Teorija kodiranja in kriptografija 2013/2014

AES

Arjana Žitnik

Univerza v Ljubljani, Fakulteta za matematiko in fiziko

Ljubljana, 18. 3. 2014

AES - zgodovina

Septembra 1997 je NIST objavil natečaj za izbor nove simetrične bločne šifre AES. Specifikacije:

- bločna dolžina 128
- dolžina ključev 128, 192, 256
- prosta uporaba

Prispelo je 15 ustreznih predlogov iz 12 držav. Kriterij izbire:

- varnost
- učinkovitost (časovna in prostorska)
- značilnosti algoritma in implementacije (fleksibilnost, preprostost, ...)

avgust 1999: izbranih je 5 finalistov

- MARS
- RC6
- Rijndael
- Serpent
- Twofish

oktober 2000: končni izbor je Rijndael (Rijmen, Daemen, Belgija)

maj 2002: AES sprejet kot US Federal Information Processing Standard (FIPS 197).

junij 2003: ameriška vlada odobrila AES za zaščito zaupnih podatkov.

Opis kriptosistema AES

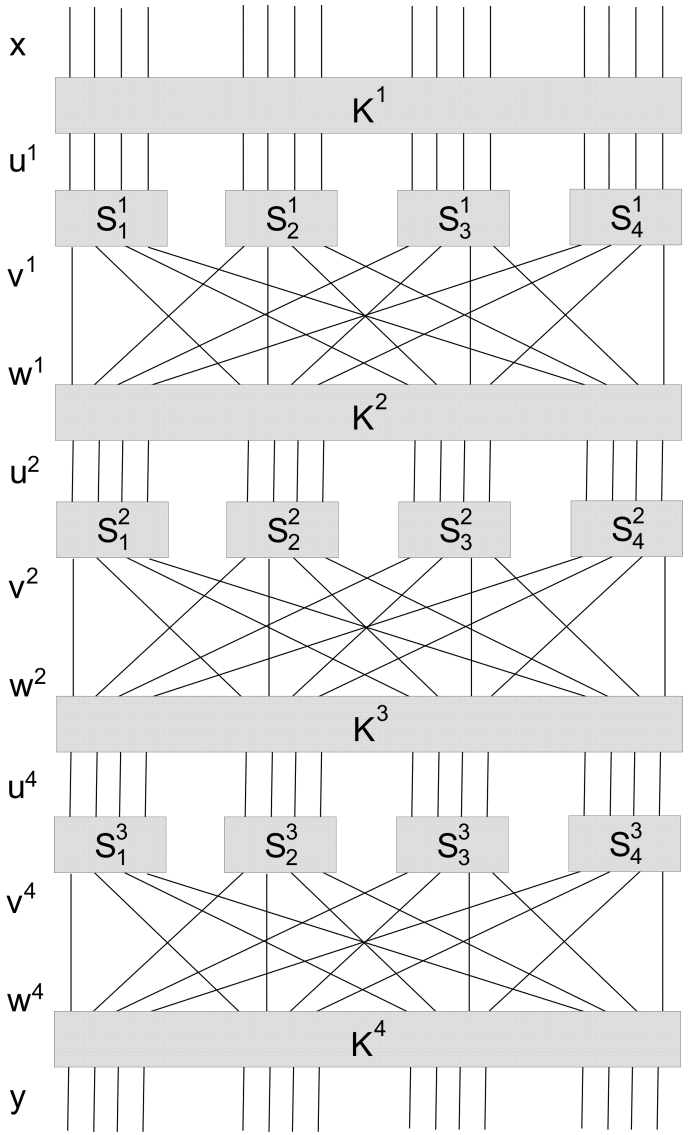
abeceda $\Sigma = \{0, 1\}$,
zlog = 8 znakov (bitov),
beseda = 4 zlogi = 32 bitov.

AES je SPN omrežje + dodatna linearna transformacija.
Parametri:

- bločna dolžina: 128 bitov = 16 zlogov = 4 besede,
- dolžina ključa N_k (v besedah) je lahko 4, 6 ali 8,
- število krogov N_r je enaka $N_k + 6$, tj. 10, 12 ali 14.
- razpored podključev: tabela besed velikosti $4N_r + 4$, $rk = rk_0, rk_1, \ldots, rk_{4N_r+3}$, kjer je rk_i beseda.

Opomba: AES podpira tudi bločne dolžine 6 ali 8 besed; omejili se bomo na opis s 4 besedami.

SPN-omrežje


x
K^1
u^1
S_1^1 S_2^1 S_3^1 S_4^1
v^1
w^1
K^2
u^2
S_1^2 S_2^2 S_3^2 S_4^2
v^2
w^2
K^3
u^4
S_1^3 S_2^3 S_3^3 S_4^3
v^4
w^4
K^4
y

Podatkovne strukture AES

besedilo shranimo v tabelo zlogov velikosti 4×4:

$$b = b_{0,0}b_{0,1}\ldots b_{3,3} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}.$$

Tudi kriptogram $c = c_{0,0}c_{0,1}\ldots c_{3,3}$ shranimo v tabelo zlogov velikosti 4×4.

Stanje je delni rezultat v posameznih krogih:

$$s = s_{0,0}s_{0,1}\ldots s_{3,3} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}.$$

Kodiranje pri AES

Podatki: byte $b[0..3, 0..3]$ besedilo
word $k[0..4N_r + 3]$ razpored podključev

Rezultat: byte $c[0..3, 0..3]$ kriptogram

Lokalne spremenljivke: byte s[0..3,0..3] stanje
int r števec krogov

Pomožne funkcije:
AddRoundKey, SubBytes, ShiftRows, MixColumns

Postopek:

$s = b$

$s = \text{AddRoundKey}(s, rk[0..3])$ (primešamo podključ)

za $r = 1, 2, \ldots N_r - 1$ **ponovi**

$\text{SubBytes}(s)$ (substitucija)

$\text{ShiftRows}(s)$ (permutacija)

$\text{MixColumns}(s)$ (linearna transformacija)

$\text{AddRoundKey}(s, rk[4r..4r + 3])$ (primešamo podključ)

(Zadnji krog)

$\text{SubBytes}(s)$

$\text{ShiftRows}(s)$

$\text{AddRoundKey}(s, rk[4N_r..4N_r + 3])$

$c = s$

Pomožne funkcije

AddRoundKey(byte $s[0..3, 0..3]$, word $rk[0..3]$)

za $j = 0, 1, 2, 3$ **ponovi**

$s_j = s_j \oplus rk[j]$

vrni s

Pri tem je s_j stolpec stanja s:

$$s_j = \begin{bmatrix} s_{0,j} \\ s_{0,j} \\ s_{0,j} \\ s_{0,j} \end{bmatrix}.$$

ShiftRow

Transformacija **ShiftRow** *j*-to vrstico *s* ciklično premakne za *j* mest v levo:

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,0} & s_{3,1} & s_{3,2} \end{bmatrix}.$$

Skupaj s transformacijo MixColumn poskrbi za razpršitev podatkov.

Operacije v obsegu GF(2^8)

Funkcije SubBytes in MixColumns ter postopek za generiranje razporeda podključev uporabljajo operacije v končnem obsegu GF(2^8):

- Obseg generiramo z nerazcepnim polinomom $f(x) = x^8 + x^4 + x^3 + x + 1$
- Njegove elemente identificiramo z zlogi: polinomu $b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$ priredimo zlog $b_7b_6b_5b_4b_3b_2b_1b_0$, kjer $b_i \in \{0, 1\}$; vsak zlog lahko zapišemo kot par šestnajstiških števk.
- Zdaj lahko zloge seštevamo, odštevamo, množimo in delimo (kot elemente GF(2^8)).

SubBytes

Na vsakem od elementov matrike s se neodvisno izvede naslednji operaciji:

1. izračun inverza $s_{i,j}$ v GF(2^8) (element 0 se preslika sam vase),
2. izračun afine transformacije nad $\mathbb{Z}_2$, podane z izrazom

$$s_{i,j} = A \cdot s_{i,j} \oplus c \quad (\text{v } \mathbb{Z}_2)$$

kjer je

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \text{ in } c = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Zgled - SubBytes

$s_{i,j} = 53_{(16)}$
$s_{i,j}^{-1} = CA_{(16)} = 11001010$

$$A \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \oplus c = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Novi $s_{i,j} = 11101101 = ED_{(16)}$.

SubBytes kot tabela

X \ 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
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

MixColumns

Za vsak stolpec s_j, $j = 0, 1, 2, 3$ ponovi

$$s_j = \begin{bmatrix} '02' & '03' & '01' & '01' \\ '01' & '02' & '03' & '01' \\ '01' & '01' & '02' & '03' \\ '03' & '01' & '01' & '02' \end{bmatrix} \cdot s_j$$

Množenje poteka v GF(2^8).

$$\begin{aligned} '01' &= 00000001 = 1 \\ '02' &= 00000010 = x \\ '03' &= 00000011 = x + 1 \end{aligned}$$

Razpored podključev: KeyExpansion

Podatek: word $K[0..N_k - 1]$ ključ

Rezultat: word $rK[0..4N_r + 3]$ razpored podključev

Pomožne funkcije:
SubWord: na vsakem zlogu uporabi SubBytes:
$(z_0, z_1, z_2, z_3) \mapsto (Az_0^{-1} + c, Az_1^{-1} + c, Az_2^{-1} + c, Az_3^{-1} + c)$.
RotWord: ciklični zamik zaporedja zlogov za 1 v levo:
$(z_0, z_1, z_2, z_3) \mapsto (z_1, z_2, z_3, z_0)$.
Rcon: $M \mapsto ('02'^{n-1}, 00, 00, 00)$
(4 zlogi, računanje v GF(2^8)).

Razpored podključev: postopek

Postopek:

za $i = 1, 2, \ldots N_k - 1$ **ponovi**

$rk[i] = K[i]$ (v prvi del prepišemo ključ)

za $i = N_k, N_k + 1, \ldots 4N_r + 1$ **ponovi**

$tmp = rk[i-1]$

če $i \bmod N_k = 0$ **potem**

$tmp =$SubWord(RotWord(tmp)) $\oplus$ Rcon$[i/N_k]$

sicer če $N_k > 6$ **in** $i \bmod N_k = 4$ **potem**

$tmp =$SubWord(tmp)

$rk[i] = rk[i - N_k] \oplus tmp$

Analiza varnosti

Pri konstrukciji novega simetričnega bločnega kriptosistema je potrebno upoštevati:

- kriptogram mora biti čim bolj naključen
  - vsak znak kriptograma mora biti odvisen od vseh znakov sporočil (razpršitev - diffusion)
  - zveza med kriptogramom in ključem mora biti čim bolj zapletena (zmeda - confusion)
- algoritem mora biti varen pred vsemi znanimi napadi
- znano matematično ozadje omogoča boljšo analizo

Nekaj znanih napadov na simetrične šifre

- izčrpen pregled vseh ključev (dovolj velik prostor ključev!),
- linearna kriptoanaliza,
- diferenčna kriptoanaliza,
- algebraični napadi (reševanje sistemov enačb)
- ...

Napadi na implementacijo:

- napadi s stranskim kanalom (merjenje časa, porabo energije,...)
- povzročanje napak pri računanju (prekinitev napajanja, laserski žarki...)

Več o varnosti kasneje!