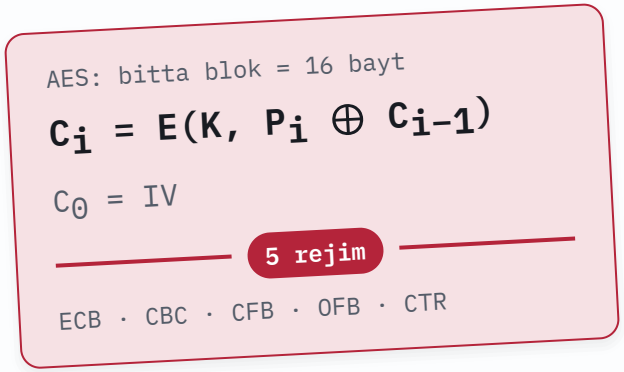
Sharof Rashidov nomidagi Samarqand davlat universiteti

KIBERXAVFSIZLIK ASOSLARI · 8-MA'RUZA

Blokli shifrlash rejimlari

Bitta blokdan cheksiz uzunlikdagi ma'lumotgacha: ECB, CBC, CFB, OFB va CTR

Axborot xavfsizligi yo'nalishi talabalari uchun



Muallif: **Esanov Otabek**

2026-yil 1-sentyabr

REJA · 8-MA'RUZA

Bitta blokdan butun faylgacha

1. Muammo: blok shifr nega yetarli emas
2. Padding: PKCS#7 qanday ishlaydi
3. ECB: eng oddiy va eng xavfli rejim
4. CBC: zanjirlash bilan naqshni yashirish
5. Xato tarqalishi: bitta blok qancha uzoqqa ta'sir qiladi
6. CFB, OFB, CTR: blok shifrni oqimga aylantirish
7. Rejimlarni solishtirish va IV/nonce qoidalari
8. Amaliyot: Python kodi va qaror daraxti

Har rejimda ikkita savolga e'tibor bering: bloklar bir-biriga qanday bog'langan va bitta xato qancha uzoqqa tarqaladi.

01

Muammo: katta ma'lumotni qanday shifrlaymiz?

Blok shifr faqat bitta belgilangan uzunlikdagi blokni biladi — qolgani bizning zimmamizda

1 · MUAMMO

Blok shifr bitta blokni biladi, xolos

AES har chaqiriqda aynan **16 baytli** (128 bit) blokni shifrlaydi. Haqiqiy fayl yoki xabar deyarli hech qachon 16 bayt bo'lmaydi — odatda bir necha kilobayt yoki megabayt.

TA'RIF

Shifrlash rejimi (mode of operation) — bitta blokli algoritmni istalgan uzunlikdagi ma'lumotga qo'llash usuli: bloklar bir-biriga qanday bog'lanishi, IV/nonce qanday ishlatilishi va oxirgi to'liq bo'lmagan blok qanday hal qilinishi.

Rejim tanlovi algoritmning o'zidan (AES, DES) mustaqil — har qanday blok shifrga qo'llanadi.

Bir chaqiriqqa faqat bitta blok kiradi — qolganlarini qanday shifrlaymiz?

● 1 · MUAMMO

Oxirgi blok deyarli hech qachon to'liq emas

Xabar: **42 bayt**. Blok o'lchami: **16 bayt**. 42 ÷ 16 = 2 to'liq blok (32 bayt) + **10 bayt qoldiq**.

> Blok shifr faqat **to'liq** blokni shifrlay oladi — 10 baytni AES bilan to'g'ridan-to'g'ri shifrlab bo'lmaydi. Yechim: **to'ldirish (padding)**.

● 1 · MUAMMO

PKCS#7: yetishmagan baytni bayt qiymati bilan to'ldirish

1. Necha bayt yetishmayotganini hisoblang: 16 − 10 = **6**
2. **Har yetishmagan joyga xuddi shu son (6) qiymatini yozing**
3. Natija: 10 bayt ma'lumot + oltita 0x06 bayt = **16 bayt to'liq blok**

Chekka holat: agar ma'lumot allaqachon blok chegarasida tugagan bo'lsa ham, butun bir qo'shimcha blok (16 ta 0x10 bayti) qo'shiladi — aks holda deshifrlashda padding chegarasi noaniq qoladi.

02

ECB — Elektron kod kitobi

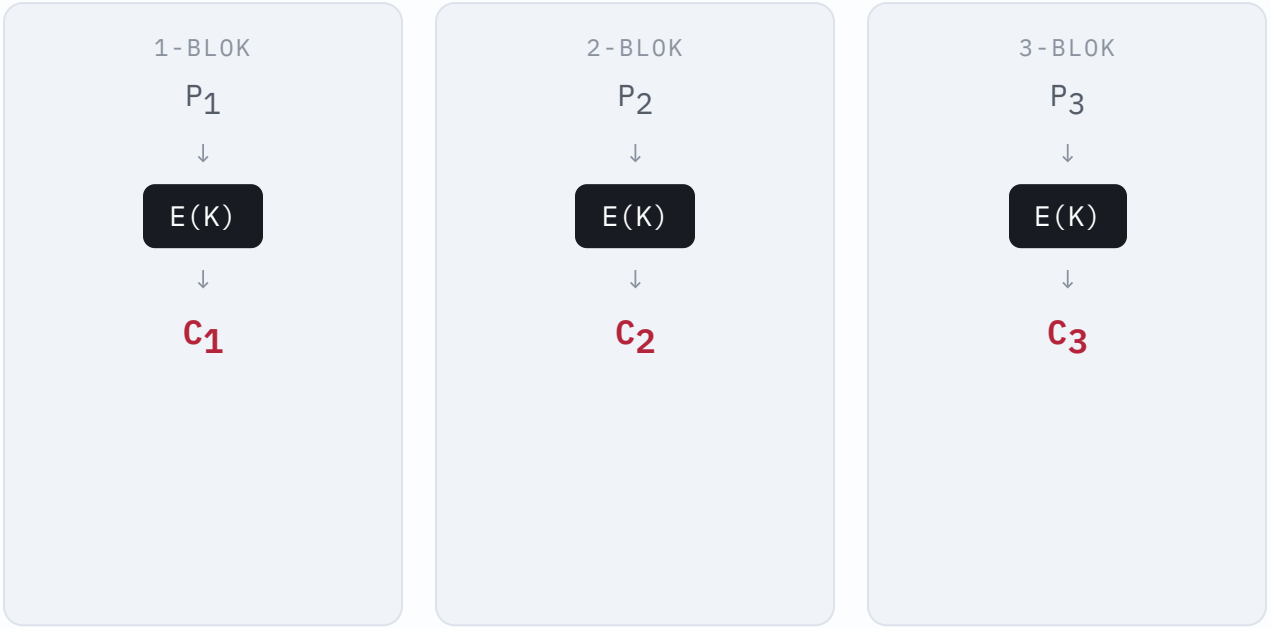
Eng sodda yechim: har blokni alohida-alohida shifrlash

2 · ECB

Har blok — mustaqil, bir xil kalit

$C_i = E(K, P_i)$

MUSTAQIL BLOKLAR — PARALLEL BAJARILADI

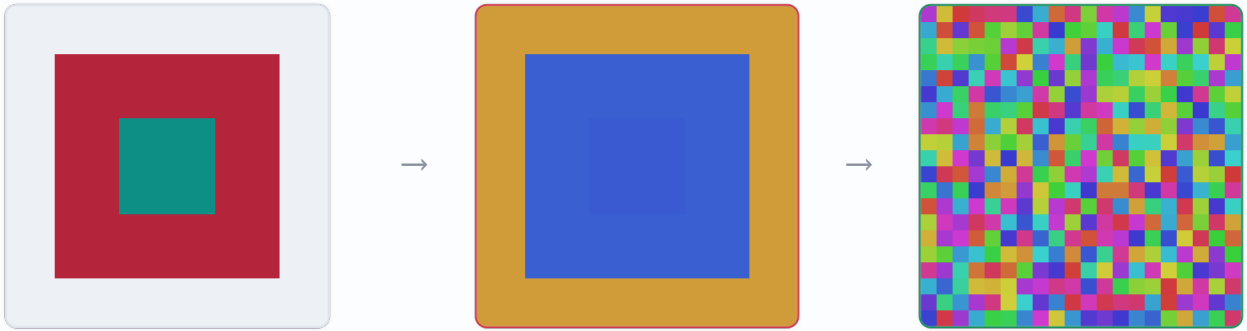


✓ **Afzalligi:** to'liq parallel — barcha bloklarni bir vaqtda shifrlash/deshifrlash mumkin, tasodifiy kirish oson.

⚠ **Lekin:** bir xil P bloki → doim bir xil C bloki. Bu naqshni oshkor qiladi.

2 · ECB

Bir xil bloklar — bir xil shifr matn



Original (sintetik namuna) → ECB — naqsh saqlanadi → CBC — naqsh yo'qoladi

Bu real AES emas — blokning "bir xil kirish → bir xil chiqish" xususiyatini aniq ko'rsatish uchun soddalashtirilgan vizualizatsiya (har kvadrat — bitta "blok"). Original rasmda uchta tekis rangli hudud bor. **ECB** har hududni bitta yangi rangga aylantiradi — pozitsiyadan qat'i nazar bir xil kirish bir xil chiqish beradi, shuning uchun chegaralar aynan saqlanadi. **CBC**da esa har blok oldingisiga zanjirlangani uchun bir xil kirish ham turli chiqish beradi — shakl yo'qoladi.

03

CBC — Shifr-blok zanjiri

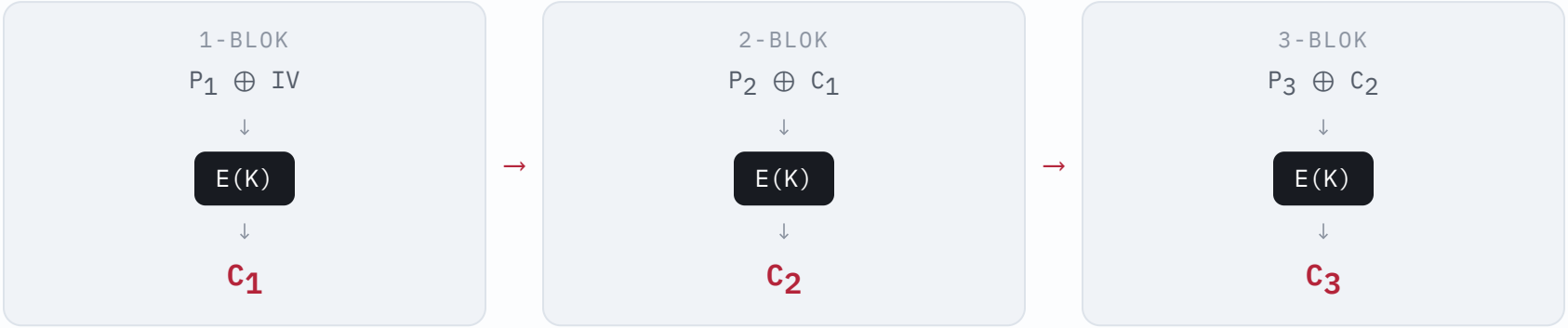
Har blokni oldingisiga bog'lab, naqshni yo'q qilish

3 · CBC

Shifrlash — ketma-ket zanjir

$C_i = E(K, P_i \oplus C_{i-1})$ · $C_0 = IV$

KETMA-KET — C_2 UCHUN AVVAL C_1 KERAK

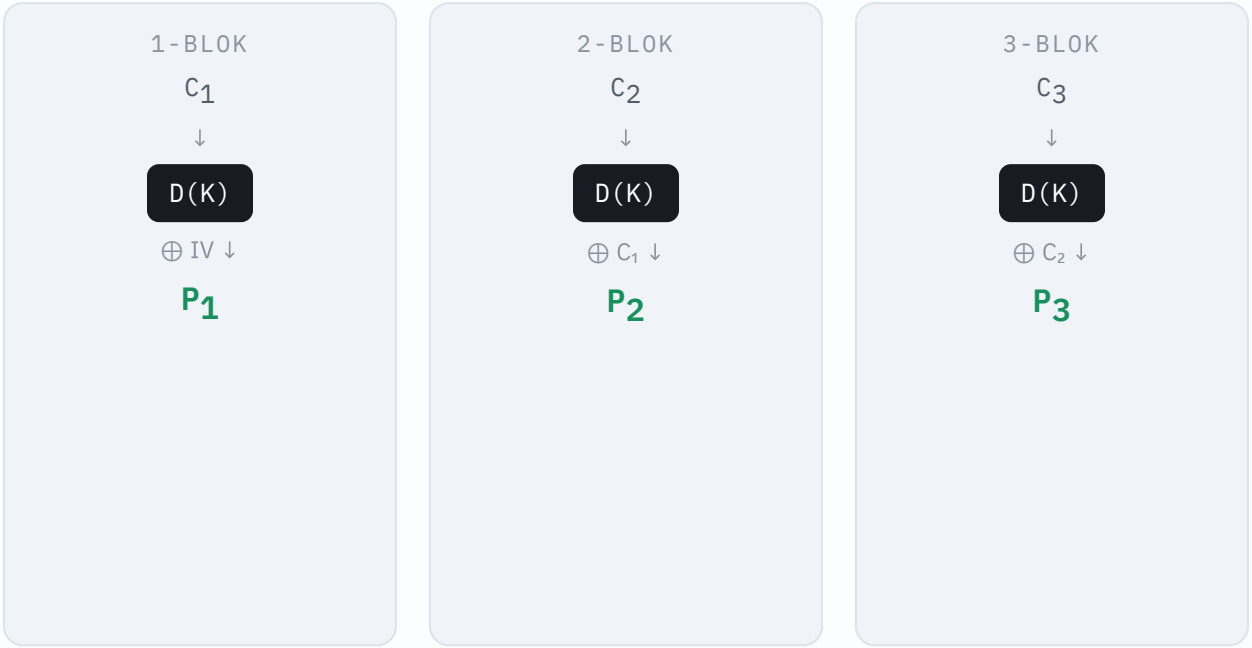


IV (boshlang'ich vektor) — xabar boshida ishlatiladigan, oldindan aytib bo'lmaydigan blok. Maxfiy bo'lishi shart emas — ciphertext bilan birga ochiq yuboriladi.

Deshifrlash — barchasi parallel

$$P_i = D(K, C_i) \oplus C_{i-1}$$

BARCHA C_i OLDINDAN MA'LUM — PARALLEL BAJARILADI



Nega parallel: deshifrlashda barcha C_i qiymatlari oldindan mavjud. $D(K,C_i)$ larni bir vaqtda hisoblab, keyin oddiy XOR qilish mumkin — hech biri boshqasini kutmaydi.

● 3 · CBC

Bitta buzilgan blok — ikkita natijaga ta'sir qiladi

1-BLOK

C_1 — toza

↓ D(K), ⊕IV

P_1 ✓

ta'sirlanmagan

2-BLOK

C_2 — ⚠ buzilgan

↓ D(K), ⊕C_1

P_2 ✗

to'liq buzildi — tasodifiy natija (avalanche)

3-BLOK

C_3 — toza

↓ D(K), ⊕C_2(buzilgan)

P_3 ⚠

faqat mos bit(lar) buziladi (bit-flip)

C_4, C_5…

P_4, P_5… ✓

hammasi qayta toza

Xulosa: bitta ciphertext blokining buzilishi aniq ikkita plaintext blokka ta'sir qiladi — biri to'liq (tasodifiy), ikkinchisi qisman (bit-flip). Shu bit-flip xususiyati tufayli CBC ma'lumot yaxlitligini kafolatlamaydi — buzilish yoki ataylab o'zgartirish alohida tekshirilishi (MAC) shart.

04

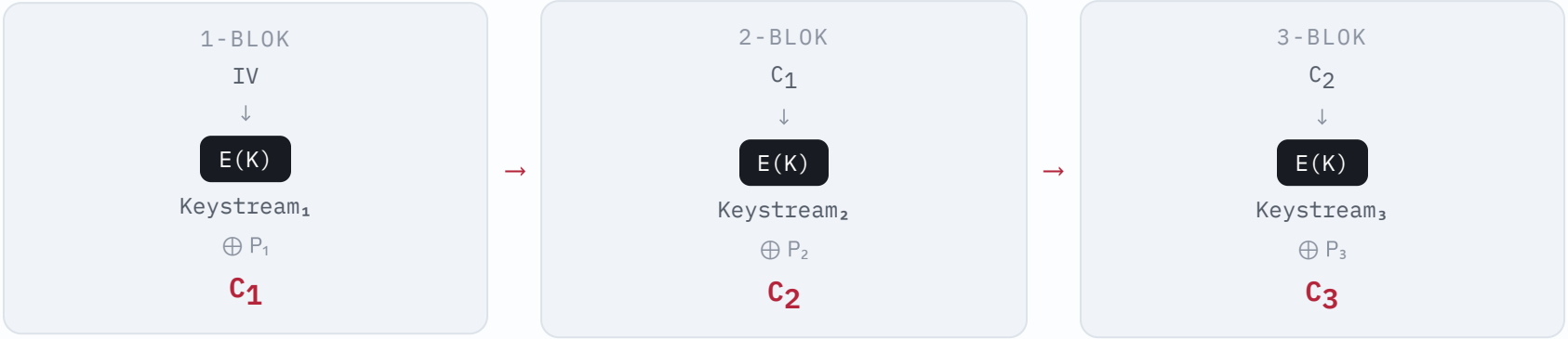
Oqimli rejimlar: CFB, OFB, CTR

Blok shifrni E(K) orqali keystream generatoriga aylantirish

● 4 · CFB

CFB — Shifr fikr-bildirish (Cipher Feedback)

$C_i = P_i \oplus E(K, C_{i-1})$ · $C_0 = IV$



Diqqat: har ikki tomonda ham faqat **E(K)** ishlatiladi, D(K) kerak emas — P_i ciphertext bilan XOR orqali olinadi.

Deshifrlash **parallel**: barcha C_i oldindan ma'lum. Padding **kerak emas** — oqim sifatida ishlaydi.

4 · OFB

OFB — Chiqish fikr-bildirish (Output Feedback)

$KS_i = E(K, KS_{i-1})$ · $C_i = P_i \oplus KS_i$

KS zanjiri faqat **kalit va IV**ga bog'liq — plaintext/ciphertextdan mustaqil. Keystreamni ma'lumot kelishidan oldin tayyorlab qo'yish (precompute) mumkin.

Eng kam xato tarqalishi: bitta ciphertext biti buzilsa, **faqat mos plaintext biti** buziladi — boshqa hech qanday blokka o'tmaydi.

● 4 · CTR

CTR — Hisoblagich rejimi (Counter)

$KS_i = E(K, Nonce \parallel i)$ · $C_i = P_i \oplus KS_i$

TO'LIQ PARALLEL — SHIFRLASH HAM, DESHIFRLASH HAM

Har blok mustaqil — faqat hisoblagich raqami farq qiladi. Ixtiyoriy blokka bevosita kirish (random access) mumkin. Zamonaviy standart tanlov (masalan AES-GCM ichida).

Shart: (kalit, nonce) jufti hech qachon takrorlanmasin — sababini keyingi bo'limda ko'ramiz.

05

Solishtirish va xavfsizlik qoidalari

Beshta rejim bir jadvalda, IV/nonce qoidalari va amaliyot

5 · SOLISHTIRISH

Beshta rejimni bir jadvalda solishtiramiz

Rejim	Shifrlash	Deshifrlash	Xato tarqalishi	Naqsh sizadimi	IV / Nonce	Padding
ECB	**Parallel**	**Parallel**	1 blok	**Ha (xavfli)**	Kerak emas	Kerak
CBC	Ketma-ket	**Parallel**	2 blok	Yo'q	IV	Kerak
CFB	Ketma-ket	**Parallel**	2 blok	Yo'q	IV	Kerak emas
OFB	Ketma-ket	Ketma-ket	1 bit	Yo'q	IV	Kerak emas
CTR ★	**Parallel**	**Parallel**	1 bit	Yo'q	Nonce	Kerak emas

IV va nonce: ikkita qoida, ikkita maqsad

IV

CBC, CFB, OFB — oldindan aytib bo'lmasin

Har xabar uchun tasodifiy yoki kamida bashorat qilib bo'lmaydigan bo'lishi kerak. **Maxfiy bo'lishi shart emas** — ciphertext bilan ochiq yuborilishi mumkin.

Nonce

CTR, OFB — hech qachon takrorlanmasin

Bir xil kalit bilan bir xil nonce ikki marta ishlatilsa, keystream takrorlanadi va xavfsizlik butunlay yo'qoladi. **Tasodifiy bo'lish shart emas** — oddiy hisoblagich bo'lishi mumkin, lekin **noyob bo'lishi shart**.

5 · XAVFSIZLIK

Nonce ikki marta ishlatilsa nima bo'ladi?

1	$C_1 = P_1 \oplus KS \quad C_2 = P_2 \oplus KS$	Bir xil (K, nonce) → bir xil KS
2	$C_1 \oplus C_2$	Tajovuzkor ikkala ciphertextni XOR qiladi
3	$= (P_1 \oplus KS) \oplus (P_2 \oplus KS)$	KS ikki marta qatnashadi
4	$= P_1 \oplus P_2$ — KS butunlay bekor bo'ldi	Kalit himoyasi yo'qoldi

Natija: ikkita plaintext bir-biri bilan XOR qilingan holda tajovuzkor qo'lida — til statistikasi yoki ma'lum matn yordamida ikkalasi ham tiklanishi mumkin. Bu xuddi bir martalik blokni (one-time pad) ikki marta ishlatishga teng — Sovet razvedkasining Venona shifrlarini AQSh tomonidan buzilishi ham aynan shu xatoga asoslangan edi.

5 · AMALIYOT

Bir xil rasm, ikki xil natija

ecb_vs_cbc.py — Python · pycryptodome

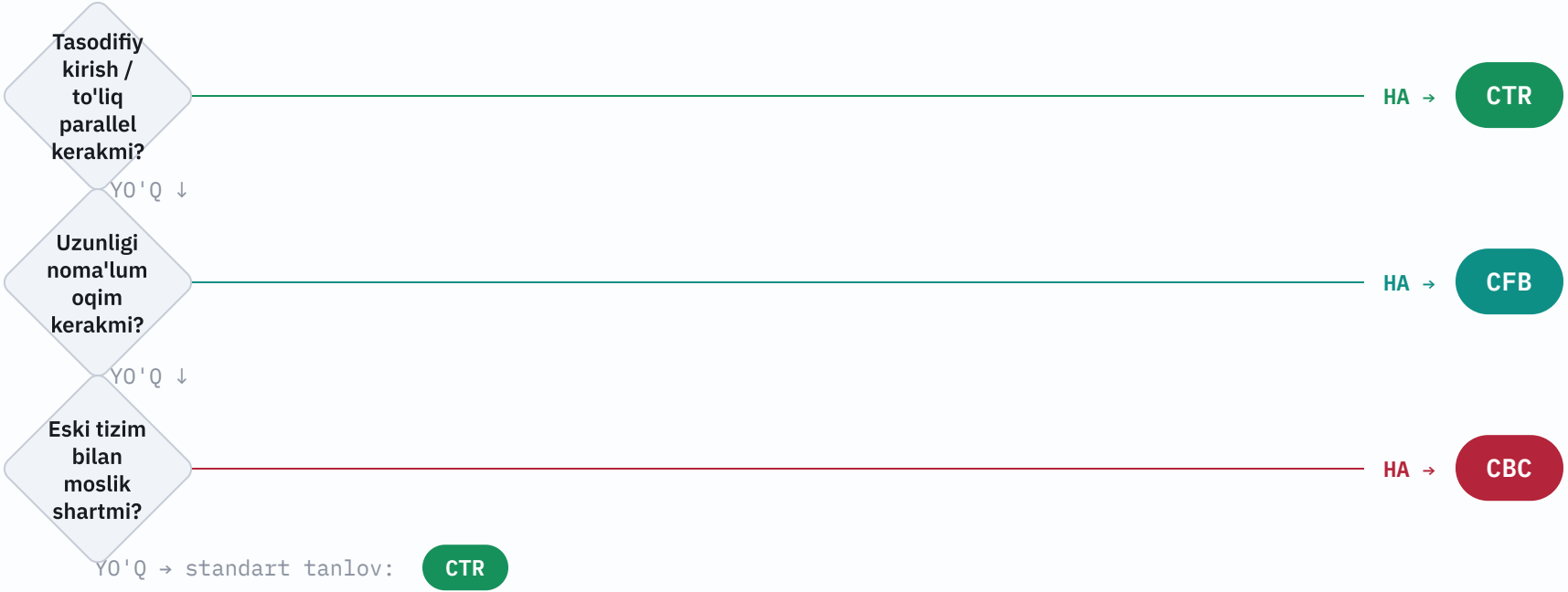
```python
from Crypto.Cipher import AES
from PIL import Image
key = b"0123456789abcdef"  # 16 bayt, AES-128
img = Image.open("rasm.bmp").convert("RGB")
data = img.tobytes()
data += bytes([0]) * (-len(data) % 16)  # blokka to'ldirish
ecb = AES.new(key, AES.MODE_ECB)
cbc = AES.new(key, AES.MODE_CBC, iv=b"\x00" * 16)
Image.frombytes("RGB", img.size, ecb.encrypt(data)).save("ecb.bmp")
Image.frombytes("RGB", img.size, cbc.encrypt(data)).save("cbc.bmp")
```

ecb.bmp — rasmning konturlari hali ko'rinadi (9-slayddagi demo bilan bir xil sabab).

cbc.bmp — toza, tekis shovqin. Hech qanday shakl ajratib bo'lmaydi.

5 · XULOSA OLDIDAN

Qaysi rejimni tanlash kerak?



ECB — javob qanday bo'lishidan qat'i nazar, ishlab chiqarishda hech qachon tanlanmaydi (naqshni oshkor qiladi).

8 · AMALIYOT VA XULOSA

Bitta g'oya, beshta rejim

1. Blok shifr faqat bitta blokni biladi — "rejim" uni istalgan uzunlikdagi ma'lumotga qo'llaydi
2. ECB — tez va oddiy, lekin naqshni oshkor qiladi; ishlab chiqarishda ishlatilmaydi
3. CBC — zanjirlash naqshni yashiradi; shifrlash ketma-ket, deshifrlash parallel; bitta xato 2 blokka tarqaladi
4. CFB/OFB/CTR — blok shifrni oqim shifrga aylantiradi, padding kerak emas; CTR to'liq parallel va zamonaviy standart
5. IV oldindan aytib bo'lmasin, nonce hech qachon takrorlanmasin — bu qoidalar buzilsa butun tizim ochiladi

KO'PRIK

Maxfiylik yetarlimi?

Rejimlar ma'lumotni maxfiy saqlaydi, lekin buzilgan yoki almashtirilganini bildirmaydi. Keyingi mavzu: **xesh funksiyalari** va ma'lumot yaxlitligi.

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