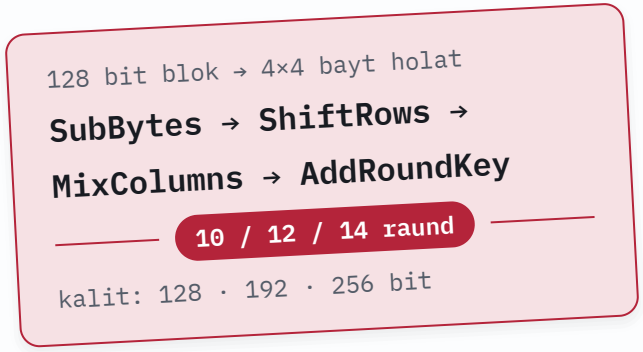
Sharof Rashidov nomidagi Samarqand davlat universiteti

KIBERXAVFSIZLIK ASOSLARI · 7-MA'RUZA

AES algoritmi: tuzilishi, raund funksiyalari, S-box va MixColumns

Holat matritsasidan Galua maydonigacha — SP-tarmoqning ichki qurilishi

Axborot xavfsizligi yo'nalishi talabalari uchun



Muallif: **Esanov Otabek**

2026-yil 25-avgust

REJA · 7-MA'RUZA

Bitta zamonaviy standartni to'liq ochamiz: AES

1. NIST tanlovi va Kerkhoffs prinsipi
2. AES parametrlari — DES bilan taqqoslash
3. Holat matritsasi: 16 baytdan 4×4 jadvalgacha
4. Umumiy oqim: dastlabki va yakuniy raund farqi
5. Raund ichida: SubBytes, ShiftRows, MixColumns, AddRoundKey
6. Galua maydoni GF(2^8) va uning arifmetikasi
7. S-box qanday qurilgan
8. Kalit kengaytirish: RotWord, SubWord, Rcon
9. Deshifrlash va teskari amallar
10. Xavfsizlik tahlili va amaliyot (AES-NI, Python)

01 · NIST TANLOVI

DES dan keyin: ochiq xalqaro tanlov

1997

NIST ochiq tanlov e'lon qiladi — DES eskirgan, 56-bit kalit endi zaif

1998

12 davlatdan **15 nomzod** qabul qilinadi va butun dunyoga tahlil uchun ochiladi

1999

5 finalist: MARS, RC6, Rijndael, Serpent, Twofish

2000–01

Rijndael g'olib (2-oktyabr 2000); 2001 — FIPS PUB 197

Rijndaelni belgiyalik **Joan Daemen** va **Vincent Rijmen** yaratgan. G'alaba sababi: xavfsizlik, tezlik, samaradorlik va moslashuvchanlikning eng yaxshi kombinatsiyasi.

01 · NIST TANLOVI

Tanlov mezonlari va nega ochiq o'tkazildi

Xavfsizlik

Barcha ma'lum kriptotahlil usullariga (chiziqli, differensial va h.k.) chidamlilik

Tezlik

8-bitli smart-kartadan 64-bit serverga qadar — dasturiy VA apparat samaradorligi

Moslashuvchanlik

Turli kalit o'lchamlari, oddiy apparat sxemasi, parallellashtirish imkoniyati

> ***«Kriptotizim algoritmi dushman qo'liga to'liq tushib qolsa ham, kalit maxfiy qolar ekan, xavfsiz bo'lishi kerak»***
>
> Ogyust Kerkhoffs · 1883, o'tgan darsdan

Rijndael to'liq ochiq e'lon qilindi. **3 yil davomida** butun dunyo kriptograflari uni buzishga urindi. Agar xavfsizlik sirni saqlashga tayansa, bu jarayonning o'zi imkonsiz bo'lardi — ochiqlik ishonchni isbotladi, yashirmadi.

02 · AES PARAMETRLARI

AES pasporti: blok o'zgarmas, kalit o'zgaruvchan

KALIT UZUNLIGI	N_k (SO'ZLAR)	N_r (RAUNDLAR)
128 bit	4	**10**
192 bit	6	**12**
256 bit	8	**14**

DIQQAT

Blok o'lchami **har doim 128 bit** (N_B = 4 so'z) — kalit qanchalik uzun bo'lmasin, holat matritsasi doim 4×4 bayt.

Kalit uzunligi ortishi **faqat raundlar sonini** oshiradi — kriptotahlilga qarshi qo'shimcha xavfsizlik zahirasi.

AES va DES: ikki avlod

DES

1977–2005

64 bit blok · 56 bit effektiv kalit · 16 raund · **Feystel tarmog'i**

kalit fazosi: $2^{56} \approx 7{,}2 \times 10^{16}$

AES-128

2001 –

128 bit blok · 128 bit kalit · 10 raund · **SP-tarmoq** (Feystel emas)

kalit fazosi: $2^{128} \approx 3{,}4 \times 10^{38}$

NIST hisobiga ko'ra, 128-bitli kalit uchun **340 undesillion** (340 dan keyin 36 ta nol) mumkin bo'lgan kalit mavjud — DES ning kalit fazosidan trillion-trillion marta katta.

16 baytdan 4×4 bayt jadvaliga

128 bitli blok — 16 bayt: $in_0 … in_{15}$. Har bayt holat matritsasi `s[r][c]` ning bitta katakchasiga **ustunlar bo'yicha** joylashadi:

$$s_{r,c} = \text{in}_{r+4c}$$

1. Avval 0-ustun to'ladi ($in_0 \to in_3$), keyin 1-ustun ($in_4 \to in_7$), va h.k.

2. Bu tartib keyinchalik ShiftRows (qator) va MixColumns (ustun) uchun muhim

to'ldirish tartibi — in_i raqami

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}

0-ustun (qizil) → 1-ustun (yashil) → …

Aniq misol: bitta 128-bitli blok

FIPS-197 standartining rasmiy misoli — darsning oxirigacha shu blokni kuzatamiz

ochiq matn (16 bayt, chiziqli)

```
32 43 f6 a8 88 5a 30 8d
31 31 98 a2 e0 37 07 34
```

kalit (16 bayt)

```
2b 7e 15 16 28 ae d2 a6
ab f7 15 88 09 cf 4f 3c
```

holat matritsasi s[r][c]

32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

0-ustun = 32,43,f6,a8 ($in_0 \ldots in_3$)

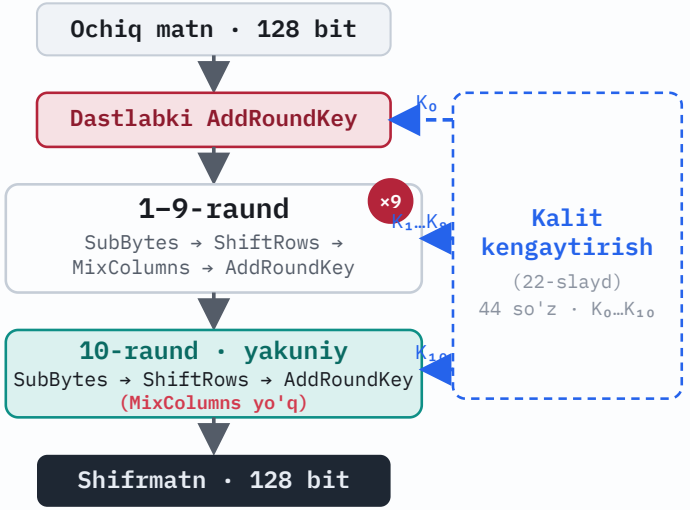
AES-128 ning to'liq yo'li

K_0 dastlabki AddRoundKey — bu bilan aslida "0-raund" boshlanadi

×9 to'liq raund: SubBytes → ShiftRows → MixColumns → AddRoundKey

10 yakuniy raund — **MixColumns yo'q** (10-slaydda sababi)

Kalit kengaytirish 11 ta 128-bitli raund kaliti (K_0...K_{10}) tayyorlaydi — 22-slaydda.



Nega yakuniy raundda MixColumns yo'q?

1–9-RAUND TARKIBI

SubBytes → ShiftRows → **MixColumns** → AddRoundKey

10-RAUND TARKIBI

SubBytes → ShiftRows → AddRoundKey

MixColumns bir ustundagi baytlarni **keyingi raundning** ShiftRows/SubBytes qadamlariga tayyorlab, diffuziyani kengaytiradi.

Yakuniy raunddan keyin — **keyingi raund yo'q**. Diffuziya qo'shishning ma'nosi qolmaydi: uni qo'shish faqat qo'shimcha (keyin teskarilanishi kerak bo'lgan) qadam bo'lardi, xavfsizlikka hech narsa qo'shmasdi.

Natija: shifrlash va deshifrlash sxemalari bir-biriga aniq mos tushadigan, ortiqcha qadamsiz struktura.

1) SubBytes — yagona nochiziqli qadam

Har bayt mustaqil ravishda 256-katakli S-box jadvalidan almashtiriladi

S-box, 5-qator (kirish yuqori nibl = 5)

5_	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf

kirish **0x53** → qator 5, ustun 3 → chiqish **0xED**

Har bayt qo'shnisiga bog'liq emas — **bayt-bayt, mustaqil** almashtirish. Formula yo'q, faqat jadval qidiruvi.

SubBytes dan OLDIN — 1-raund kirishi

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

dastlabki AddRoundKey natijasi (9-slayd)

SubBytes qo'llangandan keyin

OLDIN

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

→

KEYIN — SubBytes natijasi

d4	**e0**	**b8**	**1e**
27	**bf**	**b4**	**41**
11	**98**	**5d**	**52**
ae	**f1**	**e5**	**30**

s[0][0]: 0x19 → 0xD4 (S-boxning boshqa katagi)

S-box — **bijektiv** (har kirish uchun yagona chiqish, teskarilanadi) va **nochiziqli** — oddiy algebraik formula bilan ifodalanmaydi.

Aynan shu nochiziqlilik AES ni chiziqli va differensial kriptotahlildan himoya qiladi — DES dagi S-boxlarning vazifasi bilan bir xil g'oya.

2) ShiftRows — qatorlarni siljitish

- **r=0** siljish yo'q
- **r=1** 1 bayt chapga (siklik)
- **r=2** 2 bayt chapga (siklik)
- **r=3** 3 bayt chapga (siklik)

Qoida: **r-qator, r bayt** chapga siljiydi. SubBytes joyni o'zgartirmagani uchun, kirish — SubBytes natijasi (12-slayd).

ShiftRows dan OLDIN

rangli qatorlar — har biri o'z siljishini kutmoqda

ShiftRows qo'llangandan keyin

KEYIN — har qator siljigan

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

har rangli qator o'zining siljishi bilan chapga ko'chdi

ShiftRows — bu **o'rin almashtirish**: bayt qiymatlari o'zgarmaydi, faqat pozitsiyasi o'zgaradi. Yolg'iz o'zi qo'shimcha kriptografik kuch bermaydi.

Lekin muhim vazifasi bor: endi har ustunda **4 xil asl ustundan** bitta bayt bor. Keyingi qadam — MixColumns — shu ustunlarni aralashtiradi, shuning uchun baytlar butun holat bo'ylab tarqaladi.

ShiftRows'siz MixColumns doim FAQAT bir xil asl ustun ichida aralashtirib qolardi — diffuziya butun blokka yoyilmasdi.

3) MixColumns — ustunlarni GF(2⁸) da aralashtirish

Har ustun 4-baytli vektor sifatida qaraladi va **qat'iy 4×4 matritsaga** (GF(2^8) arifmetikasida) ko'paytiriladi:

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

bu son "×2" va "×3" — oddiy ko'paytirish emas, GF(2^8) ko'paytirish (19–20-slayd)

MixColumns dan OLDIN (ShiftRows natijasi)

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

0-ustun (qizil belgilangan) — keyingi slaydda to'liq hisoblaymiz

Bitta ustun uchun to'liq hisob

0-ustun: d4, bf, 5d, 30 (GF(2⁸) arifmetikasida)

$$s'_0 = (02\cdot d4) \oplus (03\cdot bf) \oplus (01\cdot 5d) \oplus (01\cdot 30) = B3 \oplus DA \oplus 5D \oplus 30 = \mathbf{04}$$

$$s'_1 = (01\cdot d4) \oplus (02\cdot bf) \oplus (03\cdot 5d) \oplus (01\cdot 30) = \mathbf{66}$$

$$s'_2 = (01\cdot d4) \oplus (01\cdot bf) \oplus (02\cdot 5d) \oplus (03\cdot 30) = \mathbf{81}$$

$$s'_3 = (03\cdot d4) \oplus (01\cdot bf) \oplus (01\cdot 5d) \oplus (02\cdot 30) = \mathbf{e5}$$

Har "×N" — GF(2⁸) ko'paytirish, ⊕ — XOR. Qolgan 3 ustun xuddi shu qoida bilan hisoblanadi.

KEYIN — MixColumns natijasi

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

0-ustun natijasi: 04, 66, 81, e5

4) AddRoundKey — kalit bilan yagona bog'liqlik

Holat va raund kaliti — **bayt-baytlab XOR**. Shu qadar sodda, lekin bu — algoritmning butun xavfsizligi kalitga qanday bog'lanishi.

MUHIM FAKT

SubBytes, ShiftRows, MixColumns — **hech biri sirga bog'liq emas**, hammasi ommaviy. Faqat AddRoundKey maxfiy kalitni ishlatadi.

1-raund kaliti (kalit kengaytirishdan, 22-slayd)

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05

AddRoundKey dan OLDIN (MixColumns natijasi)

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

AddRoundKey natijasi — 1-raund yakunlandi

04⊕a0=a4, e0⊕88=68, 48⊕23=6b, 28⊕2a=02, … — har bayt mustaqil XOR

1-raund YAKUNIY holati

a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49

Bu holat endi **2-raundning SubBytes** qadamiga kiradi — xuddi shu to'rt amal yana 8 marta (2–9-raund) takrorlanadi, so'ng 10-raund (MixColumons'siz) bilan yakunlanadi.

9-slaydning pipeline diagrammasiga qaytib qarang — endi har qutining ichida nima bo'layotgani aniq.

Nega oddiy mod-256 arifmetika yetarli emas

1. MixColumns'ning teskarisi bo'lishi uchun (deshifrlash!) har nolmas bayt **ko'paytirishga nisbatan teskari elementga** ega bo'lishi kerak

2. Mod-256 buni bermaydi: 256 = 2^8 tub son emas (masalan, $2 \times 128 \equiv 0 \pmod{256}$)

3. Yechim: bayt = GF(2) ustidagi 8-darajali polinom; qo'shish = XOR, ko'paytirish — keyingiga keltirilmaydigan polinom bo'yicha qoldiq

Natija — **GF(2^8)**: 256 elementli haqiqiy maydon, har nolmas element teskariga ega.

keyingiga keltirilmaydigan modul

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

bayt ko'rinishida: **0x11B** (9 bit: 1 0001 1011)

m(x) faktorlanmaydi — shuning uchun GF(2)[x]/m(x) haqiqiy maydon hosil qiladi

Amaliy qoida: 0x02 ga ko'paytirish (xtime)

1. Baytni 1 bitga chapga siljitish (= x ga ko'paytirish)
2. Agar asl baytning 7-biti 1 bo'lsa (ortiqcha chiqdi) — natijani **0x1B** bilan XOR qilish

0x1B = x^4+x^3+x+1 — chunki m(x)=0 bo'lgani uchun $x^8 \equiv x^4+x^3+x+1$ (GF(2) da XOR bilan teng)

3 ga ko'paytirish: $3 \cdot x = (2 \cdot x) \oplus x$ — faqat xtime + XOR, umumiy ko'paytiruvchi shart emas

misol — 16-slayddagi hisobdan

```
0xD4 = 1101 0100 (bit7=1)
chapga siljish → 1010 1000 = 0xA8
bit7=1 edi → XOR 0x1B
0xA8 ⊕ 0x1B = 0xB3
natija: 0x02 · 0xD4 = 0xB3
```

aynan shu qiymat 16-slayddagi s_0' hisobida ishlatilgan

S-box tasodifiy emas — ikki matematik qadam

1 GF(2^8) da teskari element

Kirish bayt b uchun b^{-1} topiladi: $b \cdot b^{-1} = 1$ (GF(2^8) da). Maxsus holat: `0x00` → `0x00` (teskarisi yo'q).

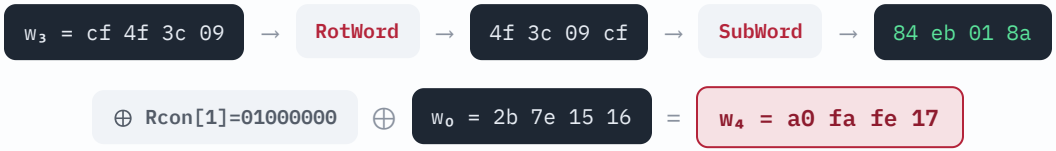
2 Affin almashtirish (GF(2) ustida)

Teskari elementning bit vektoriga qat'iy chiziqli o'zgartirish + o'zgarmas `0x63` XOR qilinadi. Bu qadam nol nuqtalarni yo'q qiladi va S-box'ni algebraik hujumlardan yanada mustahkamlaydi.

S-box — **256 katakli, ammo tasodifiy emas**: ikki qadam orqali hisoblab chiqarilgan, ochiq formula. 4-slaydda ko'rgan tamoyilning davomi — hech narsa yashirin emas, hammasi tekshirilishi mumkin.

128 bitdan 44 so'zgacha

$N_β·(N_r+1) = 4·11 = 44$ so'z — har raund uchun 4 so'zlik (128-bit) o'z kaliti



$$i \not\equiv 0 \pmod 4 :\ w_i = w_{i-1} \oplus w_{i-4} \qquad i \equiv 0 \pmod 4 :\ w_i = \mathrm{SubWord}(\mathrm{RotWord}(w_{i-1})) \oplus \mathrm{Rcon}[i/4] \oplus w_{i-4}$$

Rcon so'zining birinchi bayti (RC), i=1…10

01	02	04	08	10	20	40	80	1b	36

Nega AES Feystel emas — deshifrlash alohida

Shifrlash (bir raund)	Deshifrlash (bir raund)
SubBytes	InvShiftRows
↓	↓
ShiftRows	InvSubBytes
↓	↓
MixColumns	AddRoundKey
↓	↓
AddRoundKey	InvMixColumns

Feystelda faqat yarim blok F ga kirardi — F teskarilanishi **shart emas edi**, shuning uchun shifrlash va deshifrlash bir xil sxema edi.

> AES — SP-tarmoq: **butun blok** har raundda o'zgaradi, "tegilmagan yarim" yo'q. Shuning uchun har qatlam (SubBytes, ShiftRows, MixColumns) **o'zi** teskarilanadigan bo'lishi shart.

`InvMixColumns matritsasi: [0E 0B 0D 09 / 09 0E 0B 0D / 0D 09 0E 0B / 0B 0D 09 0E]`

25 yildan ortiq sinovdan o'tgan algoritm

BIKLIKA HUJUMI · 2011

$2^{126.1}$

AES-128 ning 2^{128} o'rniga — nazariy eng yaxshi natija. Amalda hech qanday farqi yo'q: baribir hisoblab bo'lmaydigan son.

To'liq raundli AESga

amaliy kalit-qidirish hujumi **topilmagan** — yagona-kalit sharoitida, 25+ yillik ochiq kriptotahlildan keyin

HAQIQIY XAVF

Algoritmning o'zi emas — **yon kanal hujumlari**: vaqt o'lchash (cache-timing), quvvat iste'moli tahlili (DPA), elektromagnit chiqindi. Bular matematikaga emas, **implementatsiyaga** hujum qiladi.

AES-NI va Python'da bitta blokni shifrlash

```python
fromCrypto.Cipher import AES
fromCrypto.Random import get_random_bytes

# 128-bitli tasodifiy kalit
key = get_random_bytes(16)
cipher = AES.new(key, AES.MODE_ECB)

# aynan BITTA 16-baytli blok — bugungi holat matritsasi
block = bytes.fromhex("3243f6a8885a308d313198a2e0370734")
ct = cipher.encrypt(block)
print(ct.hex())

# pycryptodome ichida — bor bo'lsa — AES-NI avtomatik ishlatiladi
```

```
$ python aes_block.py
kalit: <16 tasodifiy bayt>
shifrmatn: 39f5...0b32
```

AES-NI — Intel/AMD protsessorlaridagi maxsus buyruqlar (2010, Westmere'dan). S-box mantiqiy sxema sifatida ishlaydi, xotiradan jadval o'qilmaydi — bu ham tezlik, ham cache-timing hujumlariga tabiiy chidamlilik beradi.

MODE_ECB faqat **bitta blok** uchun ishlatildi — o'quv maqsadida. Haqiqiy xabarlar bir nechta blokdan iborat va ECB ularni xavfsiz bog'lamaydi.

Bitta blokdan xabargacha

1. AES — SP-tarmoq, Feystel emas: butun 128-bitli blok har raundda o'zgaradi
2. Holat — 4×4 bayt matritsasi, ustunlar bo'yicha to'ldiriladi
3. Har raund: SubBytes → ShiftRows → MixColumns → AddRoundKey; yakuniy raundda MixColumns yo'q
4. GF(2^8) va S-box'ning affin qurilishi — hech narsa tasodifiy, hammasi hisoblab chiqarilgan
5. 25+ yildan beri to'liq raundli AESga amaliy hujum yo'q — asosiy xavf yon kanal, algoritmning o'zi emas

KO'PRIK

Bloklardan uzun xabarni qanday shifrlaymiz?

AES bitta 128-bitli blokni shifrlaydi. Haqiqiy xabarlar ko'p blokdan iborat — ularni xavfsiz bog'lash uchun **shifrlash rejimlari** (ECB, CBC, CTR, GCM) kerak. Keyingi mavzu.

Esanov Otabek

SamDU · Sun'iy intellekt va raqamli texnologiyalar fakulteti
Boshqaruv nazariyasi va axborot xavfsizligi kafedrasi

GitHub github.com/EsanovOtabek
Telegram @esanovme
Email esanovotabek688@gmail.com
Tel +998 93 236 04 33