

Cryptographic Attacks in Modes



take multiple different other scenarios of your choice, and describe cryptographic attacks that can be possible in ECB, CBC, CFB, OFB, CTR modes

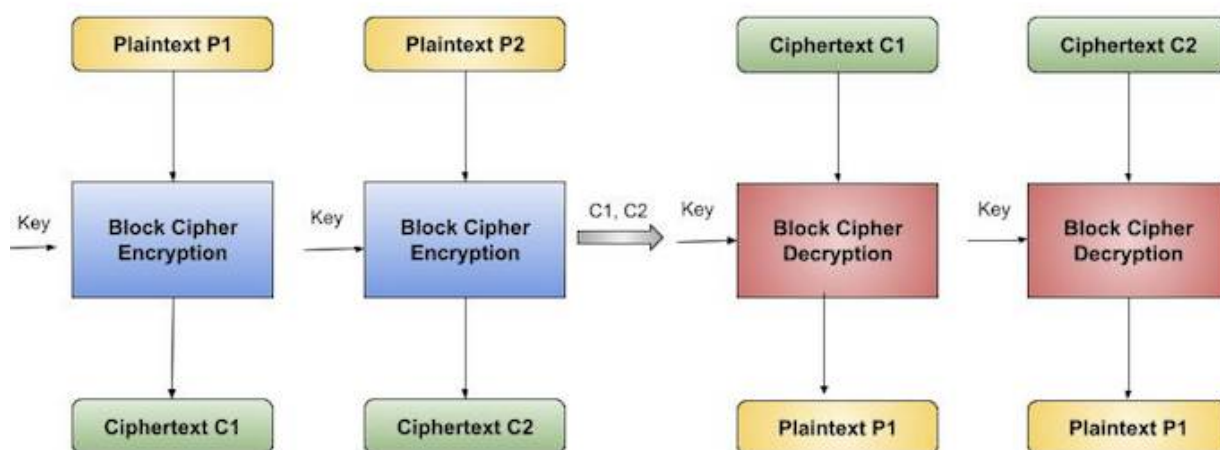
Below are **different real-world style scenarios** for each block cipher mode and the **cryptographic attacks possible** in each.

1 ECB Mode – Electronic Codebook

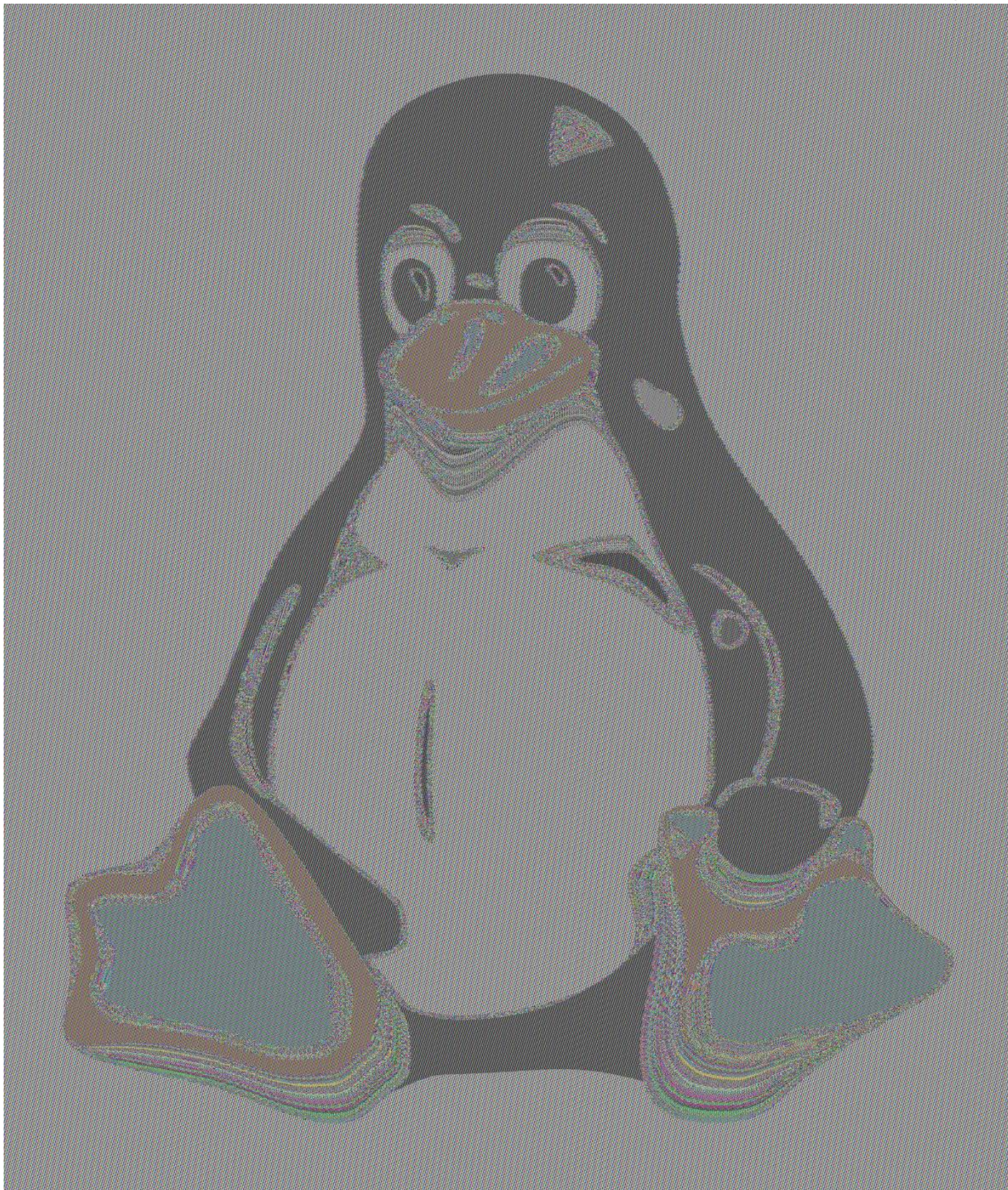
◆ Scenario: Encrypted Salary Database

A company stores employee salary records encrypted using AES-ECB.

Each record block = **Employee_ID | Salary | Bonus**.



<https://www.researchgate.net/publication/339988387/figure/fig4/AS%3A870178569072646%401584478146614/ECB-Mode-a-Identical-plaintext-under-the-same-key-results-in-identical-ciphertext-b.png>



Attacks Possible in ECB

(1) Pattern Leakage Attack

- Identical plaintext blocks produce identical ciphertext blocks.
- If many employees have the same salary, attacker sees repeating ciphertext.
- Reveals structural information.

👉 Confidentiality partially broken

(2) Block Reordering Attack

- Attacker swaps ciphertext blocks.
- Since blocks are independent, decryption still works.
- Salary values could be swapped between employees.

👉 No integrity protection.

(3) Cut-and-Paste Attack

- Attacker copies a ciphertext block (e.g., high salary block)
- Inserts into another employee record
- System decrypts successfully

2 CBC Mode – Cipher Block Chaining

◆ Scenario: Secure Banking Transaction Messages

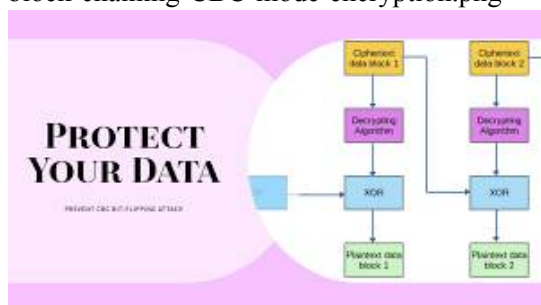
Message:

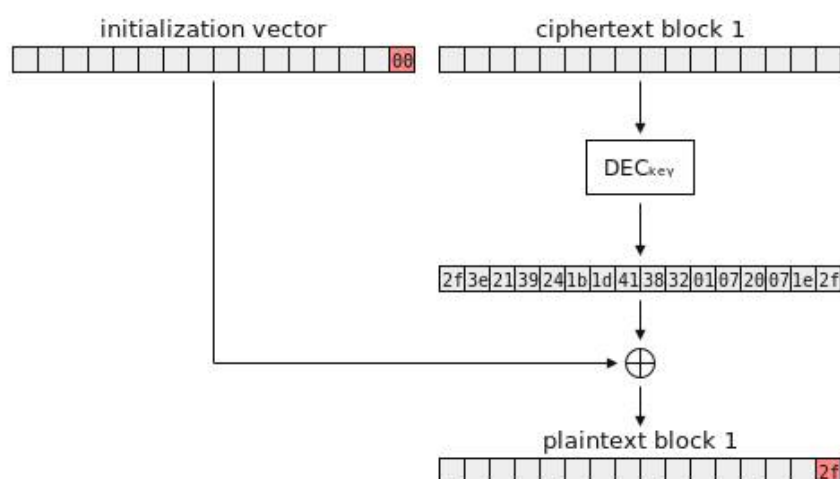
UserID | Amount | ToAccount

Encrypted using AES-CBC.



<https://www.researchgate.net/publication/215783767/figure/fig1/AS%3A394138559238144%401470981363092/Cipher-block-chaining-CBC-mode-encryption.png>





🔒 Attacks Possible in CBC

(1) Bit-Flipping Attack

- Attacker modifies ciphertext block C1.
- During decryption, corresponding bits in next plaintext block flip.
- Example: Amount = 1000 → changed to 9000.

👉 Controlled plaintext modification possible.

(2) Padding Oracle Attack

If system returns:

- "Invalid padding" vs
- "Invalid message"

Attacker uses padding response difference to:

- Gradually decrypt entire ciphertext.

Famous example: TLS attacks.

👉 Complete plaintext recovery possible.

(3) IV Manipulation Attack

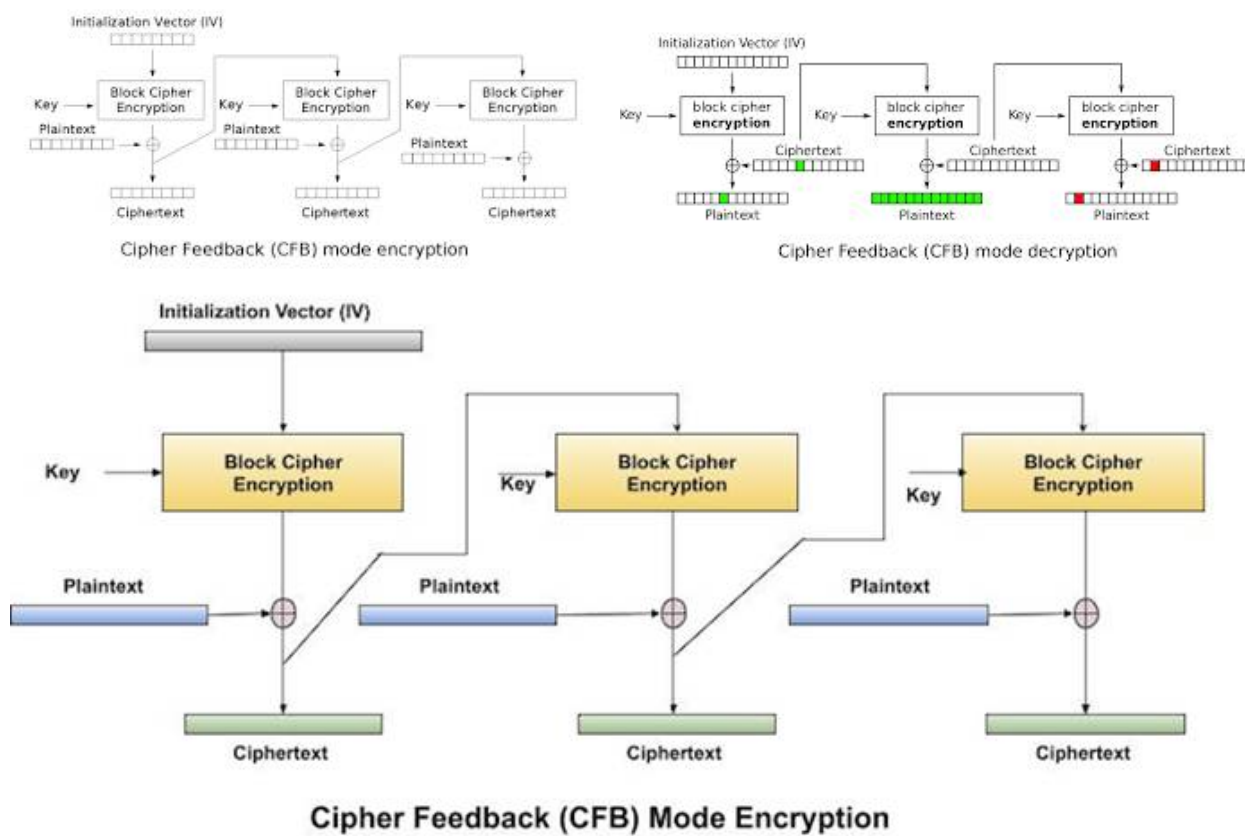
If IV is predictable:

- Attacker modifies IV to change first block plaintext.

3 CFB Mode – Cipher Feedback

◆ Scenario: Encrypted Real-Time Chat Application

Messages encrypted live using CFB.



Attacks Possible in CFB

(1) Bit-Flipping Attack

CFB behaves like stream cipher.

- Flip one ciphertext bit → flips one plaintext bit.
- Attacker can change:
"Transfer 100" → "Transfer 900"

No integrity protection.

(2) Error Propagation Exploitation

- Controlled corruption can distort next block
- Useful in targeted manipulation.

(3) Chosen Ciphertext Attack (if no authentication)

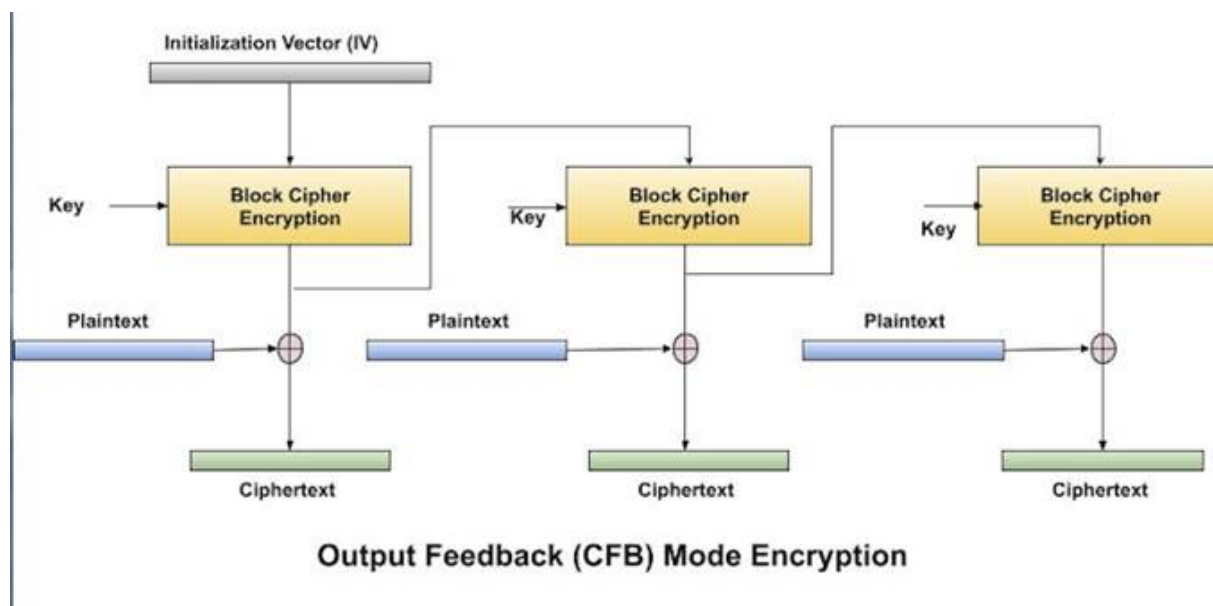
If attacker can inject ciphertext:

- Can manipulate resulting plaintext directly.

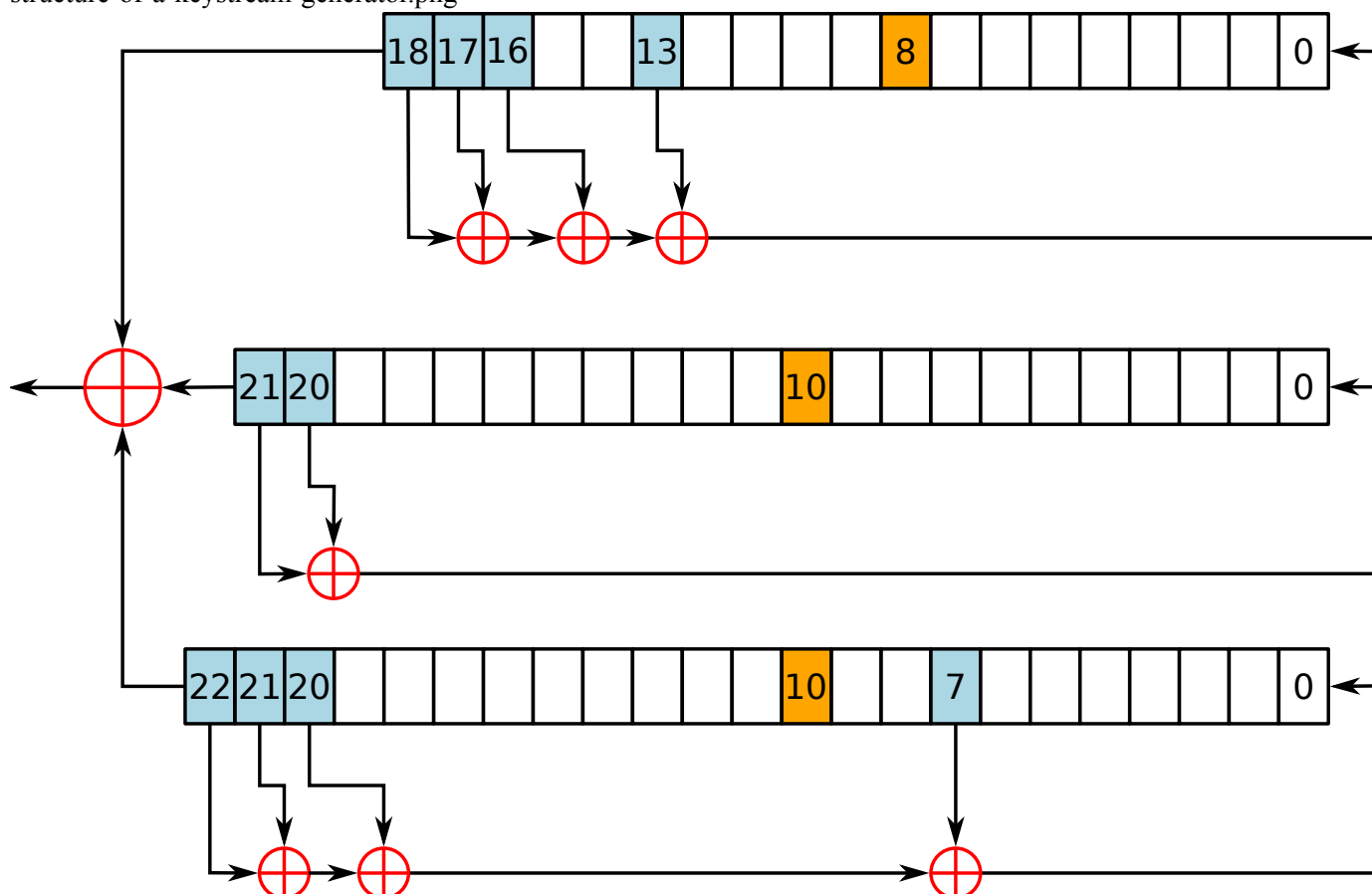
4 OFB Mode - Output Feedback

◆ Scenario: Satellite Communication Link

Data encrypted using OFB mode.



<https://www.researchgate.net/publication/267858656/figure/fig3/AS%3A295396514516994%401447439425105/Block-structure-of-a-keystream-generator.png>



Attacks Possible in OFB

(1) Keystream Reuse Attack (VERY SERIOUS)

If:

Same key + same IV used twice

Then:


$$C1 = P1 \oplus K$$

$$C2 = P2 \oplus K$$

Attacker computes:

$$C1 \oplus C2 = P1 \oplus P2$$

Now attacker can recover both messages.

 Same weakness as one-time pad reuse.

(2) Bit-Flipping Attack

- Flip ciphertext bit → flips same plaintext bit.
 - Fully controllable modification.
-

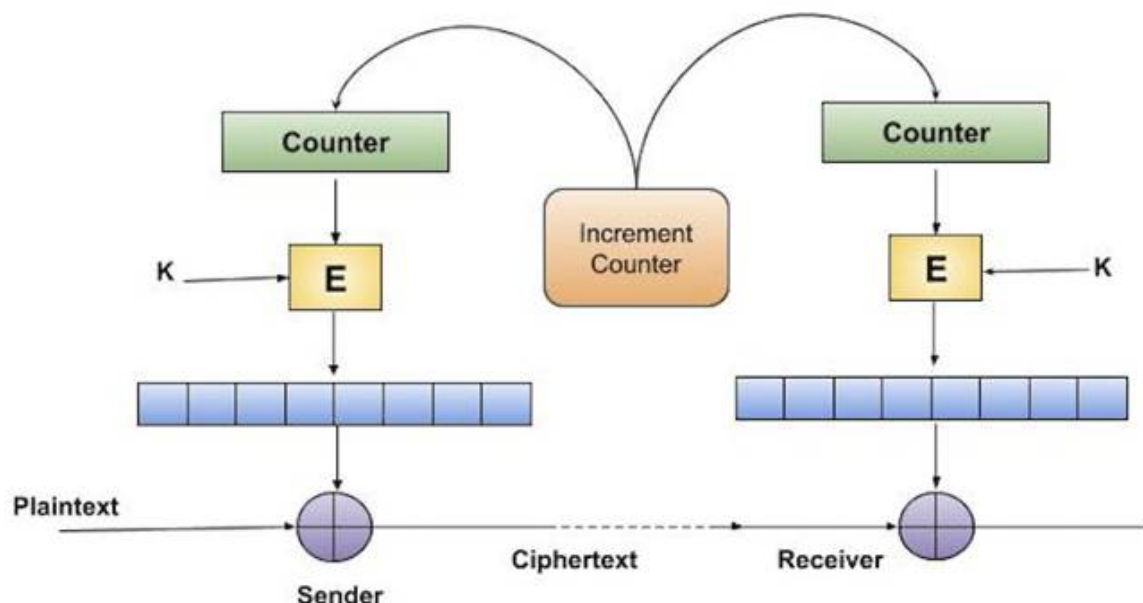
(3) Weak IV Attack

If IV predictable → keystream predictable.

CTR Mode – Counter Mode

◆ Scenario: Cloud Storage Encryption Service

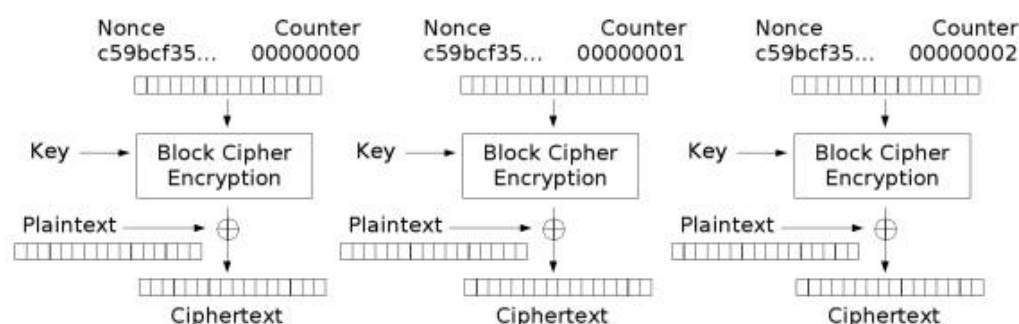
Files encrypted block-by-block using AES-CTR.



Counter (CTR) Mode



<https://www.researchgate.net/publication/351510159/figure/fig2/AS%3A1022469469122565%401620787126781/Counter-CTR-mode-of-operation-for-block-ciphers-Figure-is-from-17.png>



Counter (CTR) mode encryption

🔒 Attacks Possible in CTR

(1) Nonce Reuse Attack (Catastrophic)

If same:

Key + Nonce used twice

Then:

$$C1 \oplus C2 = P1 \oplus P2$$

Complete plaintext recovery possible.

👉 Very dangerous.

(2) Random Access Modification Attack

CTR is stream-like:

- Attacker flips ciphertext bit
- Corresponding plaintext bit flips

No chaining → precise tampering possible.

(3) Known Plaintext Attack Amplification

If part of plaintext known:

- Keystream recovered
- Can decrypt other parts

Summary Table

Mode	Confidentiality Issue	Integrity Issue	Serious Attack
ECB	Pattern leakage	Block swapping	Cut-paste
CBC	Padding oracle	Bit flipping	Padding Oracle
CFB	Stream manipulation	Bit flipping	Injection
OFB	Keystream reuse	Bit flipping	IV reuse
CTR	Nonce reuse	Bit flipping	Nonce reuse

Important Security Insight

- 👉 All these modes provide confidentiality only.
- 👉 None provide integrity.

That is why modern systems use:

- AES-GCM
- AES-CCM

These are **Authenticated Encryption (AEAD)** modes.

If you want, I can now give:

- 🔥 Extremely tricky exam-oriented attack explanations
- OR
- A short 6-mark crisp answer format
- OR
- Mathematical attack reasoning

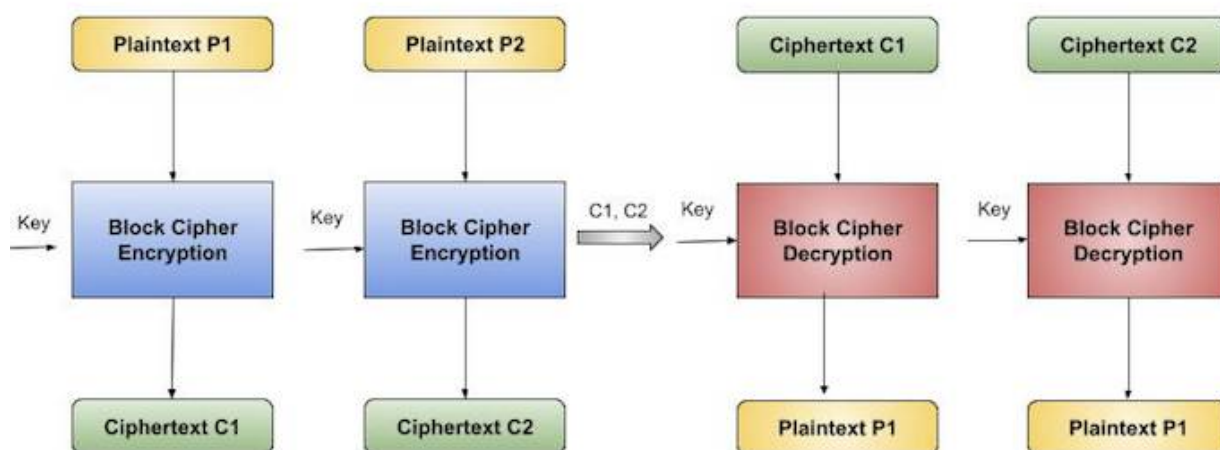
Tell me your exam style preference.



explain mitigations for each

Below are **clear, exam-ready mitigations** for attacks in **ECB, CBC, CFB, OFB, and CTR** modes.

1 ECB (Electronic Codebook)



<https://www.researchgate.net/publication/339988387/figure/fig4/AS%3A870178569072646%401584478146614/ECB-Mode-a-Identical-plaintext-under-the-same-key-results-in-identical-ciphertext-b.png>



Main Problems

- Pattern leakage
- Block swapping
- Cut-and-paste attack
- No integrity protection

✓ Mitigations

✓ 1. Do NOT Use ECB for Sensitive Data

The strongest mitigation is:

Completely avoid ECB.

ECB should never be used for real-world confidential data.

✓ 2. Use CBC / CTR / GCM Instead

Switch to secure modes like:

- CBC (with authentication)
 - CTR (with authentication)
 - GCM (preferred)
-

✓ 3. Add Integrity Protection

If forced to use ECB:

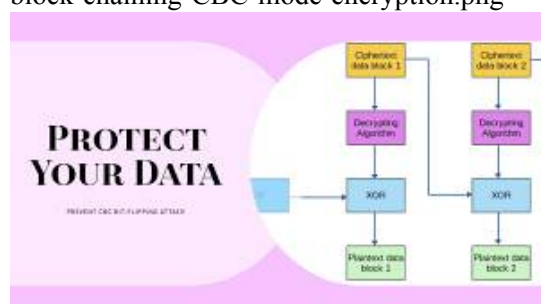
- Add HMAC over ciphertext
- Verify before decrypting

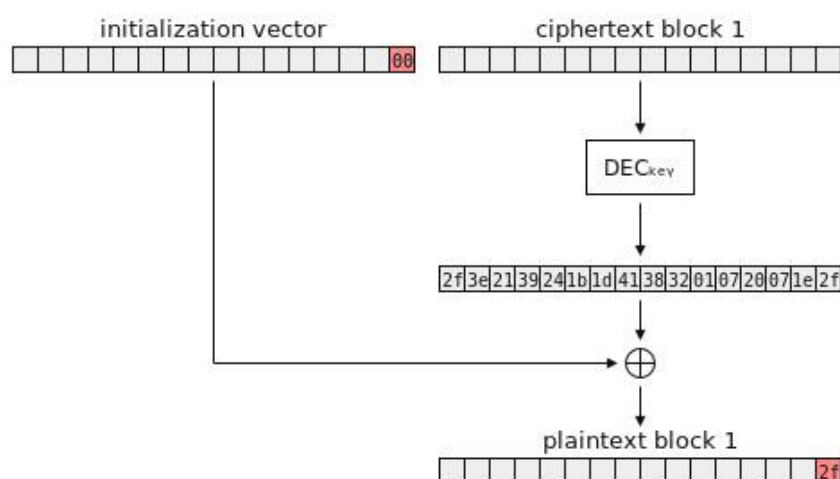
Prevents block swapping and cut-paste attacks.

2 CBC (Cipher Block Chaining)



<https://www.researchgate.net/publication/215783767/figure/fig1/AS%3A394138559238144%401470981363092/Cipher-block-chaining-CBC-mode-encryption.png>





🔒 Main Problems

- Bit-flipping attacks
- Padding oracle attacks
- IV manipulation

✅ Mitigations

✓ 1. Use Encrypt-then-MAC (Very Important)

Process:

1. Encrypt with CBC
2. Compute MAC on ciphertext
3. Verify MAC before decryption

Prevents:

- Bit-flipping
- Chosen ciphertext
- Padding oracle

✓ 2. Use Random, Unpredictable IV

- IV must be random for every message
- Never reuse IV with same key

Prevents IV manipulation attacks.

✓ 3. Uniform Error Messages

For padding oracle defense:

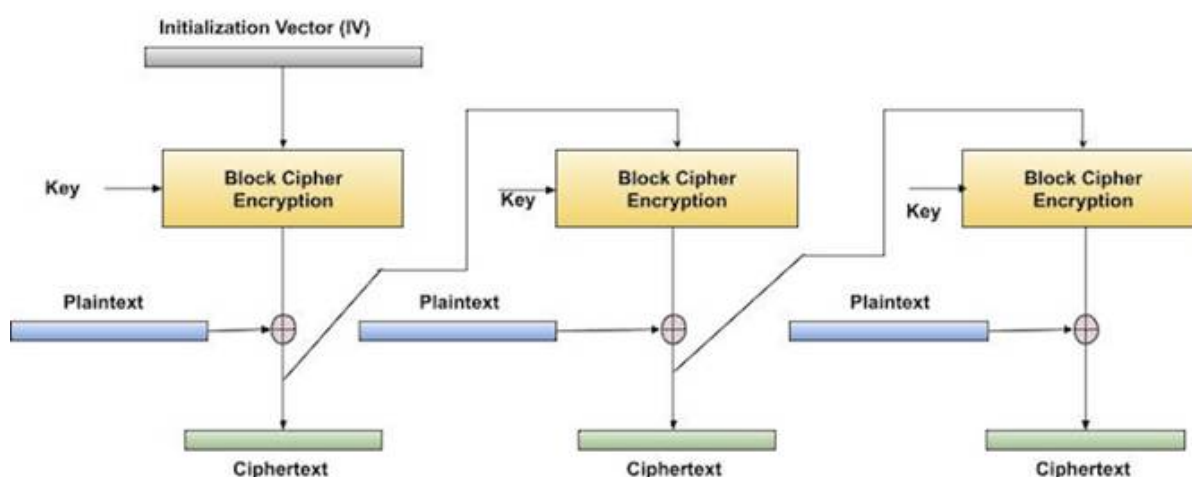
- Always return same error message
- Do not reveal padding vs MAC failure

Removes oracle side-channel.

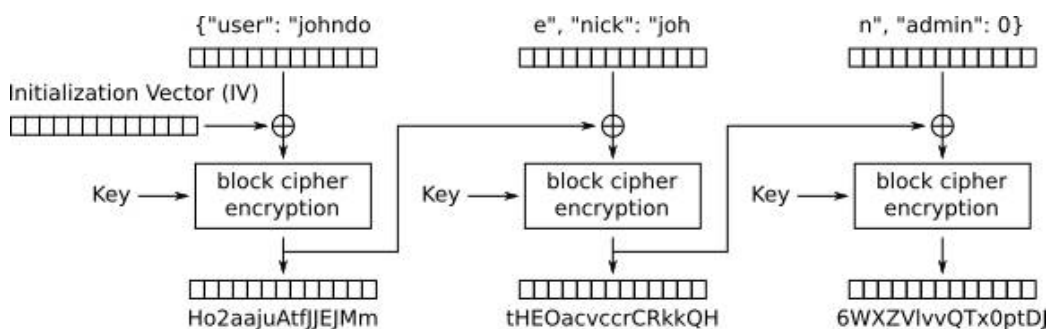
✓ 4. Prefer AES-GCM Instead of CBC

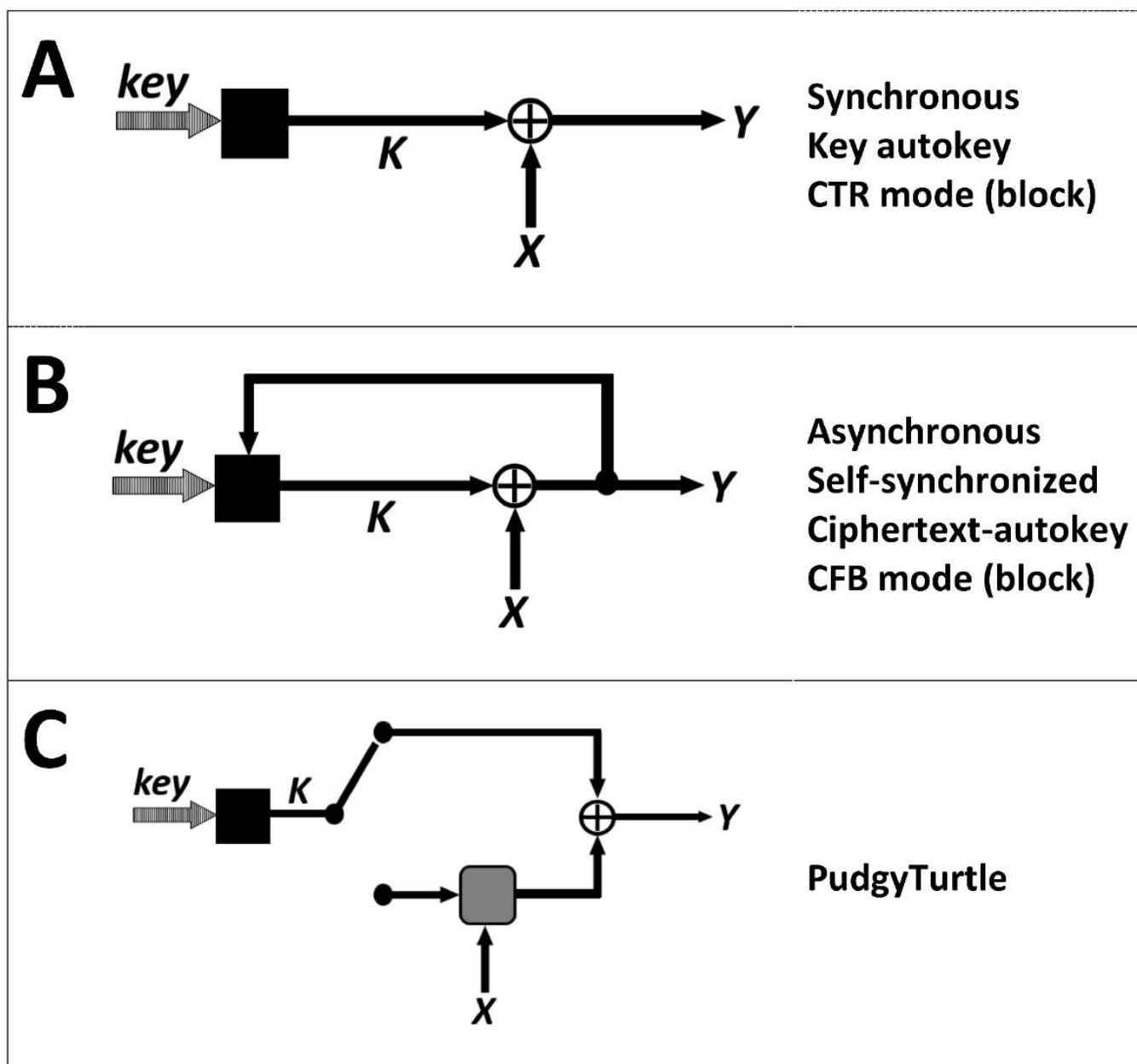
Modern TLS moved from CBC to GCM due to padding oracle attacks.

3 CFB (Cipher Feedback)



Cipher Feedback (CFB) Mode Encryption





Main Problems

- Bit flipping
- Ciphertext injection
- No integrity

Mitigations

1. Add Message Authentication Code

Use:

HMAC(key, ciphertext)

Verify before decryption.

Stops:

- Bit flipping
- Injection attacks

✓ 2. Random IV per Session

Prevents keystream predictability.

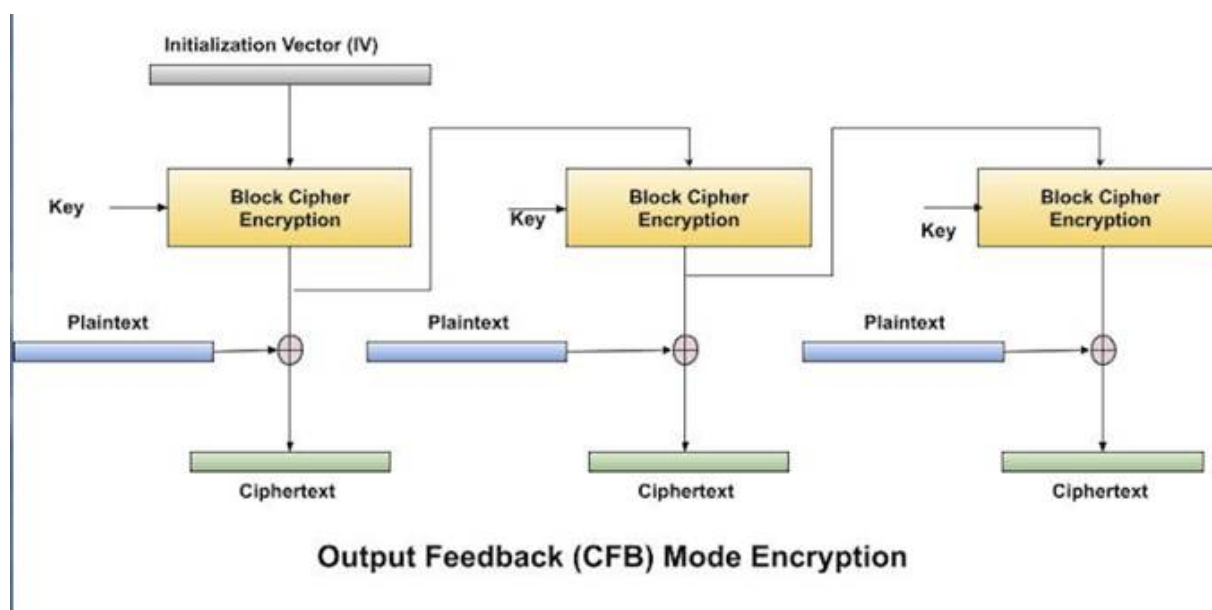
✓ 3. Use AEAD Modes

Replace CFB with:

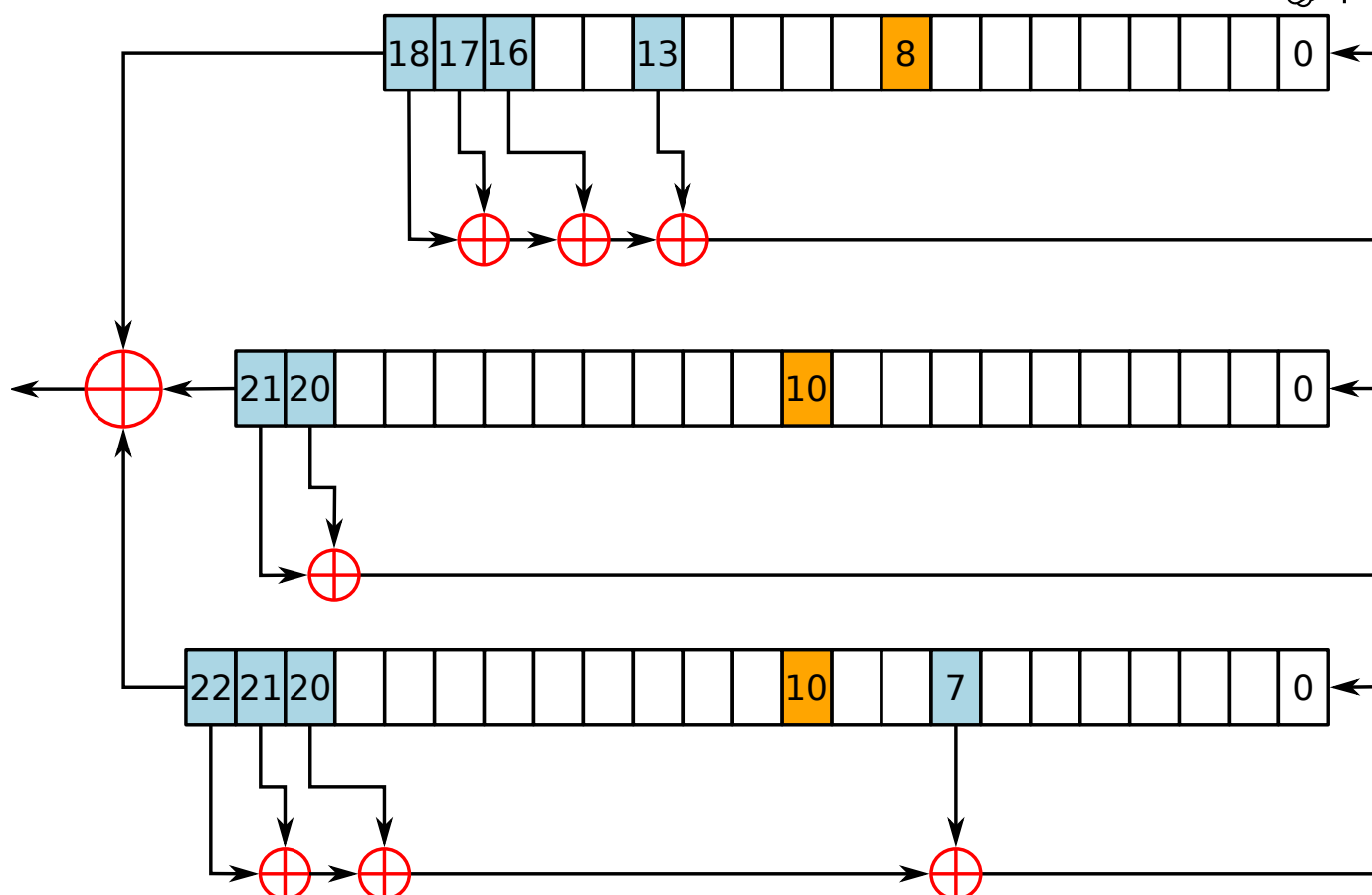
- AES-GCM
- AES-CCM

These ensure integrity + confidentiality.

4 OFB (Output Feedback)



<https://www.researchgate.net/publication/220798318/figure/fig1/AS%3A671533026512904%401537117358418/a-the-keystream-generation-mode-b-the-setup-mode-of-Grain-v1.png>



🔒 Main Problems

- Keystream reuse
- IV reuse
- Bit flipping
- No integrity

✅ Mitigations

✓ 1. Never Reuse IV with Same Key (CRITICAL)

Same Key + Same IV = Same Keystream

Must guarantee:

- Unique IV for every encryption

✓ 2. Use Secure Random IV

IV must be random and unpredictable.

✓ 3. Add Authentication (MAC)

Prevents bit-flipping attacks.

✓ 4. Use CTR or GCM Instead

OFB is outdated for modern systems.

5 CTR (Counter Mode)



<https://www.researchgate.net/publication/351510159/figure/fig2/AS%3A1022469469122565%401620787126781/Counter-CTR-mode-of-operation-for-block-ciphers-Figure-is-from-17.png>

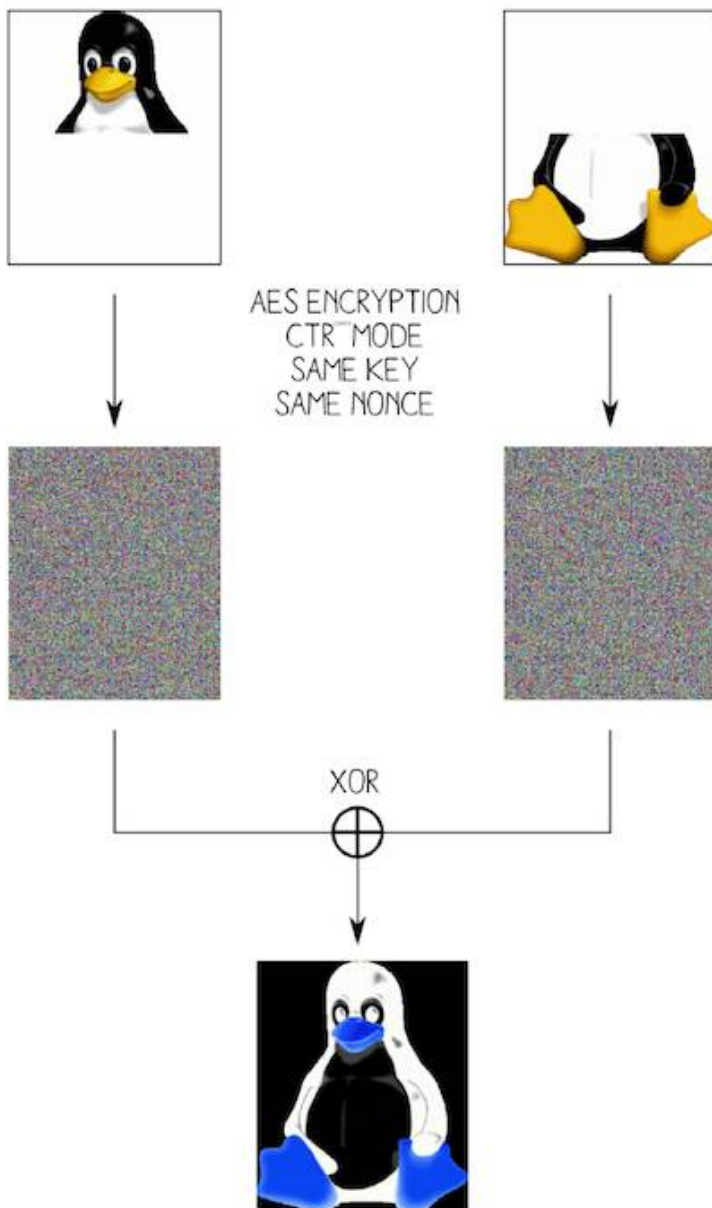


Ange

@angealbertini · Follow



Why using CTR mode with the same nonce is a bad idea



10:30 AM · Jan 21, 2014



173

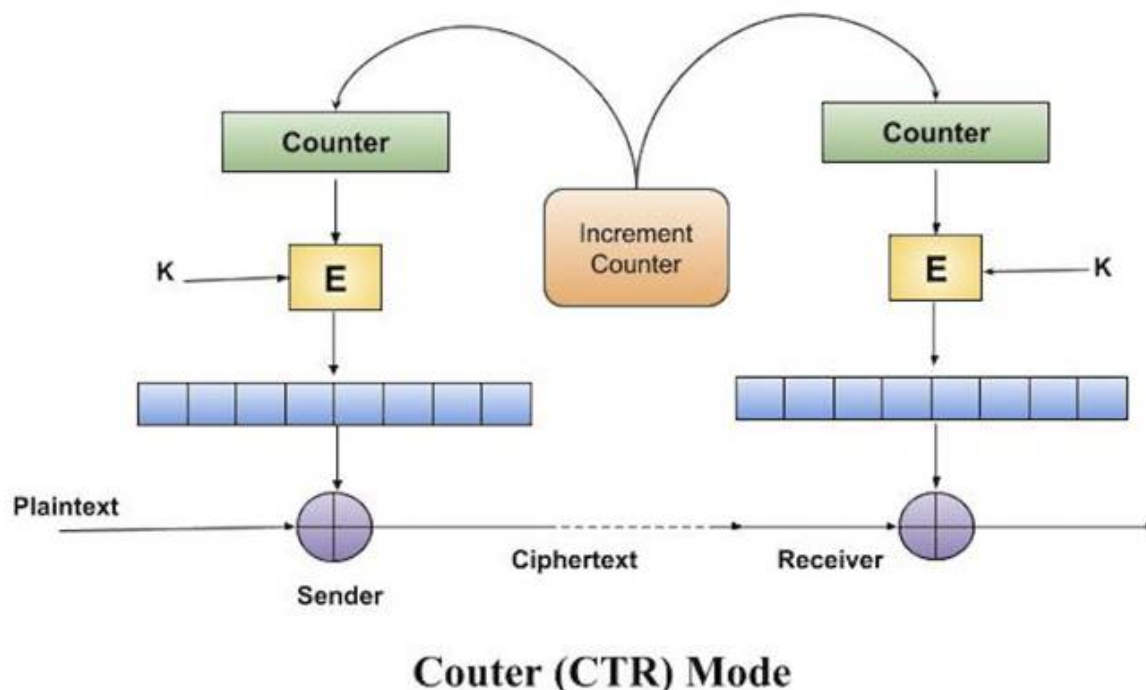


Reply



Copy link

[Read 11 replies](#)



🔒 Main Problems

- Nonce reuse (catastrophic)
- Bit-flipping
- No integrity

✅ Mitigations

✓ 1. Never Reuse Nonce with Same Key

Nonce must be:

- Unique per encryption
- Even predictable nonce is fine, but must be unique

Reuse = complete break.

✓ 2. Use Larger Nonce Space

Avoid counter overflow:

- Use 96-bit nonce (as in GCM)

✓ 3. Combine with MAC

Use Encrypt-then-MAC.

Prevents modification attacks.

✓ 4. Prefer AES-GCM

CTR + Authentication built-in.

● Final Comparison

Mode	Biggest Risk	Strongest Mitigation
ECB	Pattern leakage	Do not use ECB
CBC	Padding oracle	Encrypt-then-MAC
CFB	Bit flipping	Add MAC
OFB	IV reuse	Unique random IV
CTR	Nonce reuse	Unique nonce + MAC

🚨 Golden Rule (Exam Line)

Never rely on encryption alone.

Always use authenticated encryption (AEAD) like AES-GCM.

If you want, I can now convert this into:

- 📝 6-mark structured exam answer
- 🔥 Highly technical attack + mitigation comparison
- 📊 Short crisp revision table
- 🧠 Negative-marking trap version

Tell me your format preference.



An automatic teller machine (ATM) communicates with a central bank computer for PIN verification. A 32-bit CRC code is added to each packet to detect transmission errors and then the link is encrypted using a block cipher in counter mode. Describe an attack that is possible in this setup

take multiple different other scenarios of your choice, and describe cryptographic attacks that can be possible in ECB, CBC, CFB, OFB, CTR modes

✓ Part 1: ATM + CRC + CTR Mode — What Attack is Possible?

◆ Given System

- ATM sends PIN verification packets
- Adds **32-bit CRC (error detection only)**
- Then encrypts using **CTR mode**

🚨 Fundamental Problem

- **CRC is not cryptographically secure** (linear, predictable)
- **CTR mode provides confidentiality only**
- No message authentication (no MAC)

👉 This creates a **malleability vulnerability**.

🎯 Possible Attack: Bit-Flipping + CRC Recalculation Attack

CTR works as:

$$C = P \oplus K$$

If attacker flips ciphertext bit:

$$C' = C \oplus \Delta$$

Then decrypted:

$$P' = P \oplus \Delta$$

Meaning:

Attacker can modify plaintext predictably without knowing key.

🔒 Attack Scenario

Suppose packet contains:

Account_No | PIN | Amount | CRC

Attacker intercepts ciphertext.

Steps:

1. Flip bits in ciphertext to change:

- **Amount = 1000 → 9000**
- or change approval flag

2. Since CRC is linear:

- Attacker recomputes correct CRC difference
- Adjusts ciphertext accordingly

3. Bank decrypts:

- Sees modified plaintext
- CRC still valid
- No tampering detected

Why This Works

- CRC is linear:

$$CRC(P \oplus \Delta) = CRC(P) \oplus CRC(\Delta)$$

- CTR is XOR-based
- Linearity + Linearity = Exploitable

Result

Attacker can:

- Modify transactions
- Change amounts
- Manipulate flags
- Possibly bypass PIN logic

Root Cause

Error detection \neq Message authentication

Encryption \neq Integrity

Correct Fix

Use **Authenticated Encryption (AEAD)** like:

- AES-GCM
- Encrypt-then-MAC

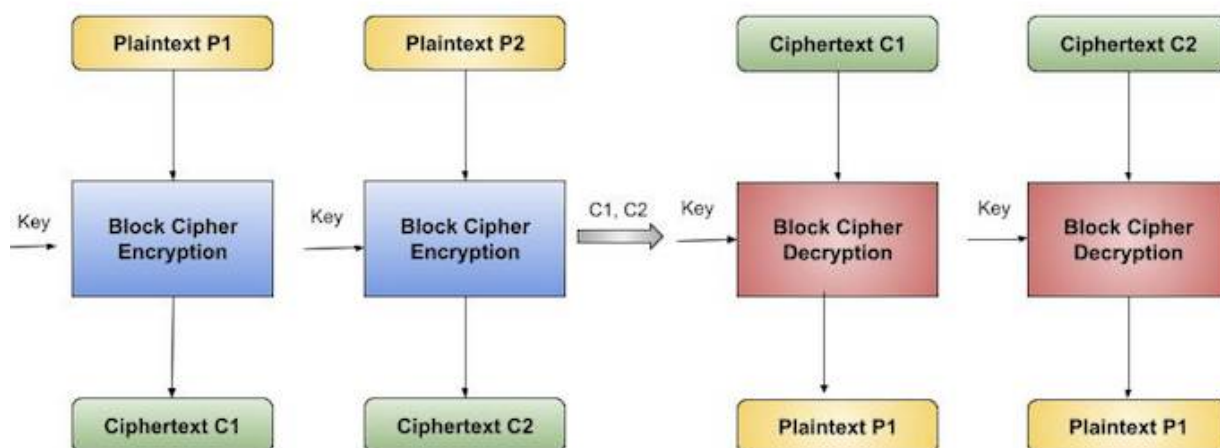
CRC must NOT be used for security.

Part 2: Multiple Different Scenarios for Each Mode + Attacks

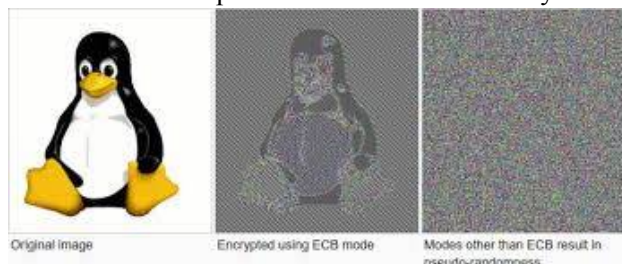
1 ECB Mode

◆ Scenario: Encrypted Medical Records Database

Each patient record encrypted block-by-block.



<https://www.researchgate.net/publication/339988387/figure/fig4/AS%3A870178569072646%401584478146614/ECB-Mode-a-Identical-plaintext-under-the-same-key-results-in-identical-ciphertext-b.png>



🔒 Attacks in ECB

1) Pattern Leakage Attack

Identical diagnosis codes → identical ciphertext blocks
Attacker infers disease patterns.

2) Record Splicing Attack

Attacker swaps ciphertext blocks between patients.
System decrypts successfully.

3) Database Watermarking Attack

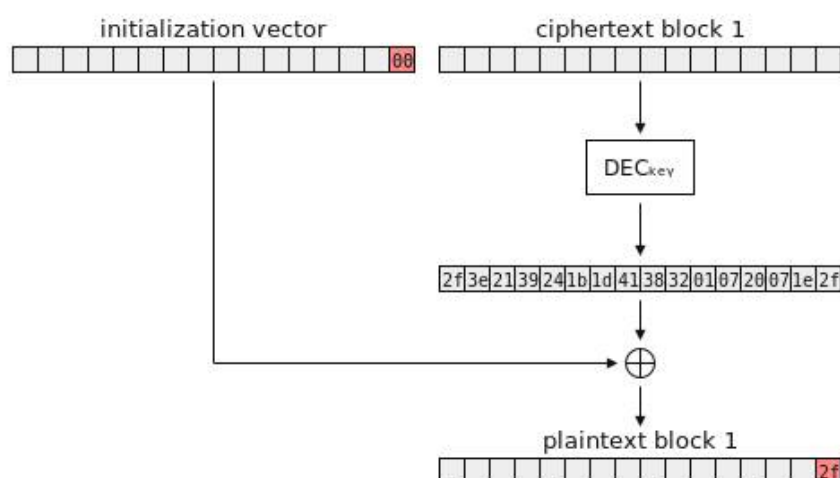
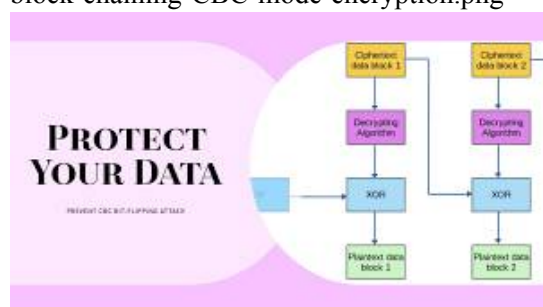
Attacker inserts known plaintext
 Observes matching ciphertext blocks later.

2 CBC Mode

◆ Scenario: Secure Online Loan Approval Forms



<https://www.researchgate.net/publication/215783767/figure/fig1/AS%3A394138559238144%401470981363092/Cipher-block-chaining-CBC-mode-encryption.png>



Attacks in CBC

1) Bit-Flipping Attack

Modify ciphertext block C_i
 Controls plaintext of block P_{i+1}

Attack example:

`LoanAmount = 10000 → 90000`

2) Padding Oracle Attack

If server reveals padding errors:
 Attacker decrypts full ciphertext block-by-block.
 Used historically in TLS attacks.

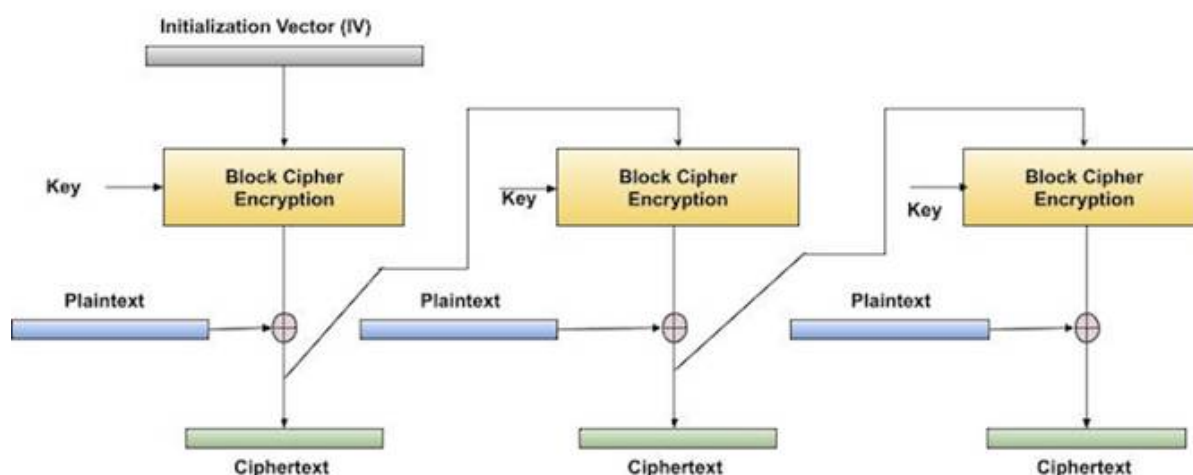
3) IV Tampering Attack

If IV predictable:

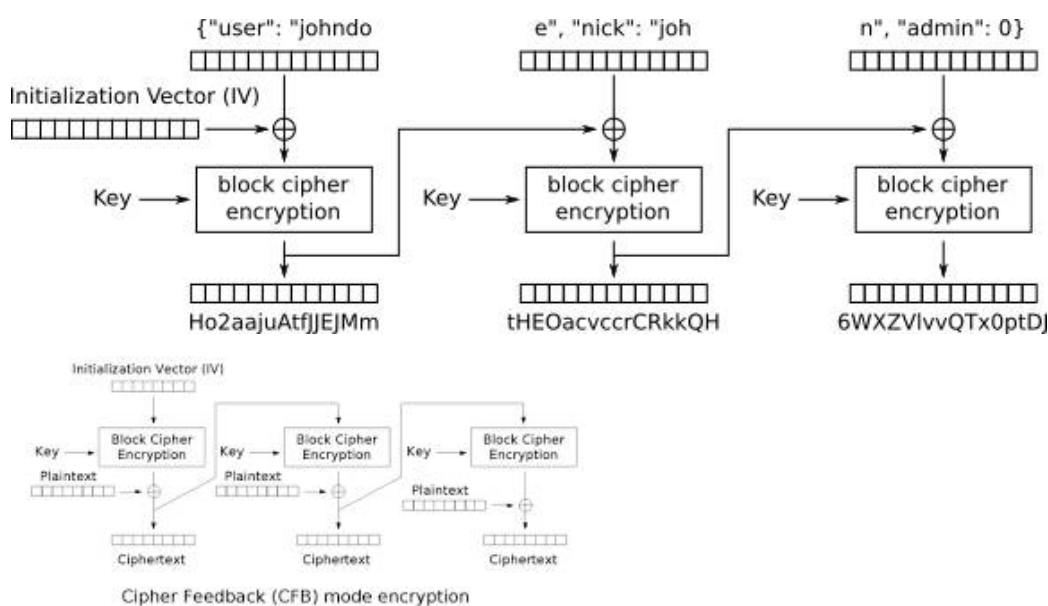
Attacker modifies first plaintext block.

3 CFB Mode

◆ Scenario: Encrypted Industrial Control Commands



Cipher Feedback (CFB) Mode Encryption



🔒 Attacks in CFB

1) Controlled Bit-Flipping

Flip one ciphertext bit → flips one plaintext bit

Change:

Valve=Open → Valve=Close

2) Injection Attack

If no authentication:

Attacker injects valid-looking ciphertext stream.

3) Replay Attack

If IV reused:

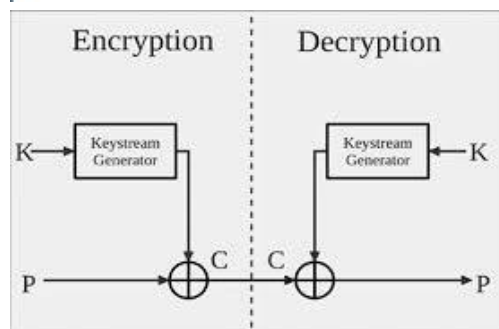
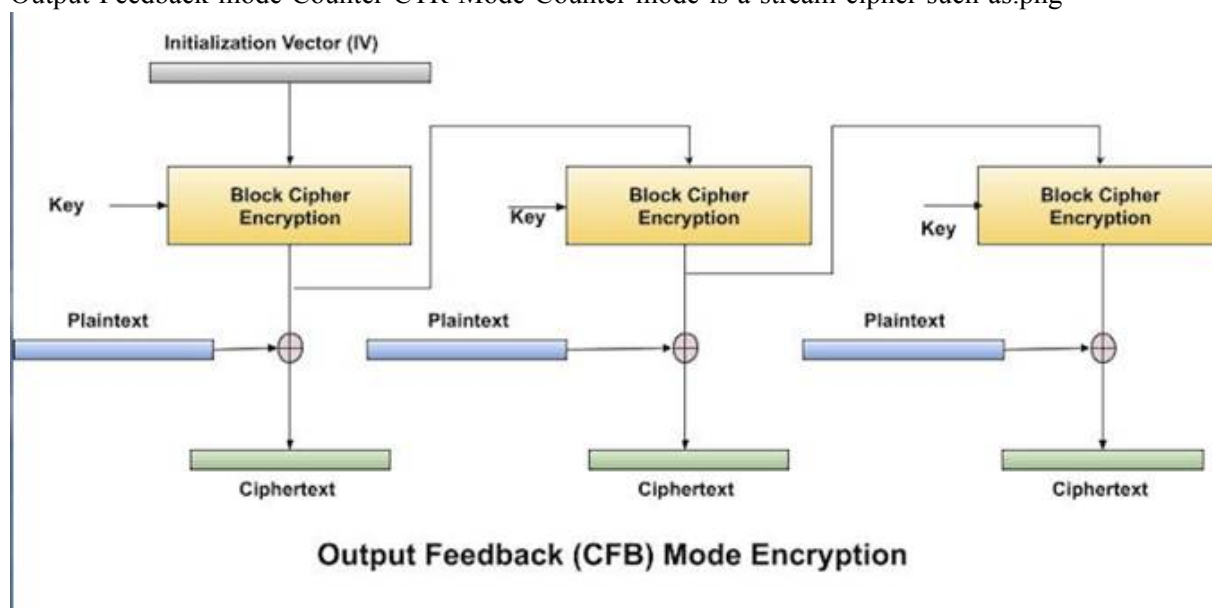
Old commands replayed.

4 OFB Mode

◆ Scenario: Military Field Radio Transmission



https://www.researchgate.net/profile/Drdinesh_Goyal/publication/273260684/figure/fig4/AS%3A391769960271878%4014%20Output-Feedback-mode-Counter-CTR-Mode-Counter-mode-is-a-stream-cipher-such-as.png



🔑 Attacks in OFB

1) Keystream Reuse Attack

Same Key + Same IV → Same Keystream

Attacker computes:

$$C_1 \oplus C_2 = P_1 \oplus P_2$$

Complete message recovery possible.

2) Bit-Flipping Attack

Direct plaintext manipulation.

5 CTR Mode

◆ Scenario: Cloud Encrypted Document Storage



<https://www.researchgate.net/publication/260540463/figure/fig3/AS%3A392489967079430%401470588308704/Diagram-of-the-CTR-encryption-decryption-mode.png>

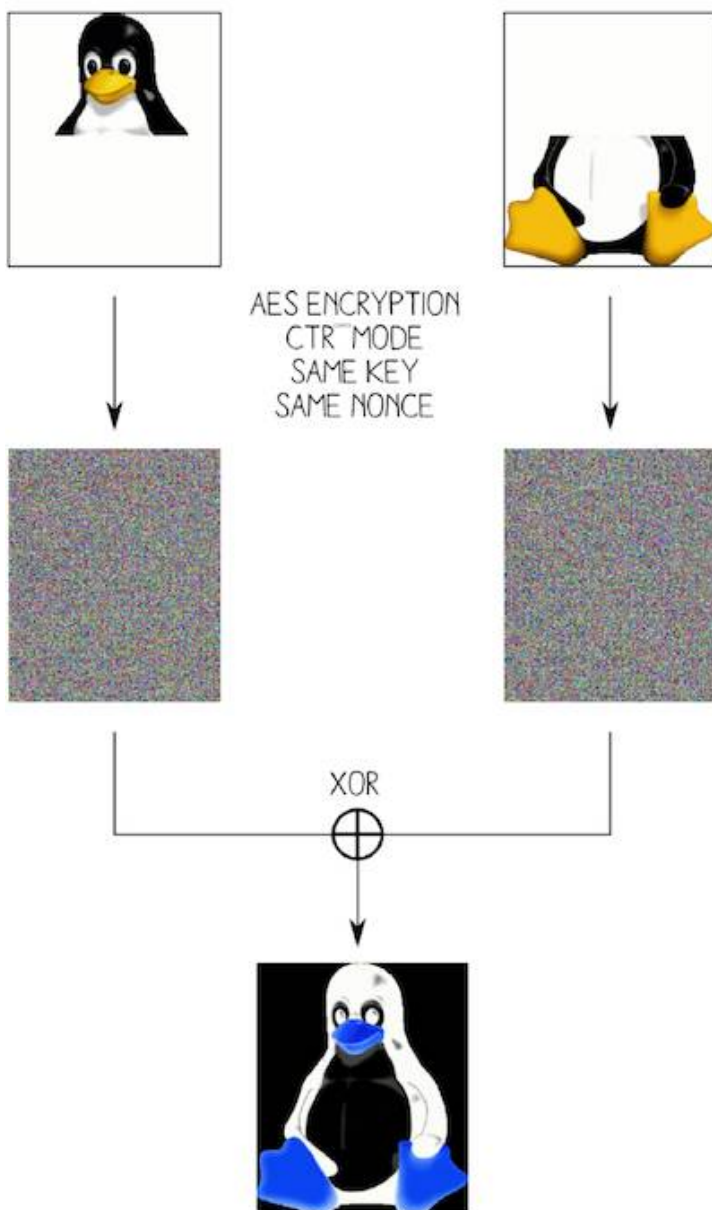


Ange

@angealbertini · Follow



Why using CTR mode with the same nonce is a bad idea



10:30 AM · Jan 21, 2014



173

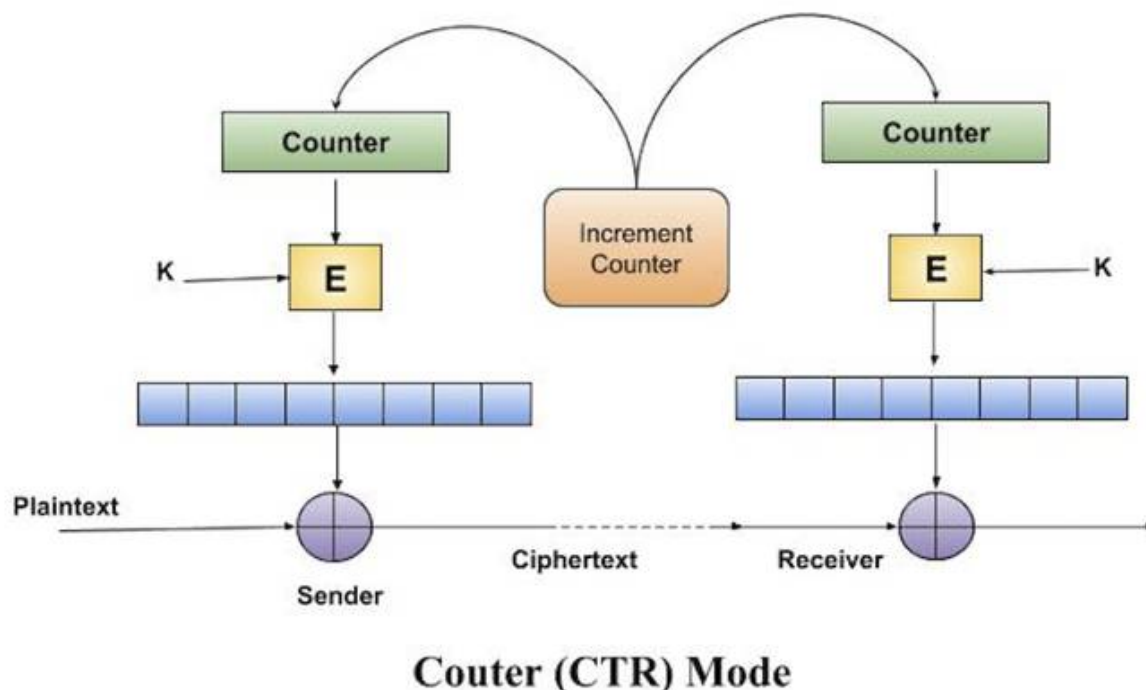


Reply



Copy link

[Read 11 replies](#)



🔒 Attacks in CTR

1) Nonce Reuse Attack (Catastrophic)

If same nonce reused:

Attacker gets:

$$C_1 \oplus C_2 = P_1 \oplus P_2$$

Full plaintext recovery possible.

2) Precise Plaintext Manipulation

Attacker flips ciphertext bit

Predictable plaintext change

3) Known Plaintext Amplification

If one plaintext known:

Keystream recovered

Other messages decrypted.

● Core Insight Across All Modes

Mode	Confidentiality	Integrity	Main Weakness
ECB	❌ Pattern leak	❌ None	Block independence

Mode	Confidentiality	Integrity	Main Weakness
CBC	✓	✗ None	Padding oracle
CFB	✓	✗ None	Malleability
OFB	✓	✗ None	Keystream reuse
CTR	✓	✗ None	Nonce reuse

🔥 Final Exam-Ready One Line

All basic block cipher modes provide confidentiality only; without authentication, they are vulnerable to active attacks such as bit-flipping, replay, and chosen ciphertext attacks.

If you want, I can now give:

- 🧠 10-mark structured answer format
- 100 Ultra-confusing negative-marking traps
- 📖 2-page revision sheet
- 🧪 Mathematical explanation version