů.



Zdroj: Wikipedia

MixColumns

- Matice vnitřního stavu A se zleva vynásobí pevně danou maticí:

$$B = MA, \quad \text{kde } M = \begin{pmatrix} (02)_{16} & (03)_{16} & (01)_{16} & (01)_{16} \\ (01)_{16} & (02)_{16} & (03)_{16} & (01)_{16} \\ (01)_{16} & (01)_{16} & (02)_{16} & (03)_{16} \\ (03)_{16} & (01)_{16} & (01)_{16} & (02)_{16} \end{pmatrix}$$

- Násobení zleva transformuje sloupce vnitřního stavu.

- M je tzv. MDS matice (maximum distance separable), tj. $(I \mid M)$ je generující matice MDS kódu.

Dešifrování

- Při dešifrování se provádějí inverzní operace v obráceném pořadí.
- Je třeba spočítat inverzní S-box.
- V inverzní operaci k MixColumns se matice vnitřního stavu násobí zleva maticí:

$$\begin{pmatrix} (0e)_{16} & (0b)_{16} & (0d)_{16} & (09)_{16} \\ (09)_{16} & (0e)_{16} & (0b)_{16} & (0d)_{16} \\ (0d)_{16} & (09)_{16} & (0e)_{16} & (0b)_{16} \\ (0b)_{16} & (0d)_{16} & (09)_{16} & (0e)_{16} \end{pmatrix}$$

Dešifrovací algoritmus

- Na vnitřním stavu se provede posloupnost operací:
  1. AddRoundKey
     InvShiftRows
     InvSubBytes
  2. **for** $i = N_r - 1, \ldots, 1$ **do**
     AddRoundKey
     InvMixColumns
     InvShiftRows
     InvSubBytes
  3. AddRoundKey

Expanze klíče

- Při expanzi klíče se používají dvě vektorové operace.

$$\text{RotWord} : \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \mapsto \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_0 \end{pmatrix}$$

$$\text{SubWord} : \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \mapsto \begin{pmatrix} \text{S-box}(a_0) \\ \text{S-box}(a_1) \\ \text{S-box}(a_2) \\ \text{S-box}(a_3) \end{pmatrix}$$

- Pro $i \geq 1$ se definují konstantní vektory

$$\text{Rcon}_i = \begin{pmatrix} ((02)_{16})^{i-1} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{kde } ((02)_{16})^{i-1} = x^{i-1} \bmod p(x).$$

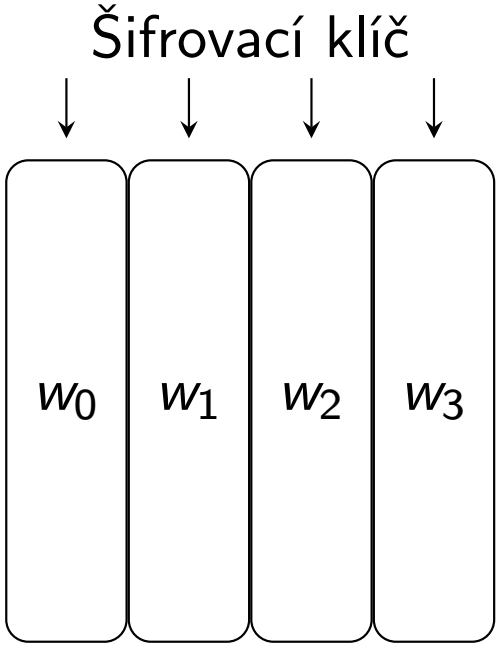
Expanze klíče

- Označme $N_k = (\text{počet bitů klíče})/32$
- Expanze:
  1. Bajty klíče se zapíší do sloupců matice $(\mathbf{w}_0 \mid \cdots \mid \mathbf{w}_{N_k-1})$ typu $4 \times N_k$.
  2. Matice se zprava rozšíří na celkových $4(N_r + 1)$ sloupců. Ty jsou definovány rekurzivně pro $i \geq N_k$:

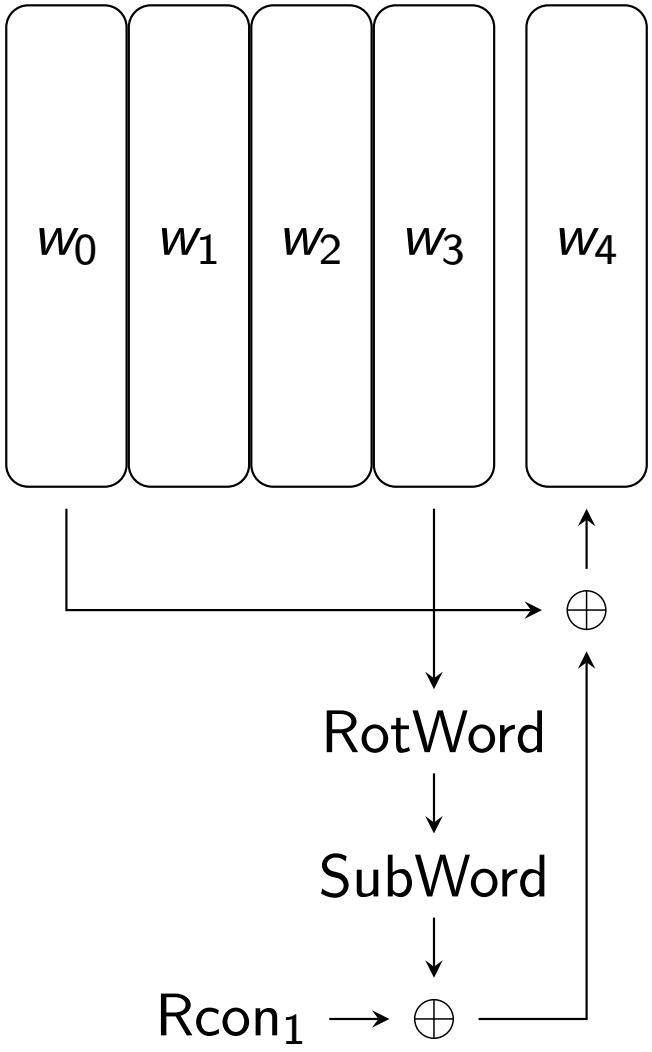
$$\mathbf{w}_i = \begin{cases} \text{SubWord}(\text{RotWord}(\mathbf{w}_{i-1})) \oplus \text{Rcon}_{i/N_k} \oplus \mathbf{w}_{i-N_k}, \\ \qquad\qquad \text{pokud } i \equiv 0 \pmod{N_k}; \\ \text{SubWord}(\mathbf{w}_{i-1}) \oplus \mathbf{w}_{i-N_k}, \\ \qquad\qquad \text{pokud } N_k = 8 \text{ a } i \equiv 4 \pmod{8}; \\ \mathbf{w}_{i-1} \oplus \mathbf{w}_{i-N_k} \qquad \text{v ostatních případech.} \end{cases}$$

  3. Matice se rozřeže na podmatice typu 4×4. Tím získáme posloupnost $N_r + 1$ rundovních klíčů.

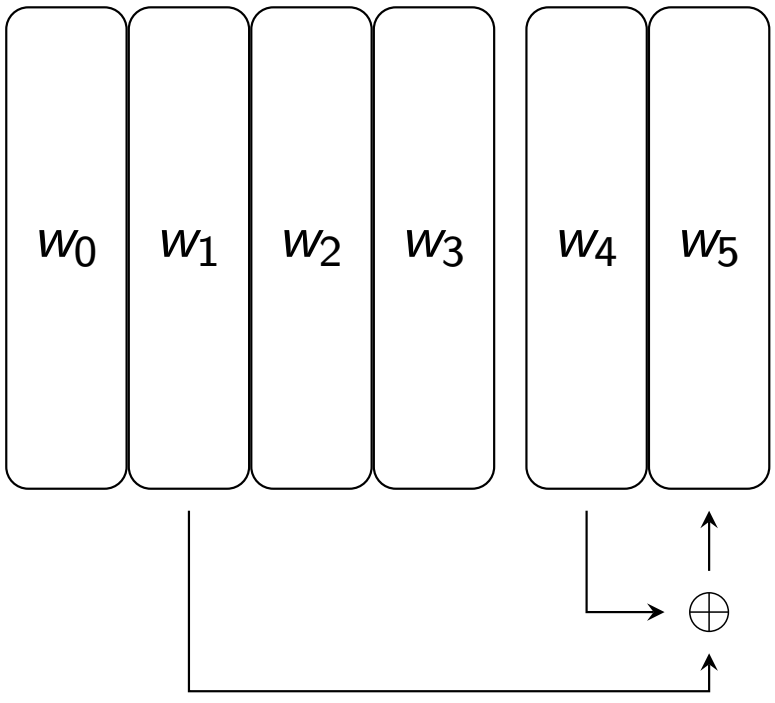
Expanze klíče AES-128



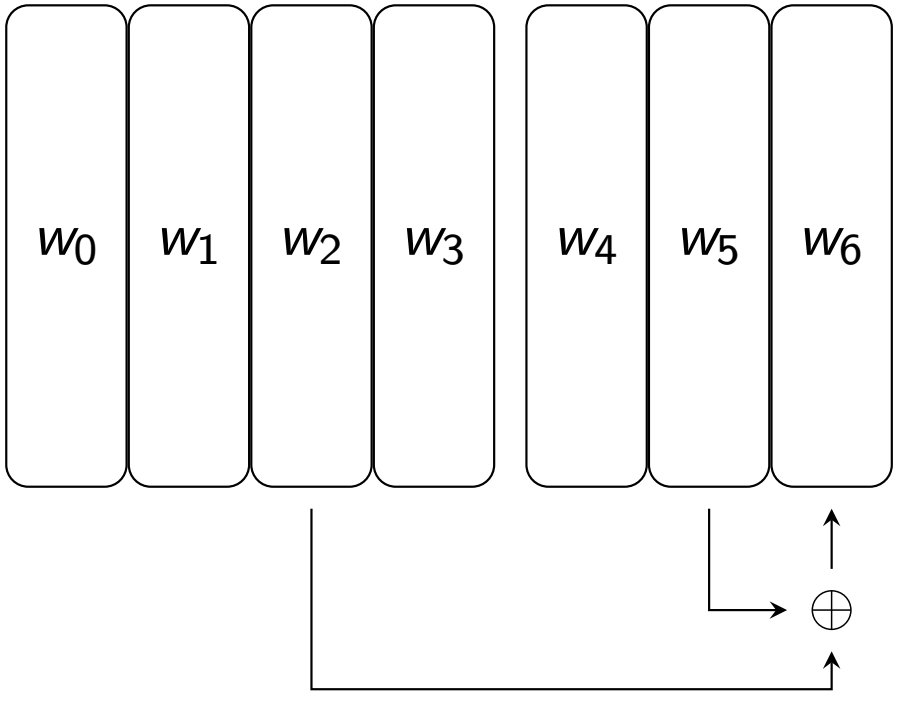
Expanze klíče AES-128



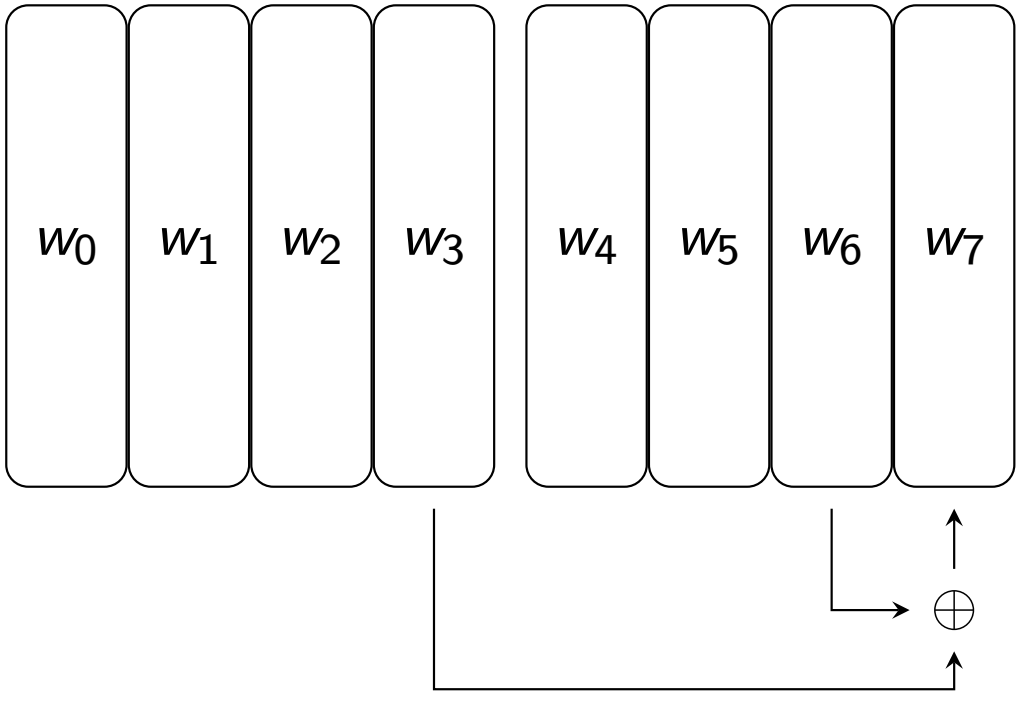
Expanze klíče AES-128



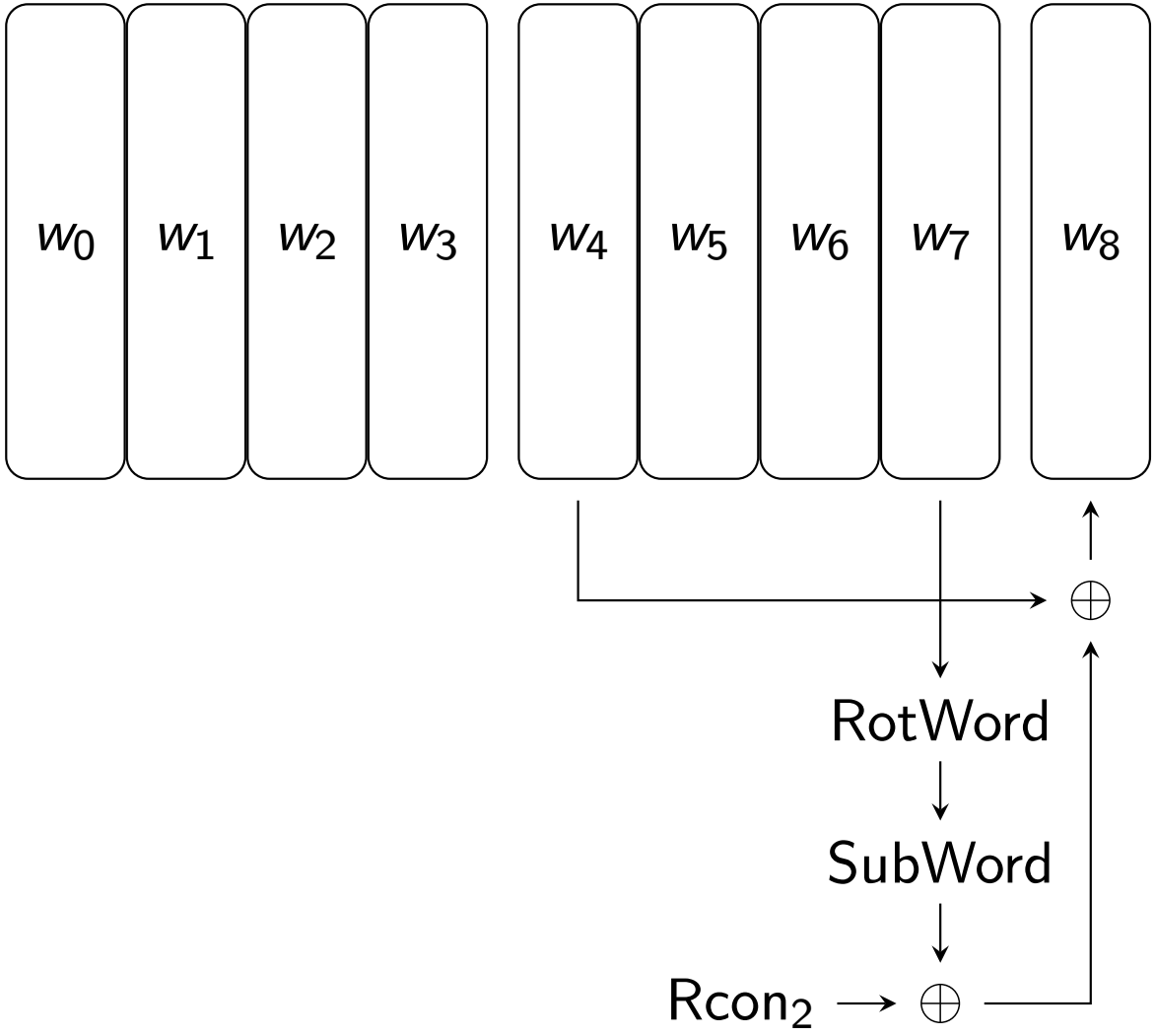
Expanze klíče AES-128



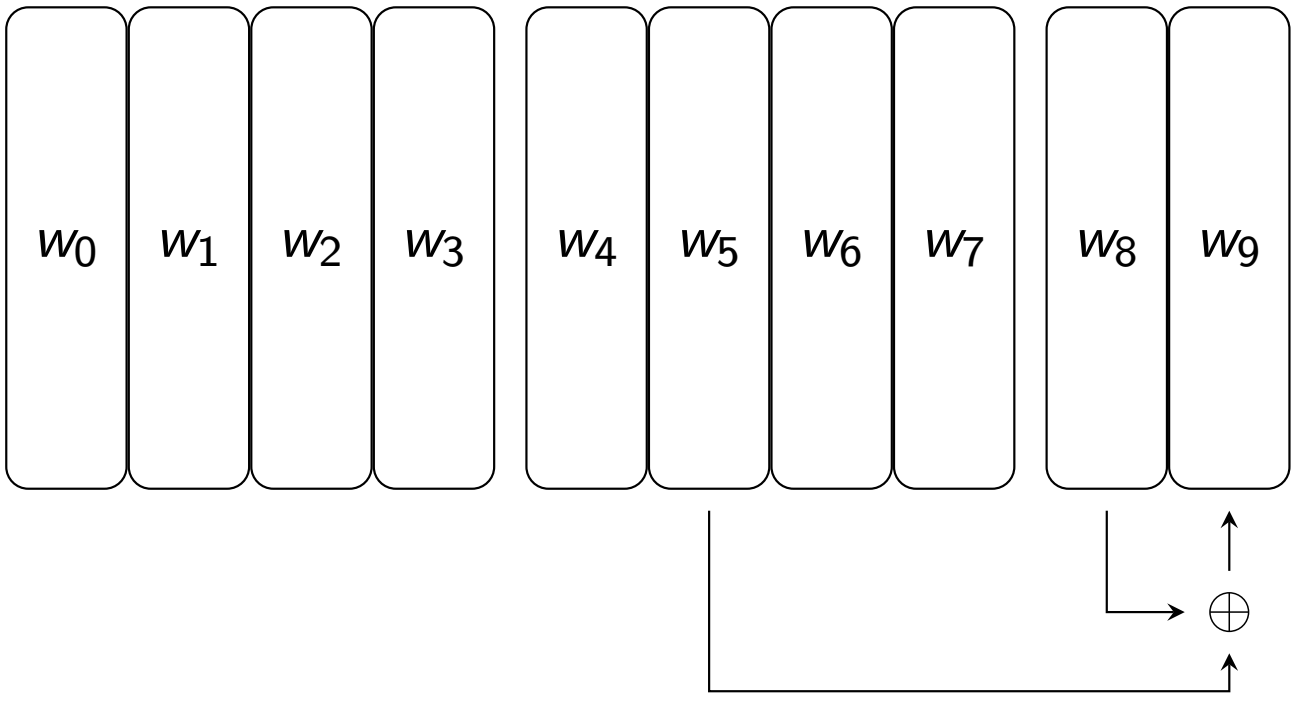
Expanze klíče AES-128



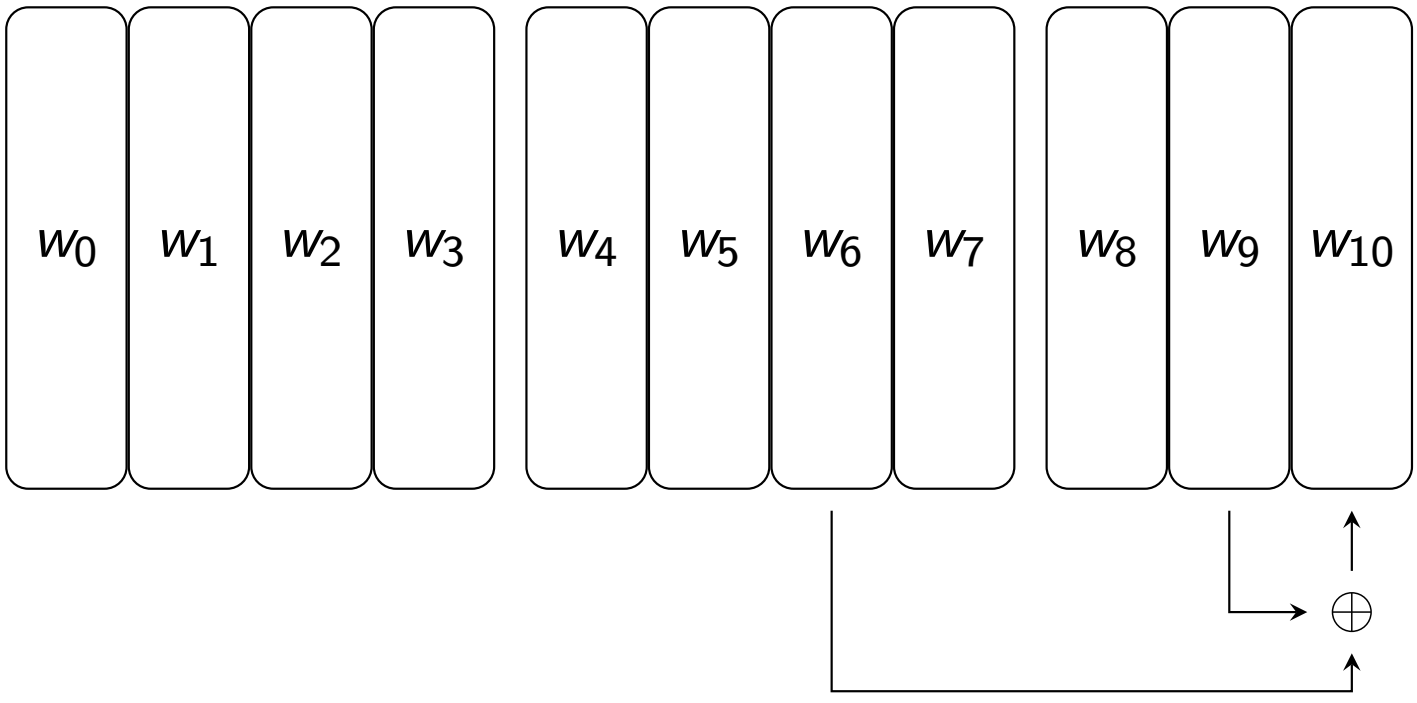
Expanze klíče AES-128



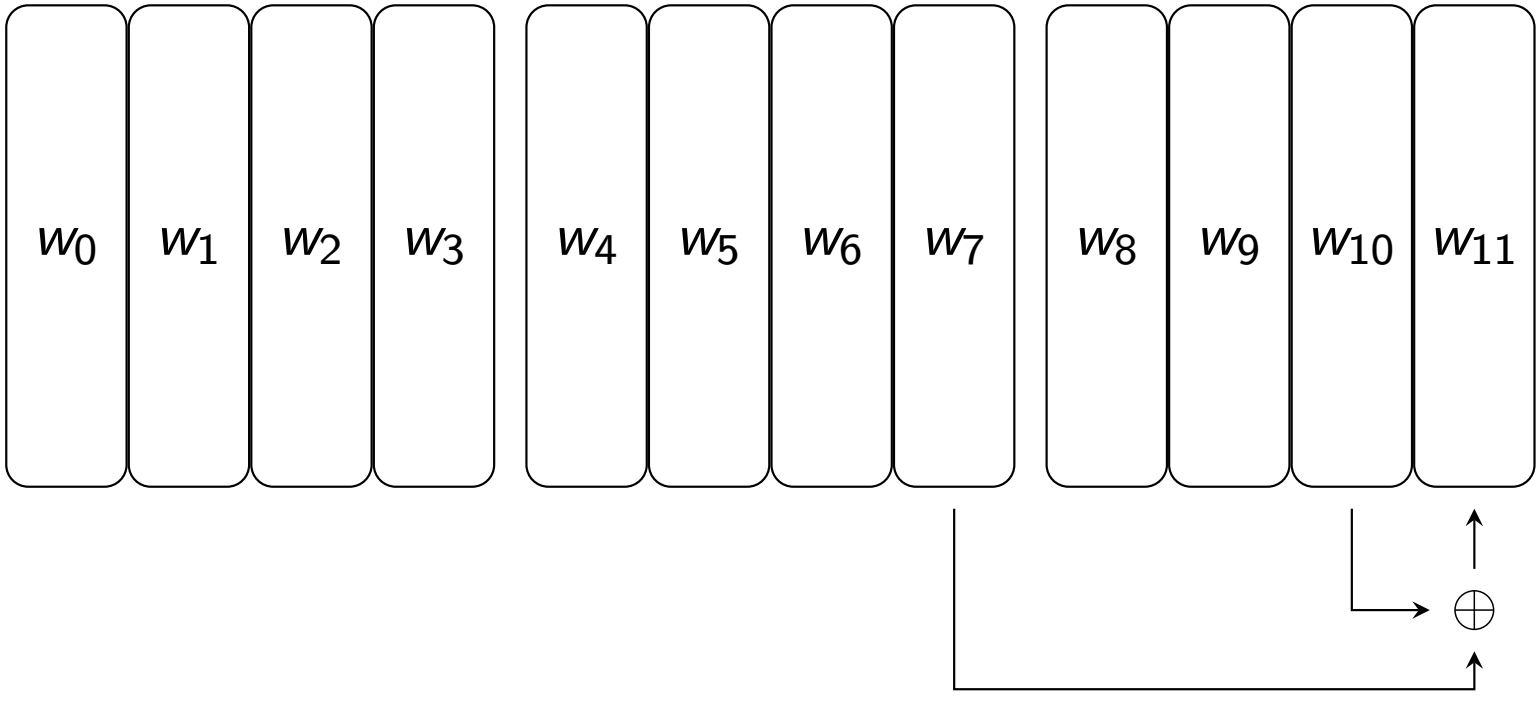
Expanze klíče AES-128



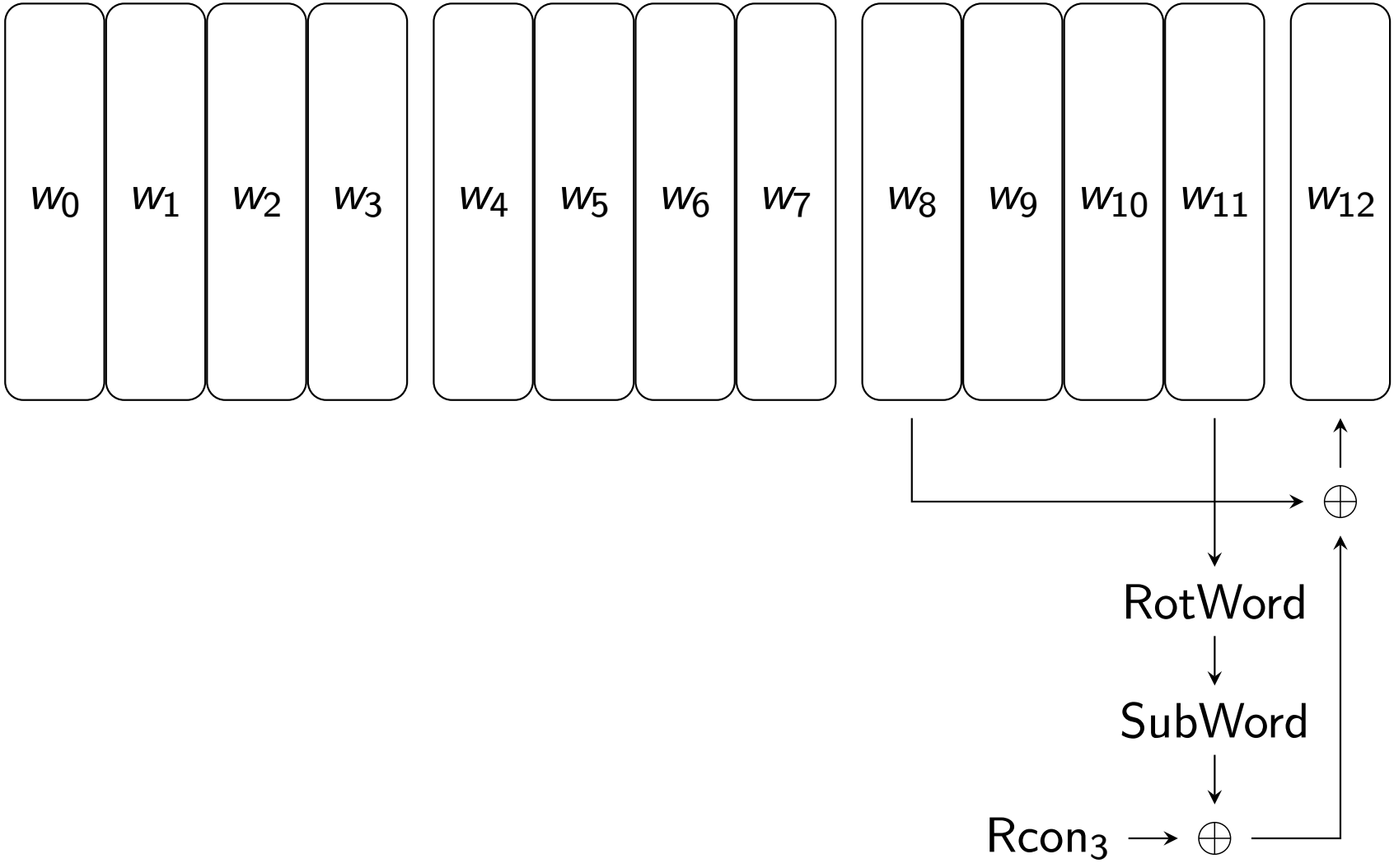
Expanze klíče AES-128



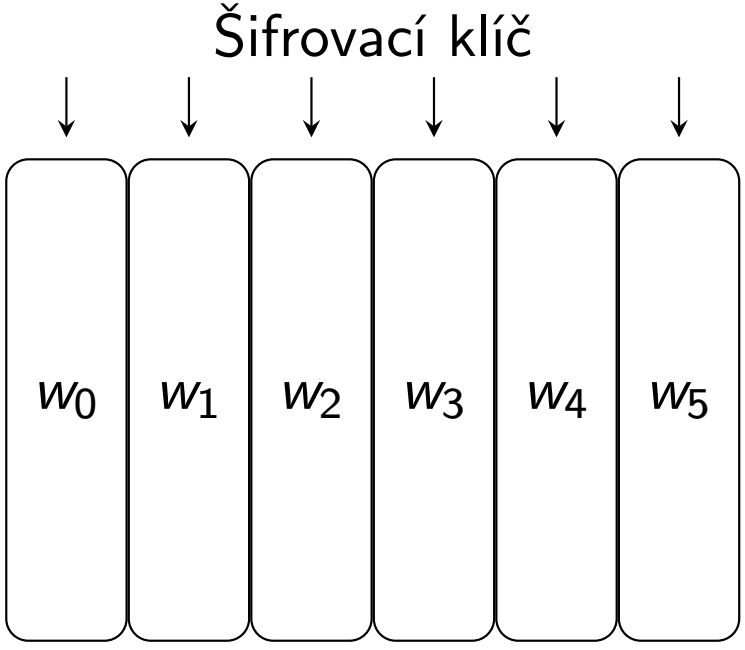
Expanze klíče AES-128



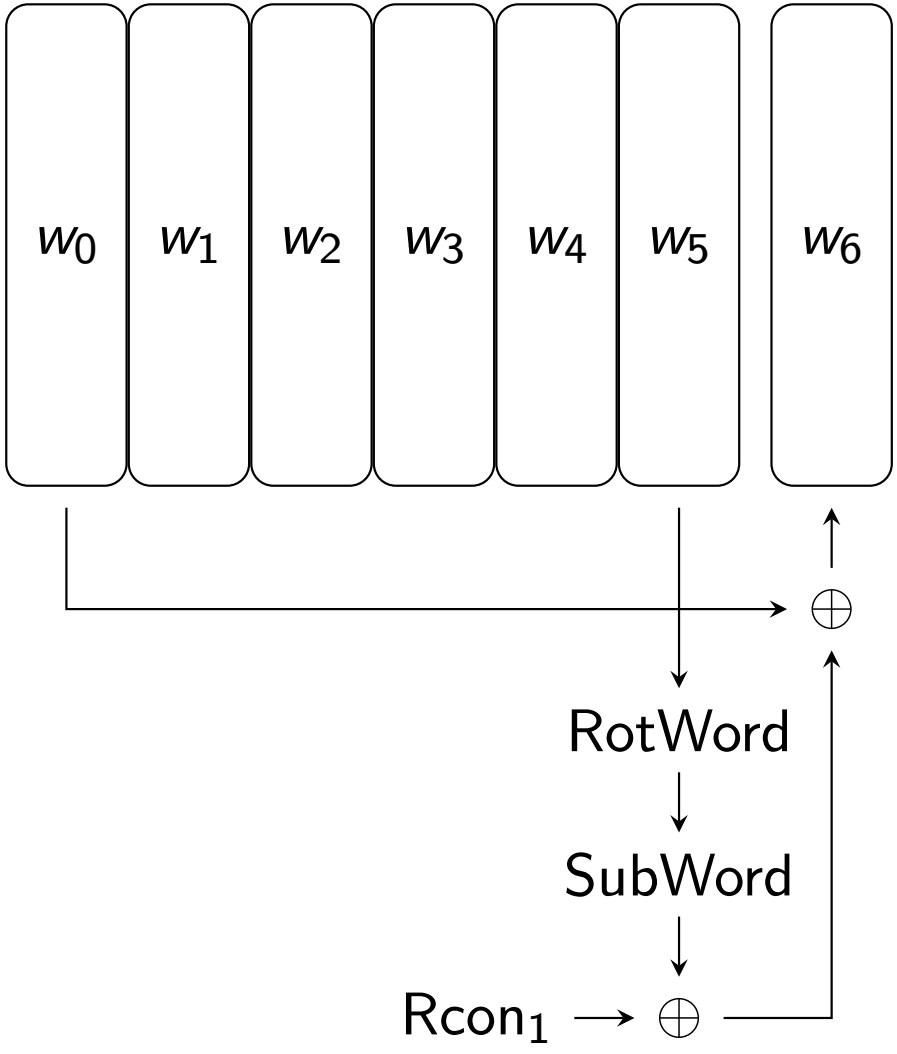
Expanze klíče AES-128



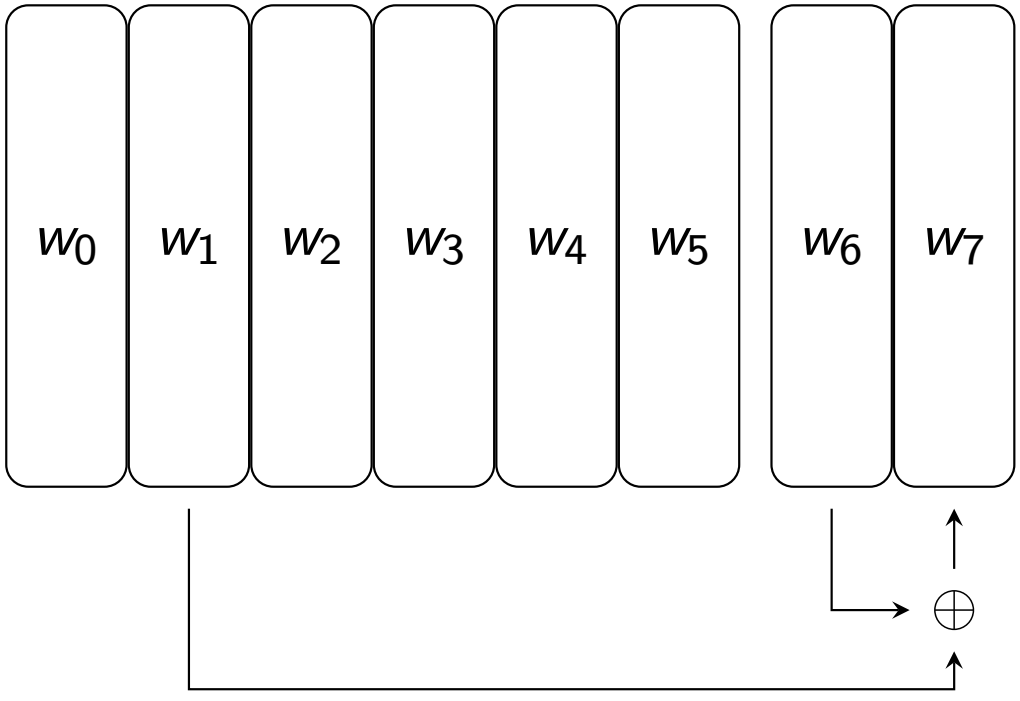
Expanze klíče AES-192



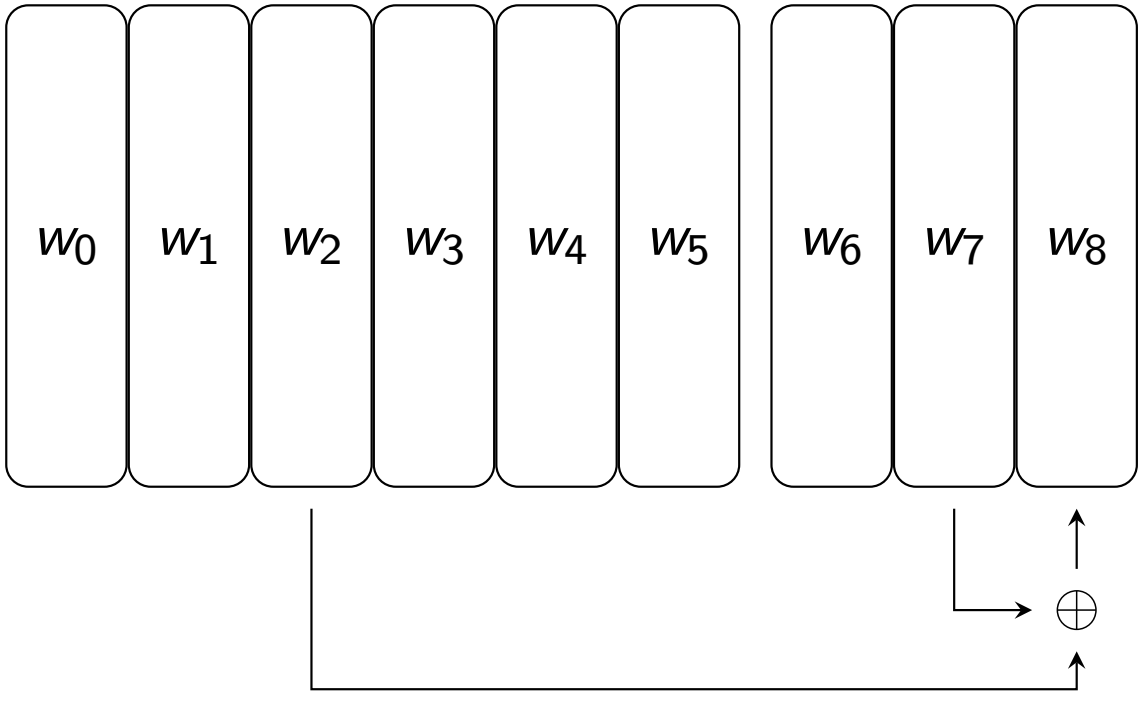
Expanze klíče AES-192



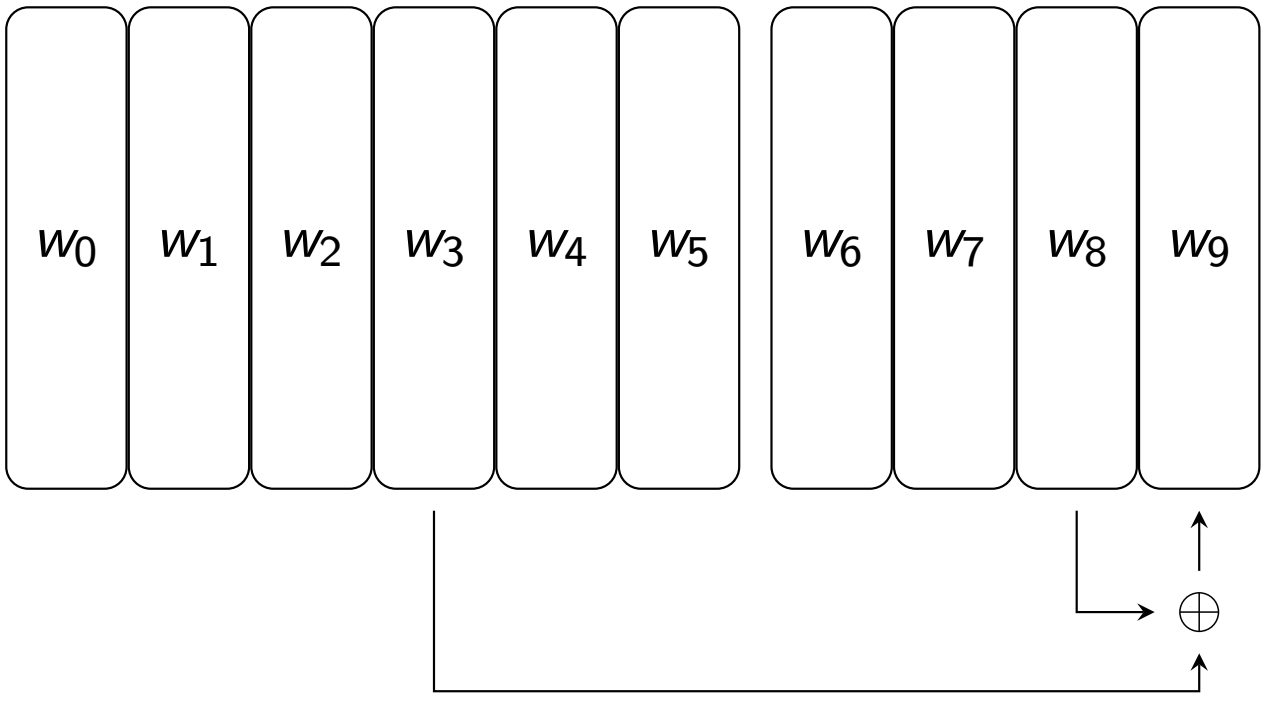
Expanze klíče AES-192



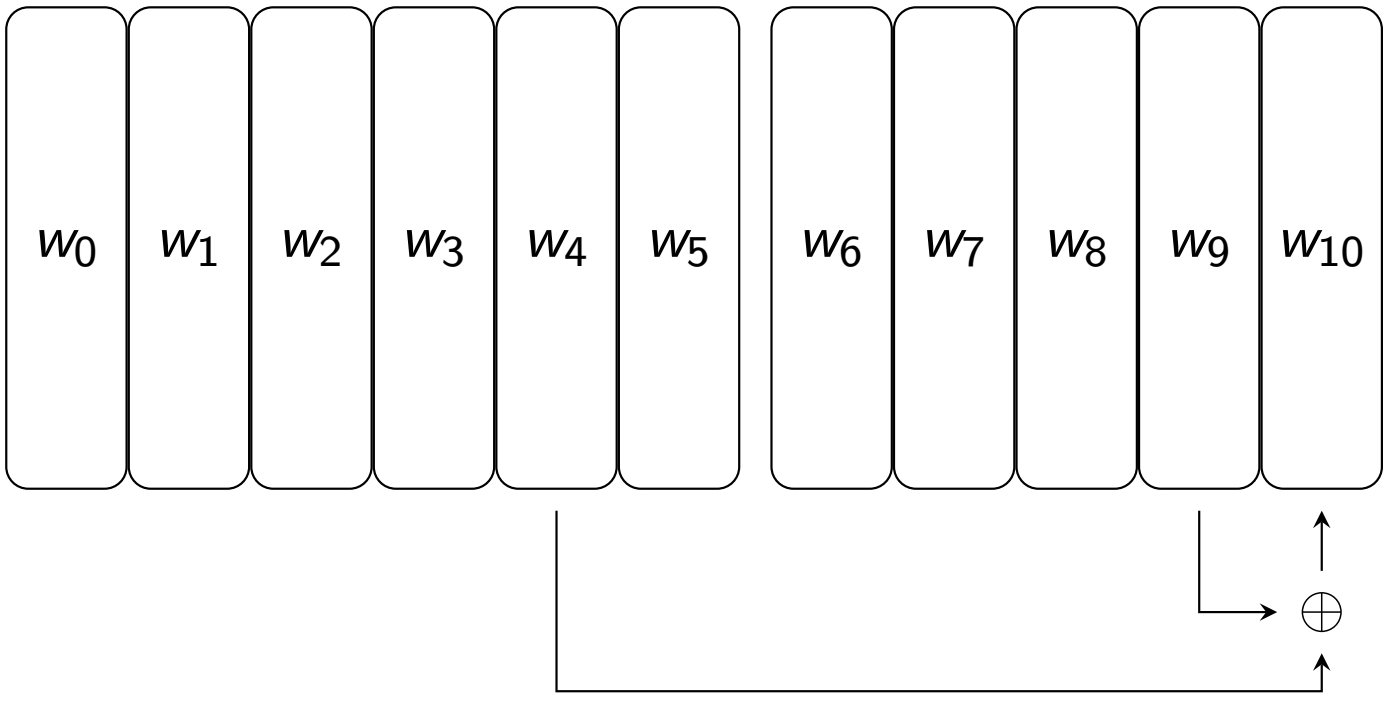
Expanze klíče AES-192



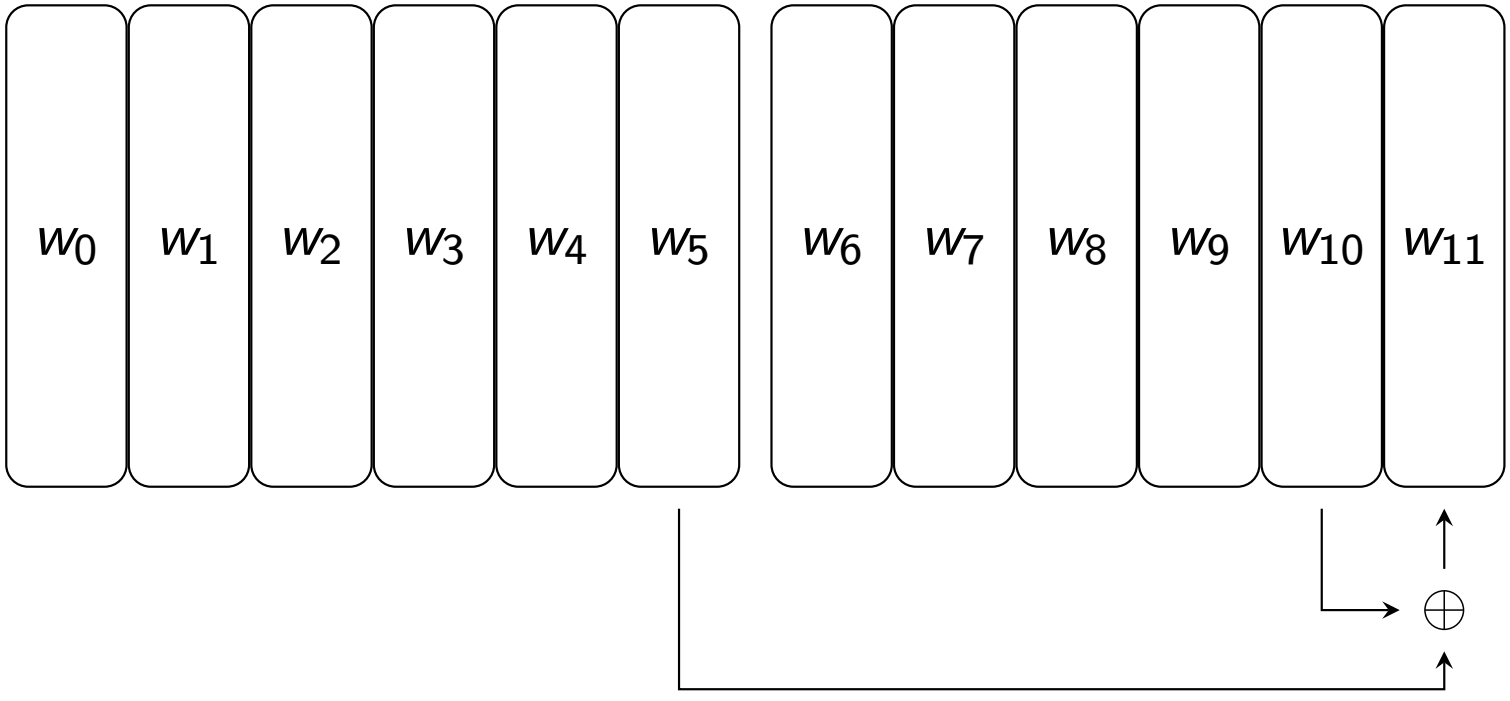
Expanze klíče AES-192



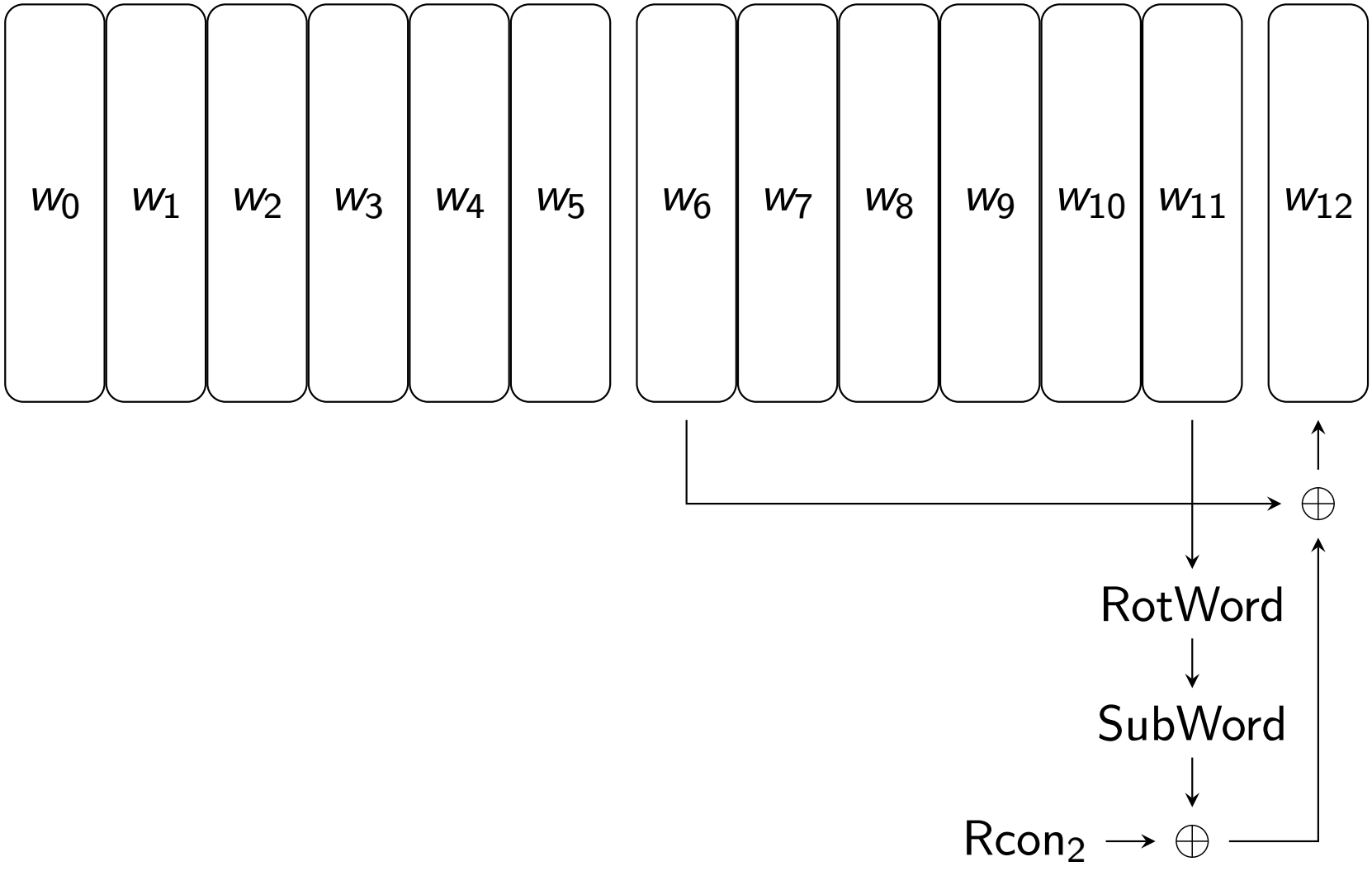
Expanze klíče AES-192



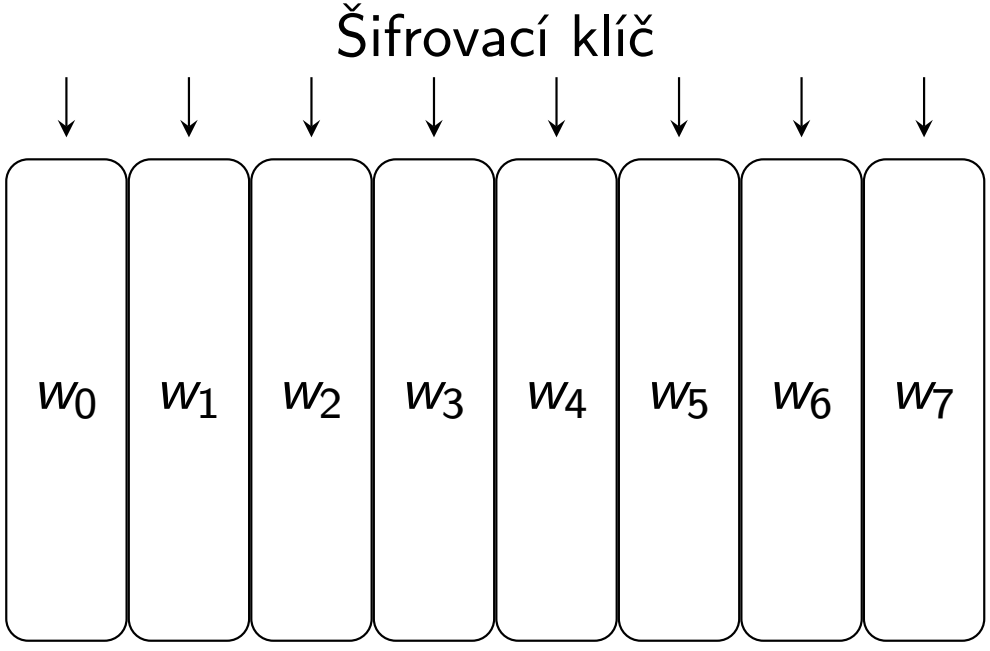
Expanze klíče AES-192



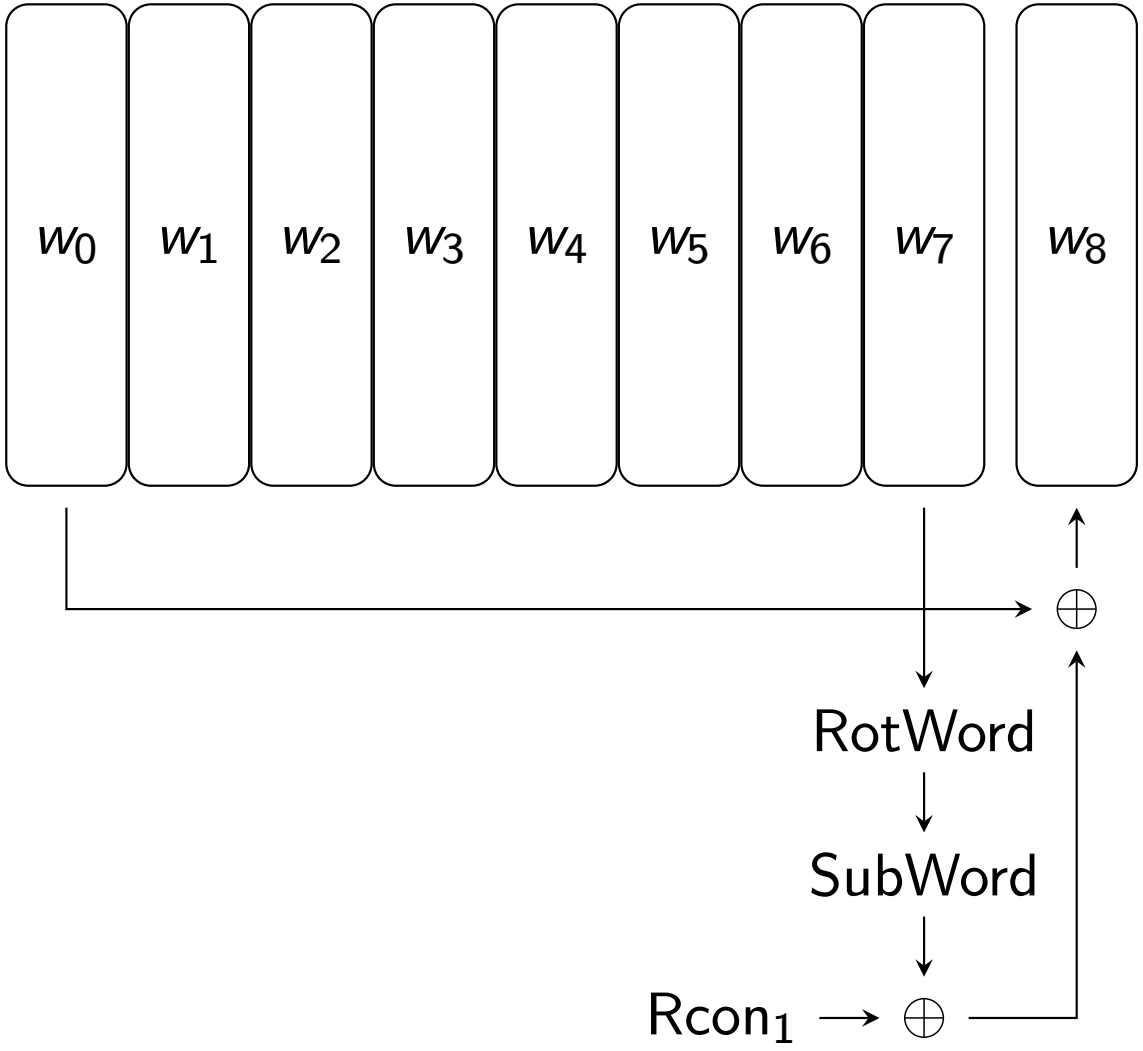
Expanze klíče AES-192



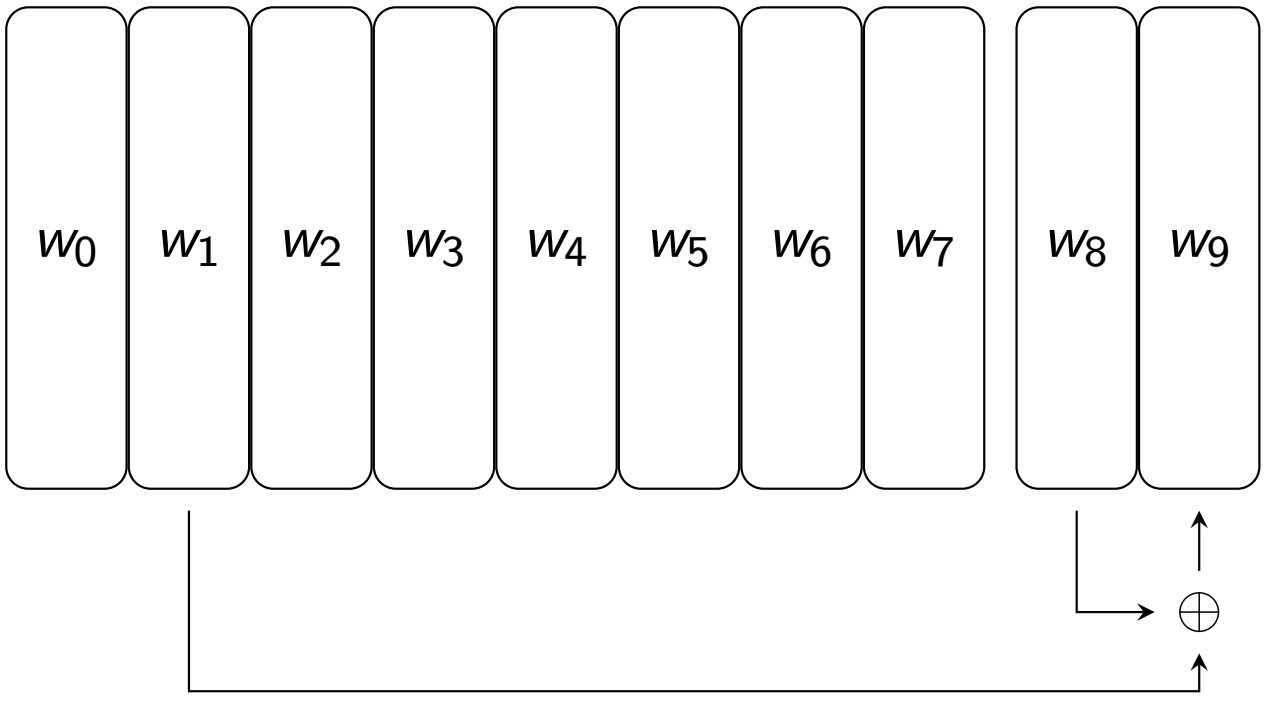
Expanze klíče AES-256



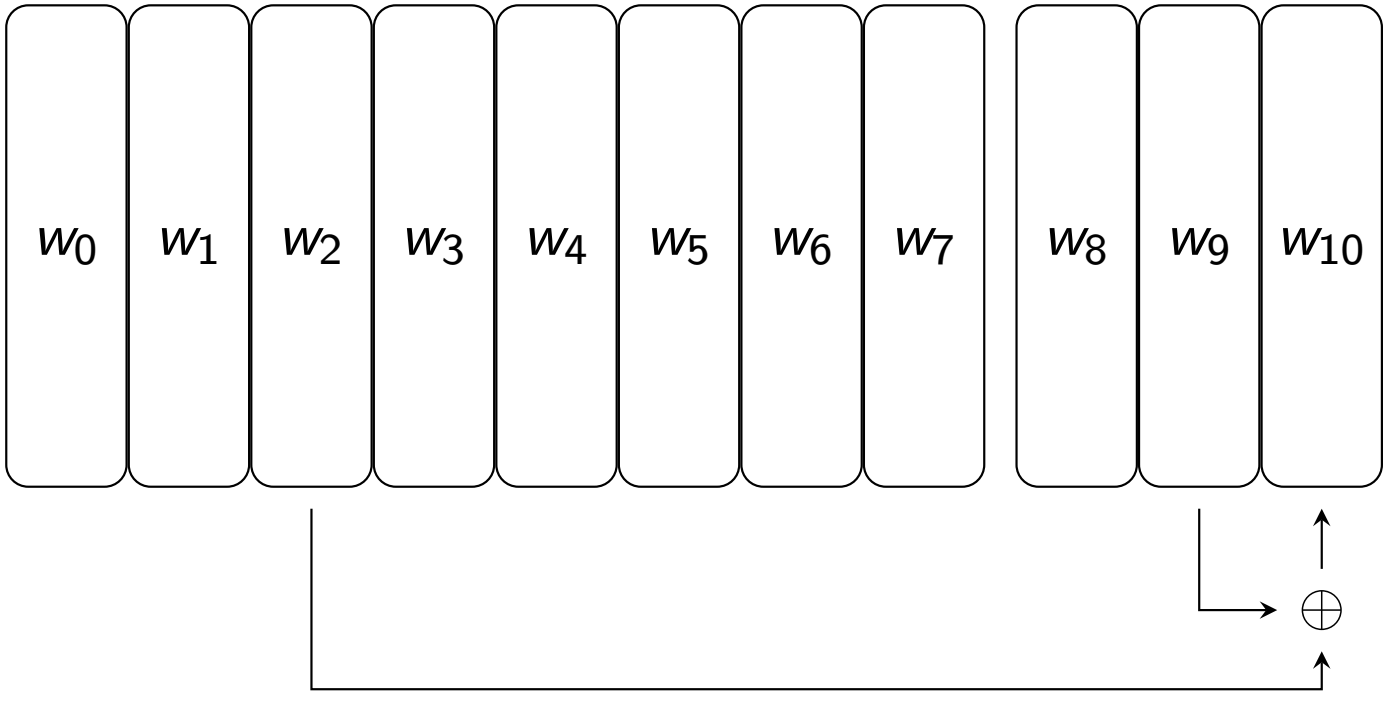
Expanze klíče AES-256



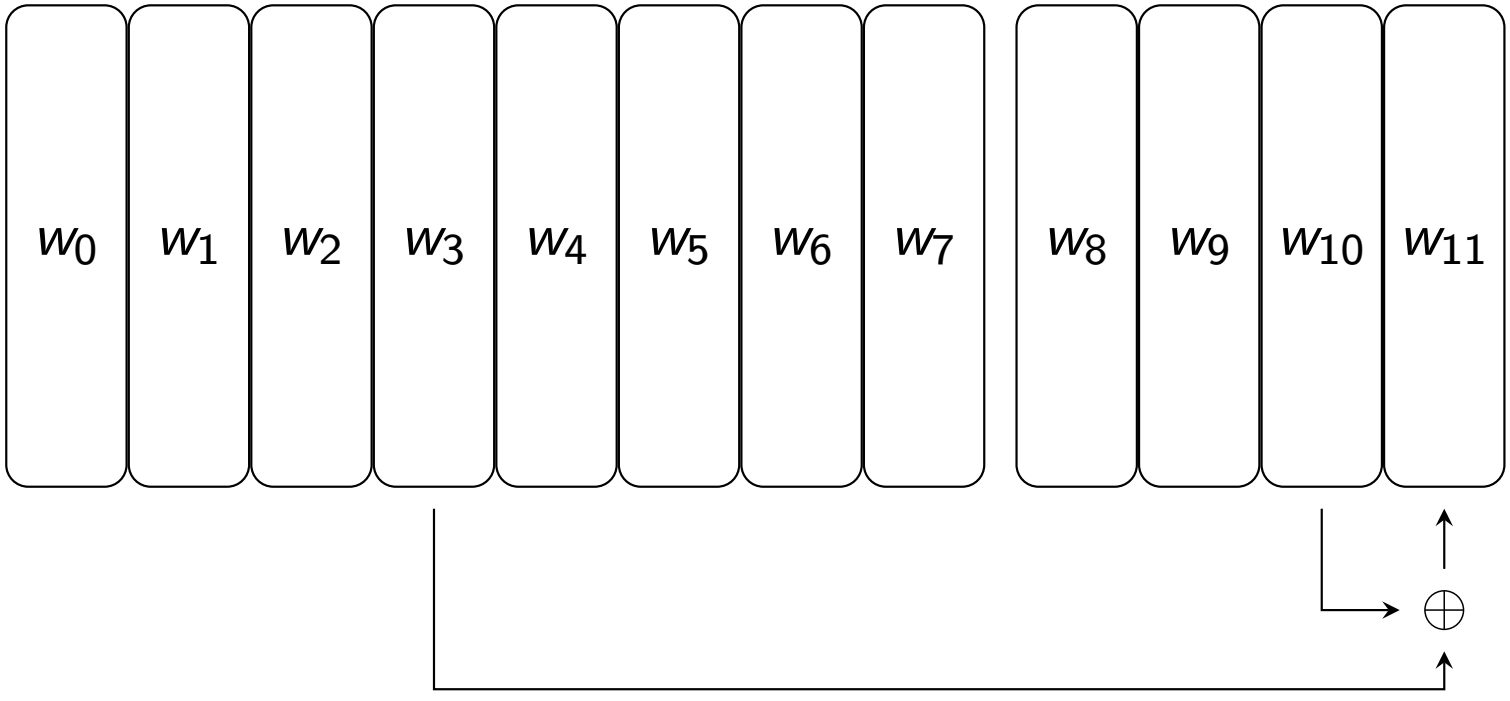
Expanze klíče AES-256



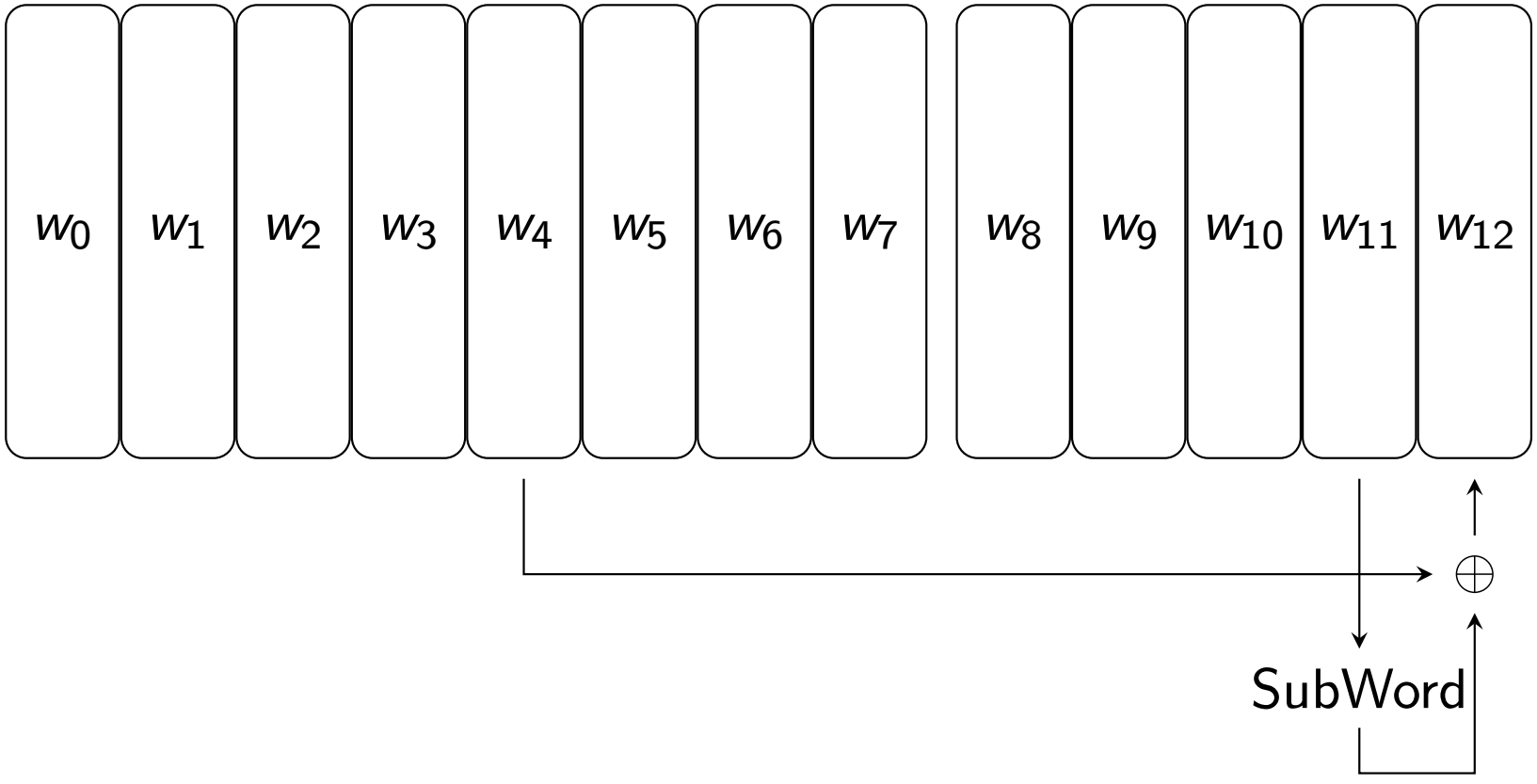
Expanze klíče AES-256



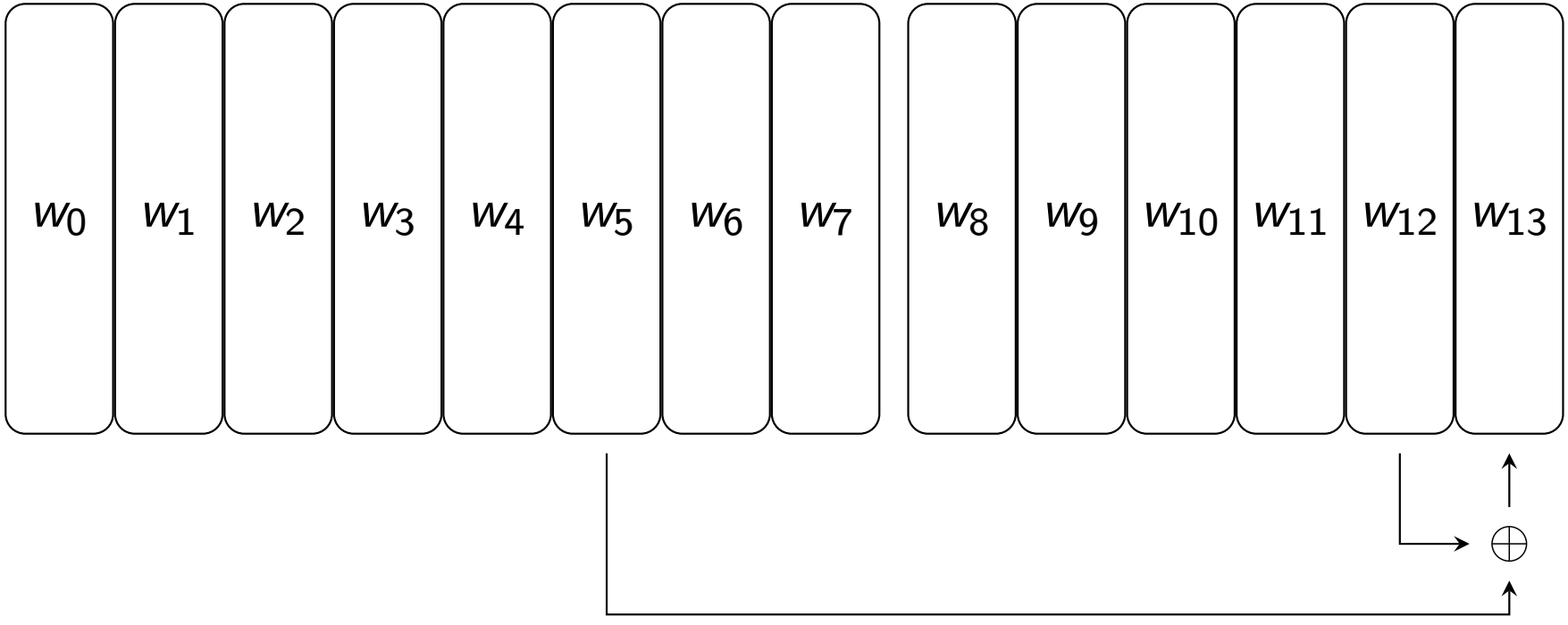
Expanze klíče AES-256



Expanze klíče AES-256



Expanze klíče AES-256



Srovnání rychlosti (bez expanze klíče)

- DES (software) 46 CPB (cycles per byte)
- 3DES (software) 123 CPB
- AES-128 (software) 14 CPB
- AES-128 (hardware) 1,3 – 4,2 CPB (Intel AES-NI)
  tj. 1,4 GB/s – 440 MB/s na 2 GHz jádru.