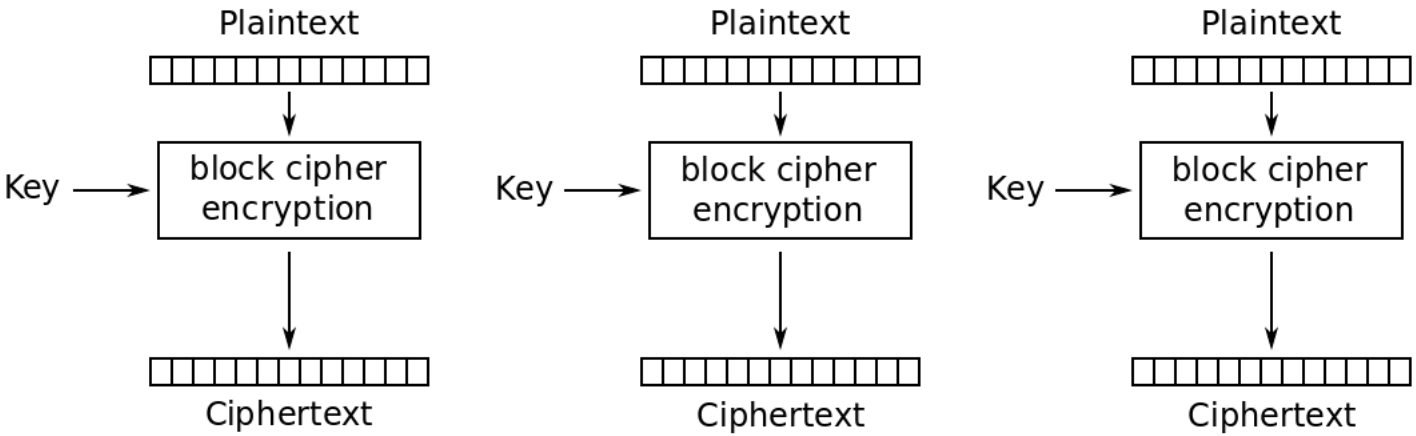
Block Cipher Modes

Multiple Blocks

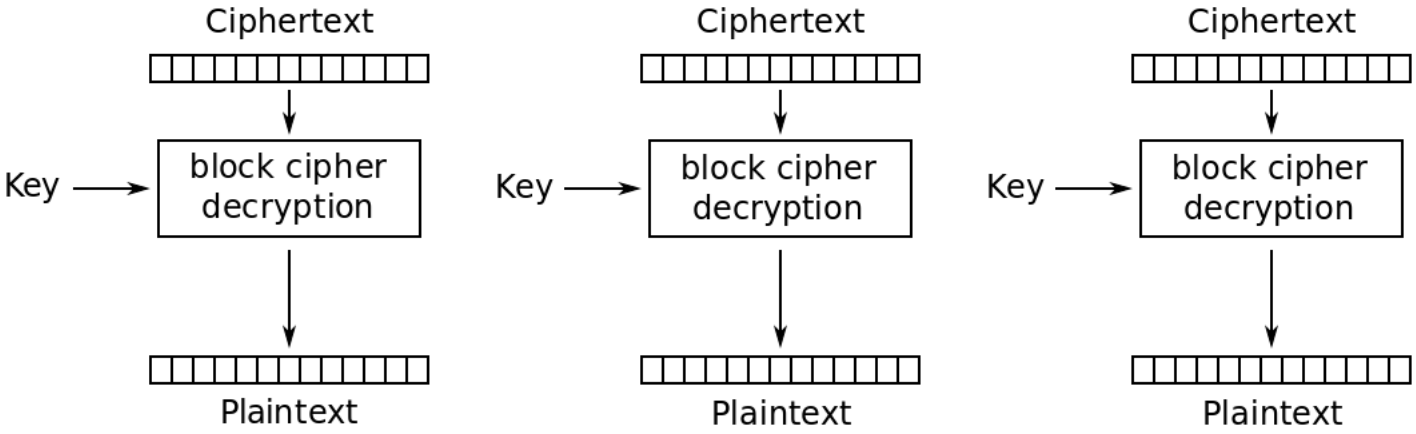
- How to encrypt multiple blocks?
- Do we need a new key for each block?
  - As bad as (or worse than) a one-time pad!
- Encrypt each block independently?
- Make encryption depend on previous block?
  - That is, can we “chain” the blocks together?

Modes of Operation

- Electronic Codebook (**ECB**) mode
  - Encrypt each block independently
  - Most obvious, but has a serious weakness
- Cipher Block Chaining (**CBC**) mode
  - Chain the blocks together
  - More secure than ECB, virtually no extra work
- Counter Mode (**CTR**) mode
  - Block ciphers acts like a stream cipher
  - Popular for random access
- Cipher Feedback (**CFB**) mode
- Output Feedback (**OFB**) mode



Electronic Codebook (ECB) mode encryption



Electronic Codebook (ECB) mode decryption

ECB Mode

- Notation: $C = E(P,K)$
- Given plaintext $P_0, P_1, \ldots, P_m, \ldots$
- Most obvious way to use a block cipher:

Encrypt	Decrypt
$C_0 = E(P_0, K)$	$P_0 = D(C_0, K)$
$C_1 = E(P_1, K)$	$P_1 = D(C_1, K)$
$C_2 = E(P_2, K) \ \ldots$	$P_2 = D(C_2, K) \ \ldots$

- For fixed key K, this is "electronic" version of a codebook cipher (without additive)
  - With a different codebook for each key

ECB Weakness

- Suppose $P_i = P_j$
- Then $C_i = C_j$ and Trudy knows $P_i = P_j$
- This gives Trudy some information, even if she does not know P_i or P_j
- Trudy might know P_i
- Is this a serious issue?

ECB Cut and Paste Attack

- Suppose plaintext is

  `Alice digs Bob. Trudy digs Tom.`
- Assuming 64-bit blocks and 8-bit ASCII:

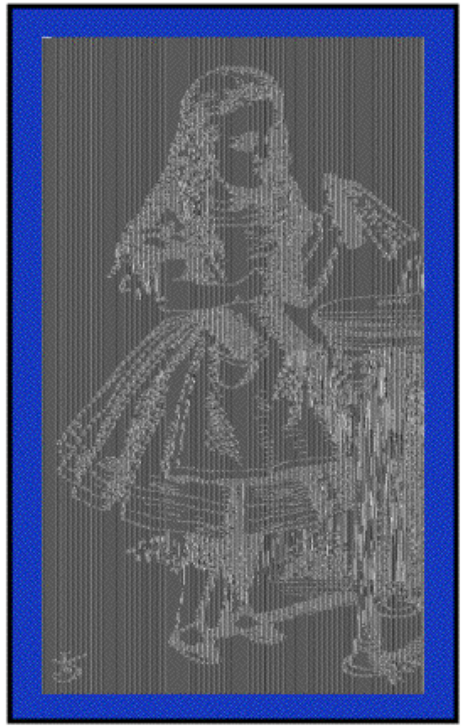
  $P_0 = \text{“}$`Alice di`$\text{”}, P_1 = \text{“}$`gs Bob. `$\text{”},$

  $P_2 = \text{“}$`Trudy di`$\text{”}, P_3 = \text{“}$`gs Tom. `”
- Ciphertext: C_0, C_1, C_2, C_3
- Trudy cuts and pastes: C_0, C_3, C_2, C_1
- Decrypts as

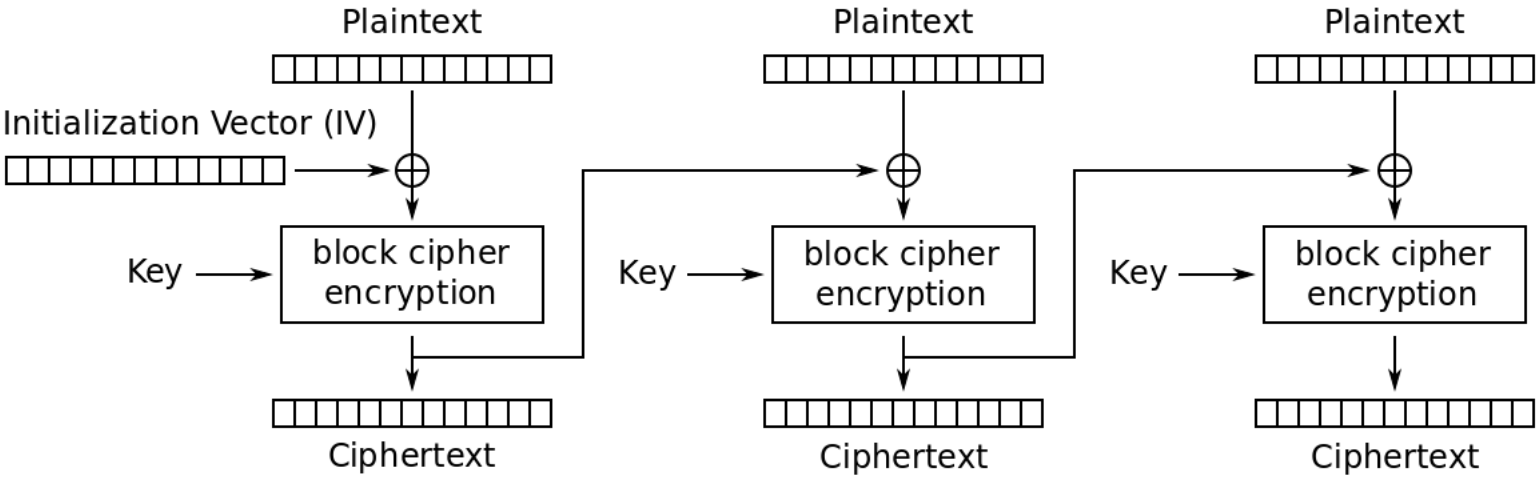
  `Alice digs Tom. Trudy digs Bob.`

Alice Hates ECB Mode

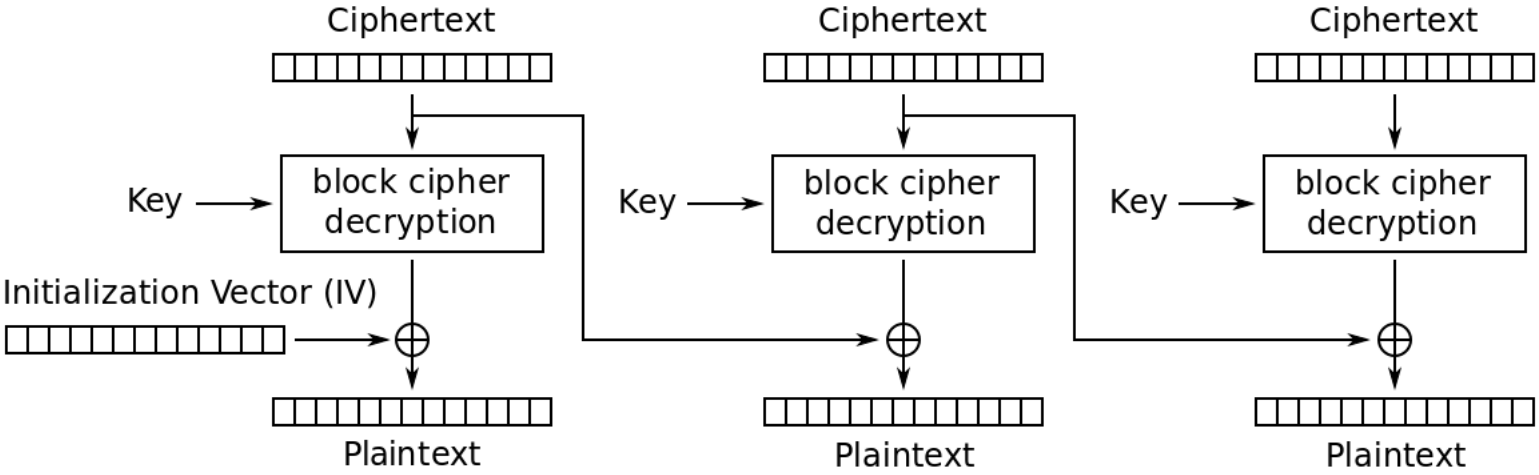
- Alice's uncompressed image, and ECB encrypted



- Why does this happen?
- Same plaintext yields same ciphertext!



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

CBC Mode

- Blocks are "chained" together
- A random initialization vector, or IV, is required to initialize CBC mode
- IV is random, but not secret

Encryption

$$C_0 = E(IV \oplus P_0, K),$$
$$C_1 = E(C_0 \oplus P_1, K),$$
$$C_2 = E(C_1 \oplus P_2, K), \ldots$$

Decryption

$$P_0 = IV \oplus D(C_0, K),$$
$$P_1 = C_0 \oplus D(C_1, K),$$
$$P_2 = C_1 \oplus D(C_2, K), \ldots$$

- Analogous to classic codebook *with additive*

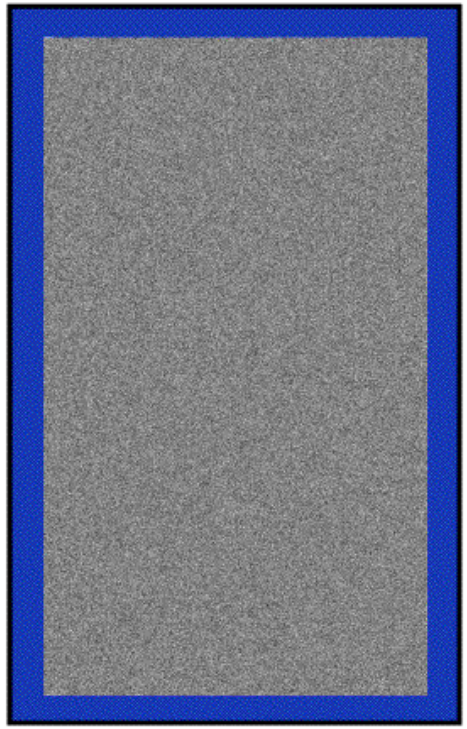
CBC Mode

- Identical plaintext blocks yield different ciphertext blocks — this is good!
- If C_1 is garbled to, say, G then

  $P_1 \neq C_0 \oplus D(G, K), P_2 \neq G \oplus D(C_2, K)$
- But $P_3 = C_2 \oplus D(C_3, K), P_4 = C_3 \oplus D(C_4, K), \ldots$
- Automatically recovers from errors!
- Cut and paste is still possible, but more complex (and will cause garbles)

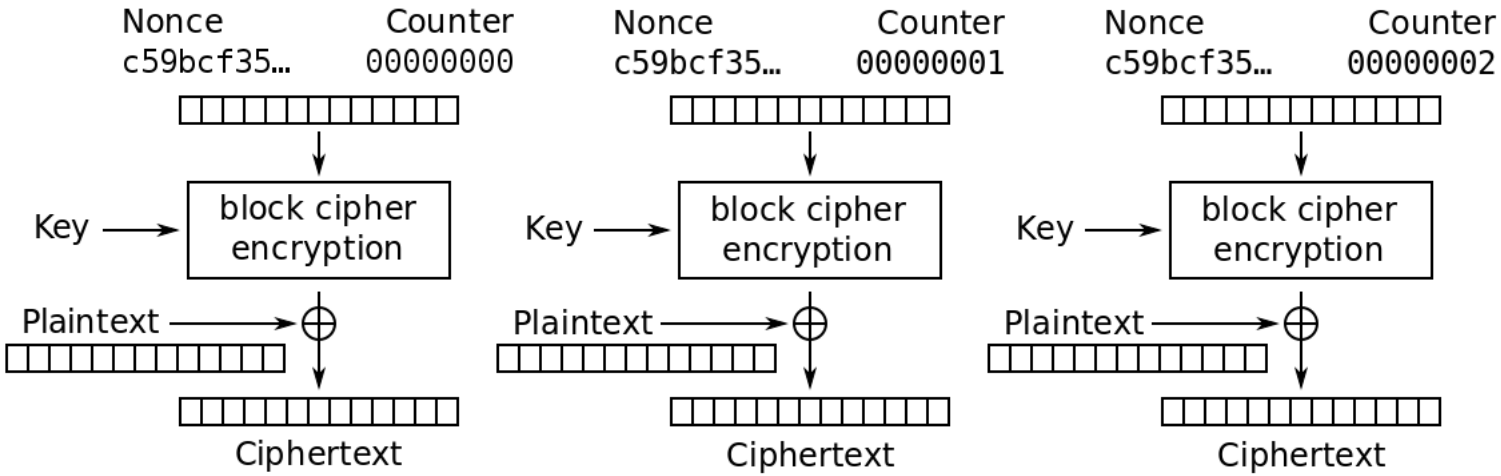
Alice Likes CBC Mode

- Alice's uncompressed image, Alice CBC encrypted (TEA)

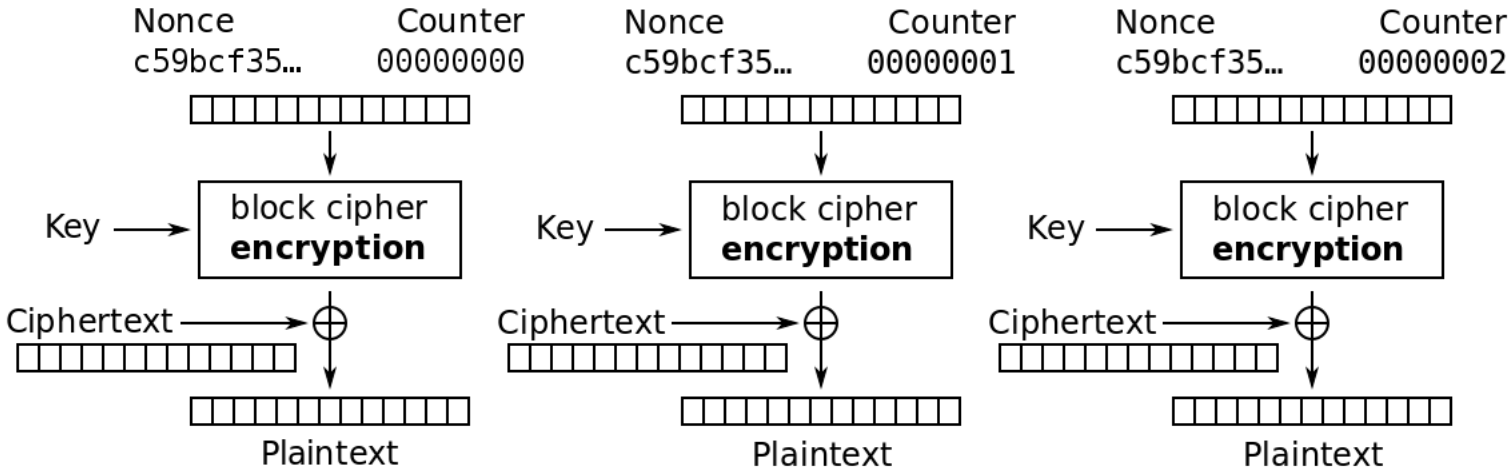


Attack on CBC

- Modifying Cipher Blocks:
  - You can modify the contents of one cipher block (c6) to make the plain text (m7) as you wish, however the preceding block (m6) will be garbled



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Counter Mode (CTR)

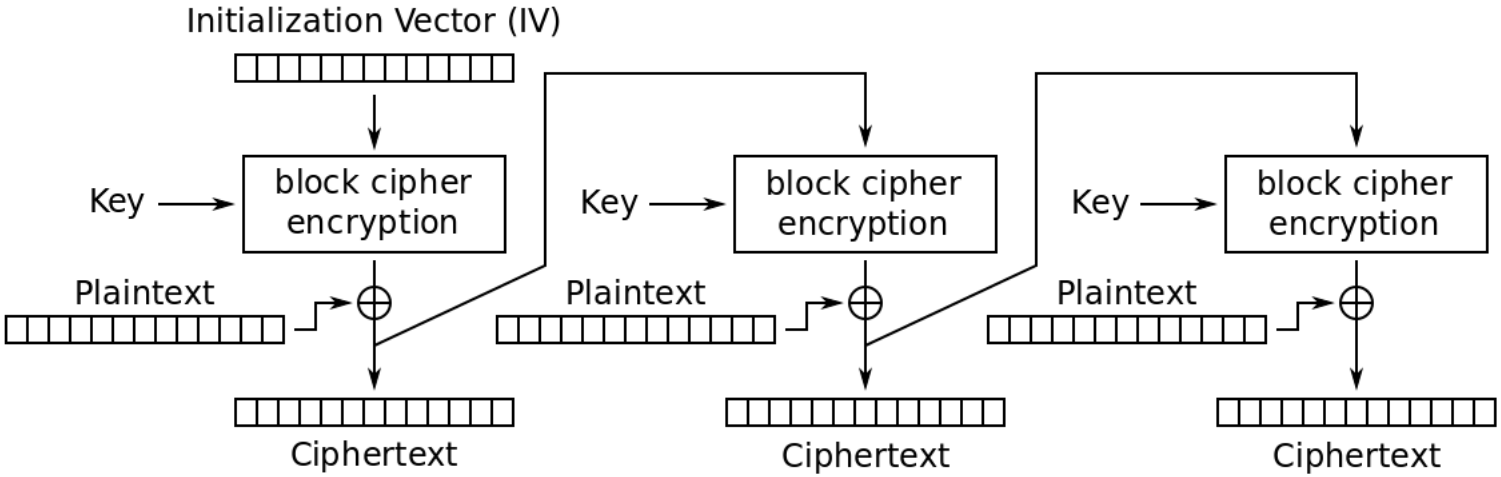
- CTR is popular for random access
- Use block cipher like a stream cipher

Encryption

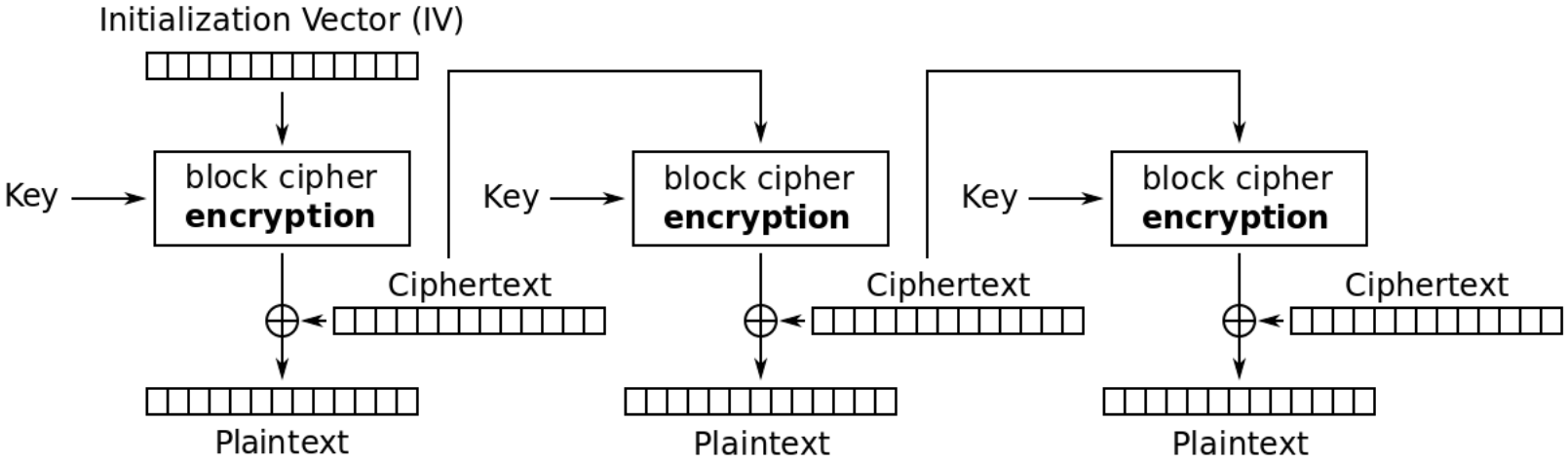
$$C_0 = P_0 \oplus E(IV, K),$$
$$C_1 = P_1 \oplus E(IV+1, K),$$
$$C_2 = P_2 \oplus E(IV+2, K),\ldots$$

Decryption

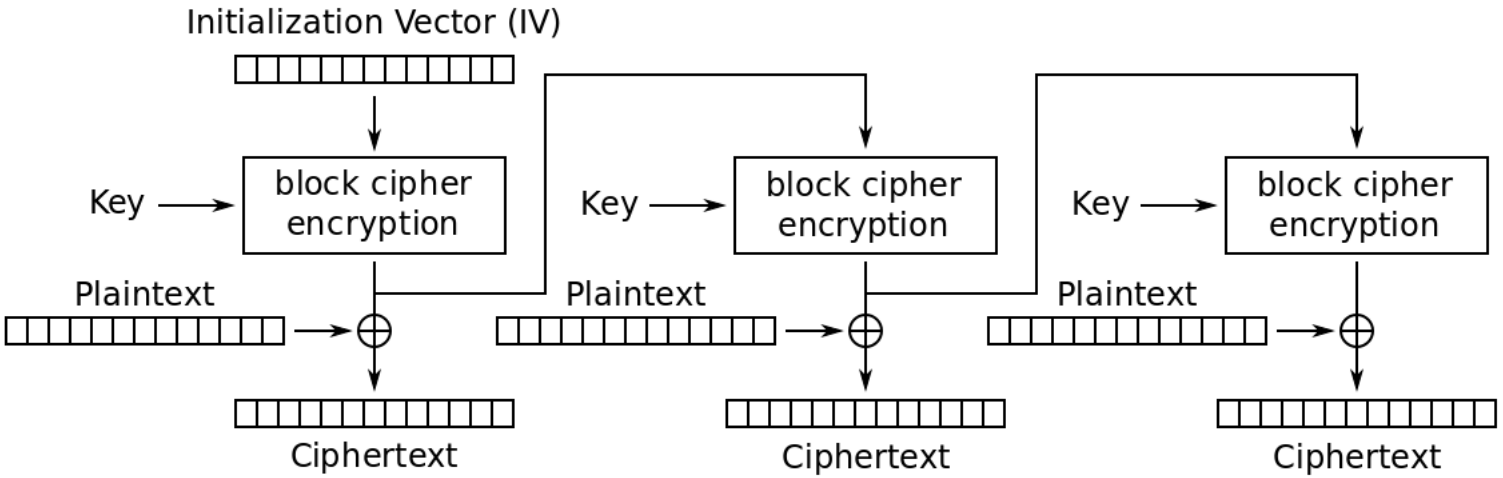
$$P_0 = C_0 \oplus E(IV, K),$$
$$P_1 = C_1 \oplus E(IV+1, K),$$
$$P_2 = C_2 \oplus E(IV+2, K),\ldots$$



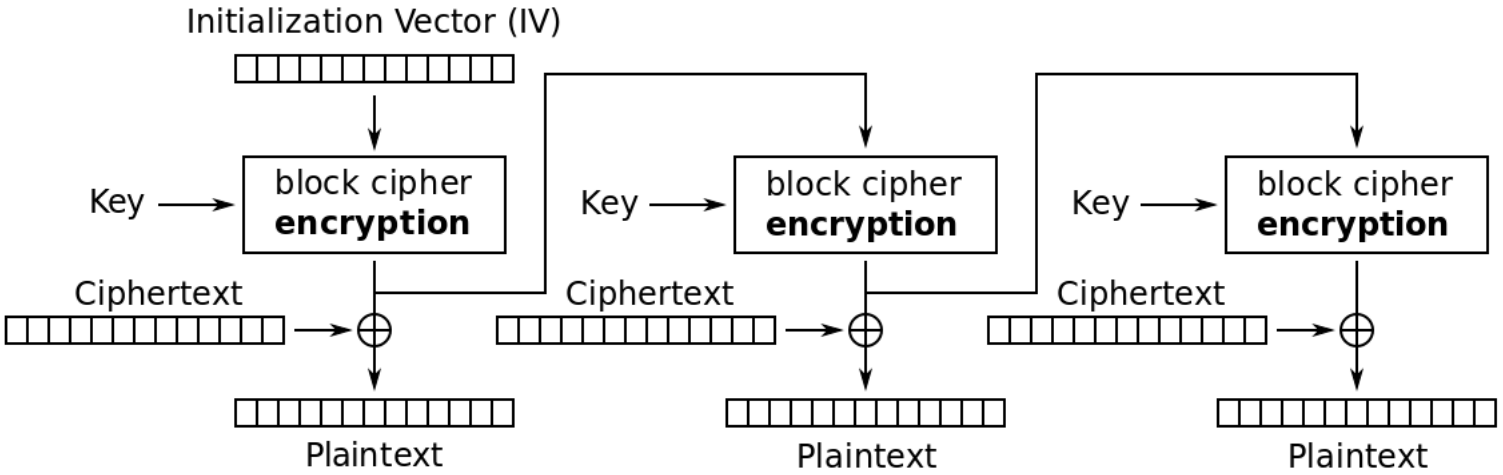
Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption



Output Feedback (OFB) mode encryption



Output Feedback (OFB) mode decryption