CS 465

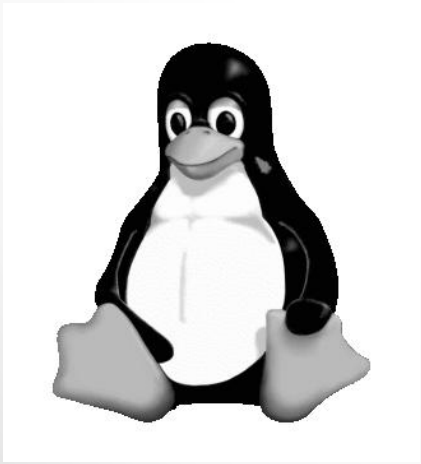
Block Cipher Modes

Last Updated: Aug 26, 2013

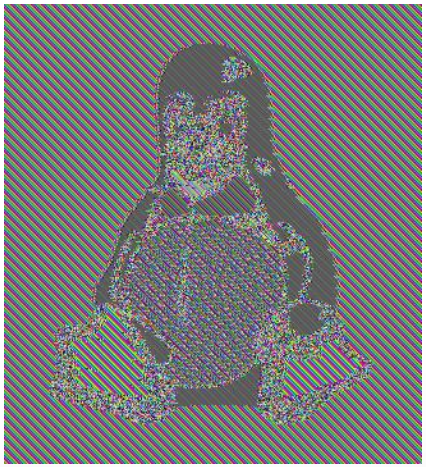
ECB Mode

- Electronic Code Book
- Divide the plaintext into fixed-size blocks
- Encrypt/Decrypt each block independently
- There is a weakness with this approach

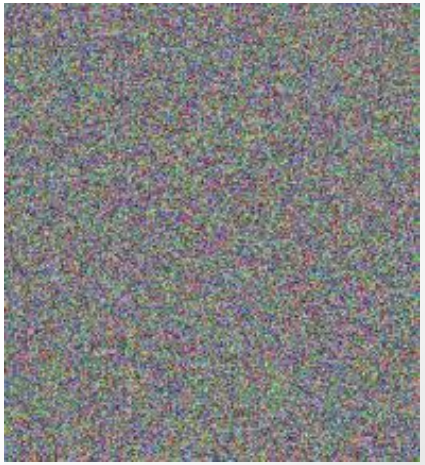
“Plain-Tux”

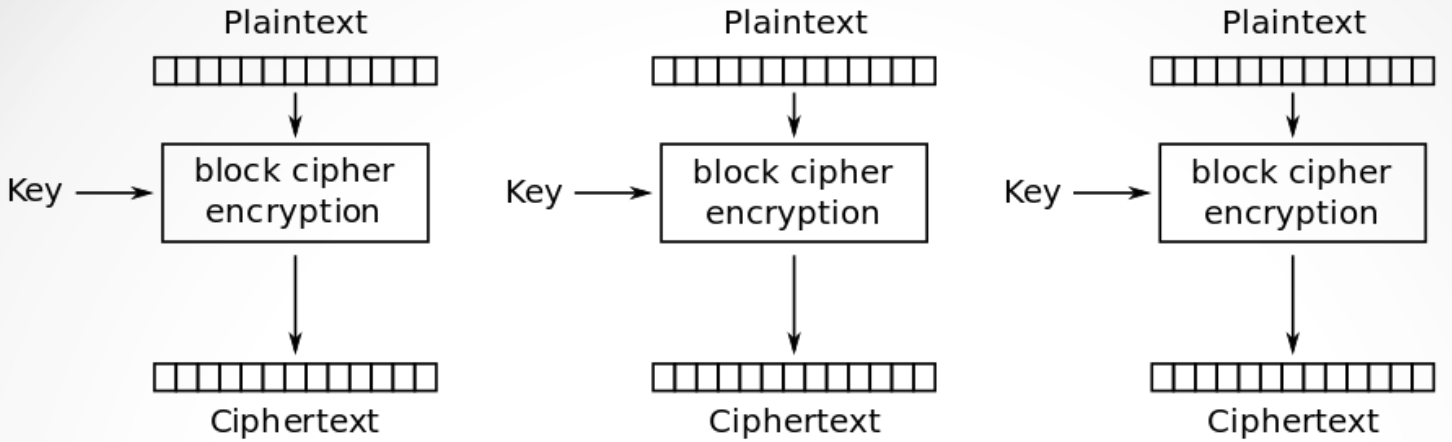


“Cipher-Tux”

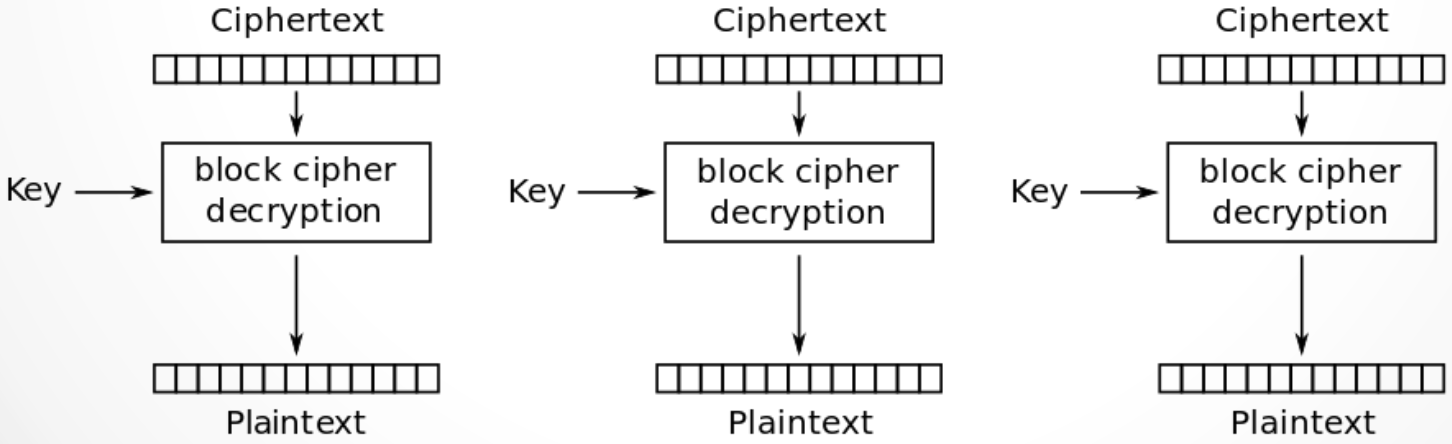


“Cipher-Tux2”





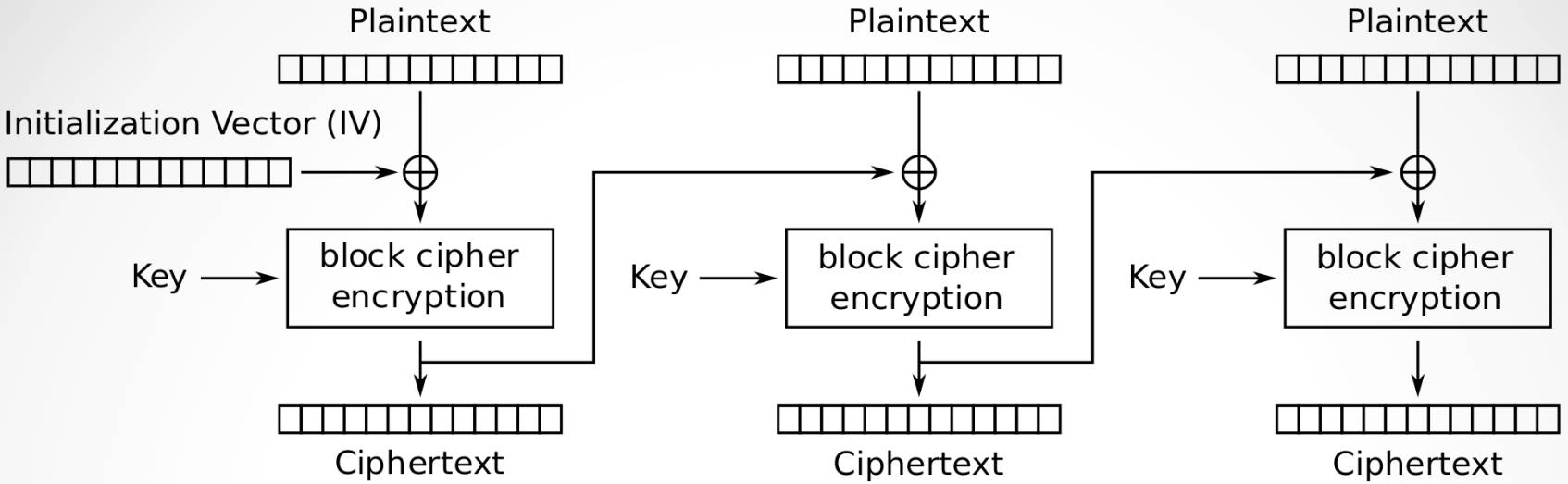
Electronic Codebook (ECB) mode encryption



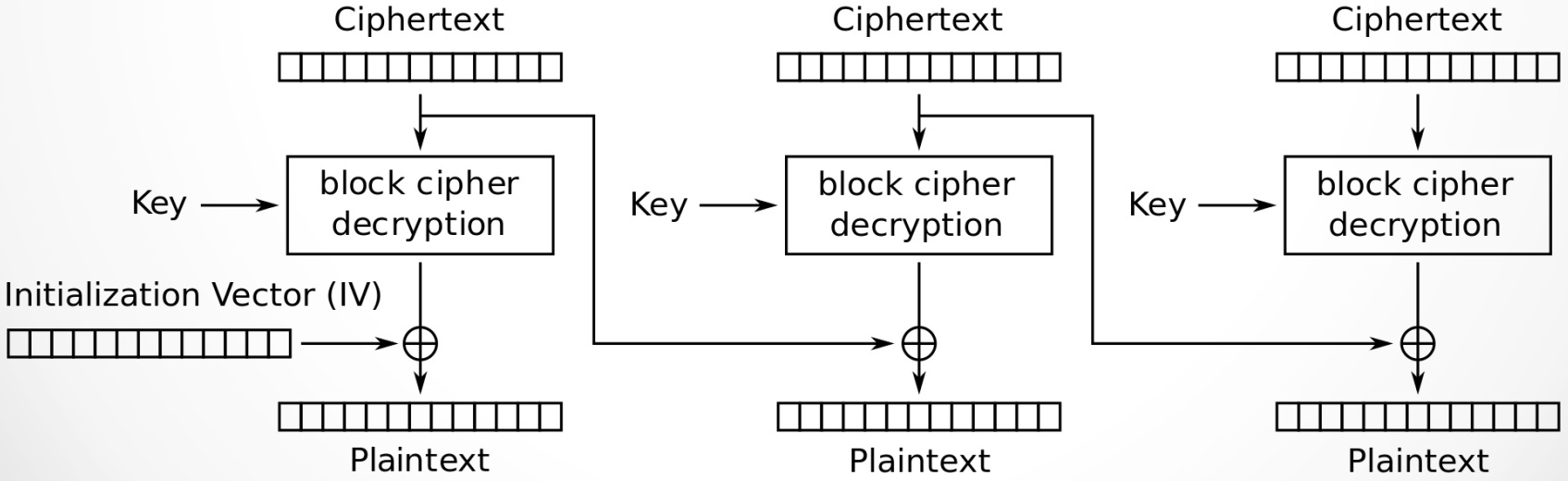
Electronic Codebook (ECB) mode decryption

CBC Mode

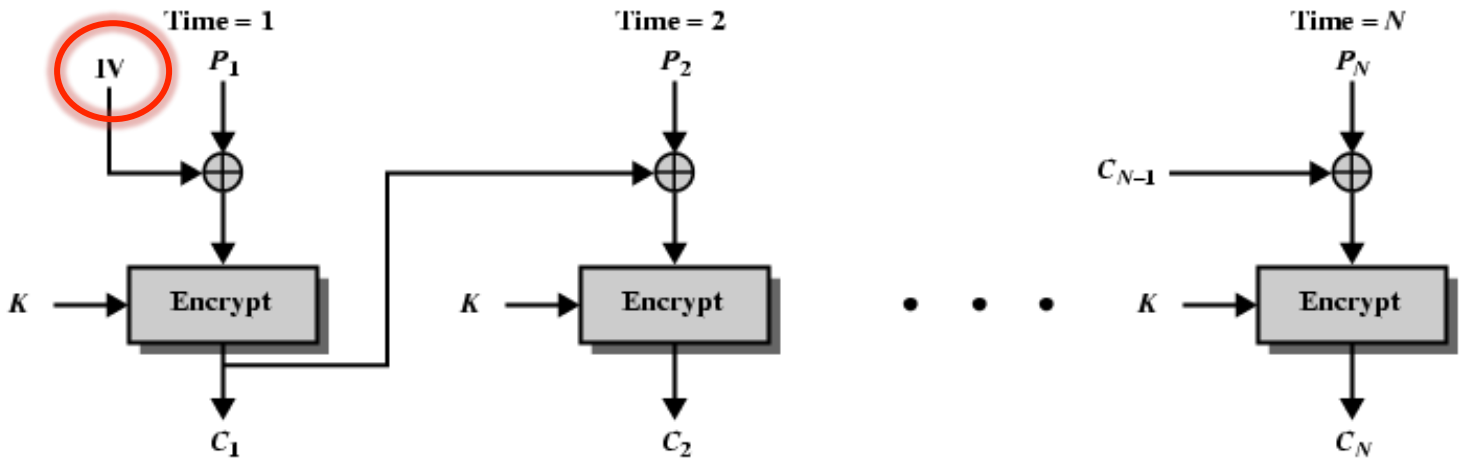
- Cipher Block Chaining
- Overcomes the problem with ECB
- XOR the plaintext with the prior ciphertext block
- What about the first block?



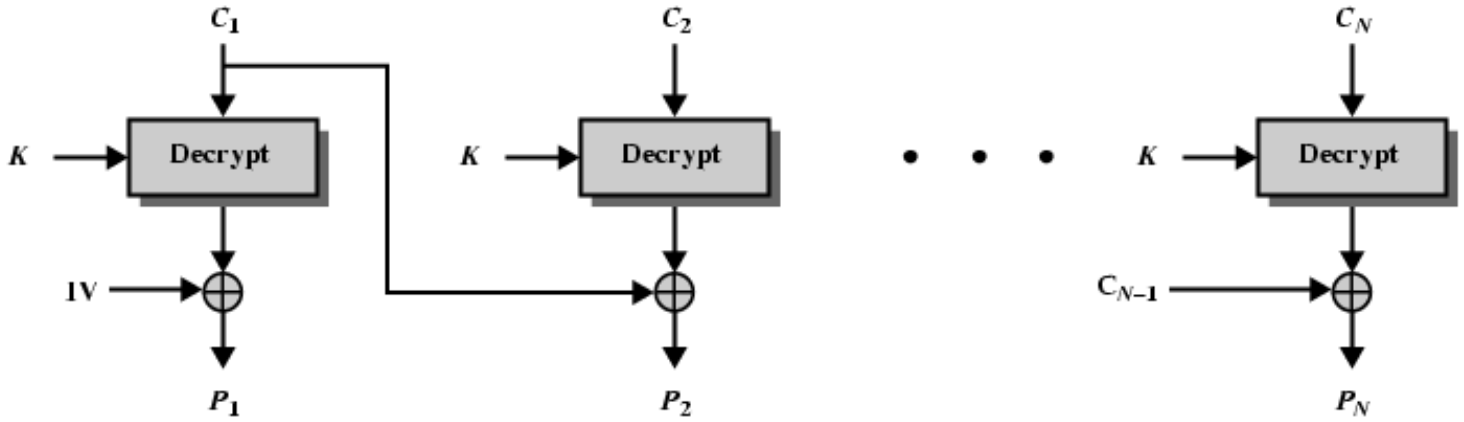
Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption



(a) Encryption



(b) Decryption

Figure 2.7 Cipher Block Chaining (CBC) Mode

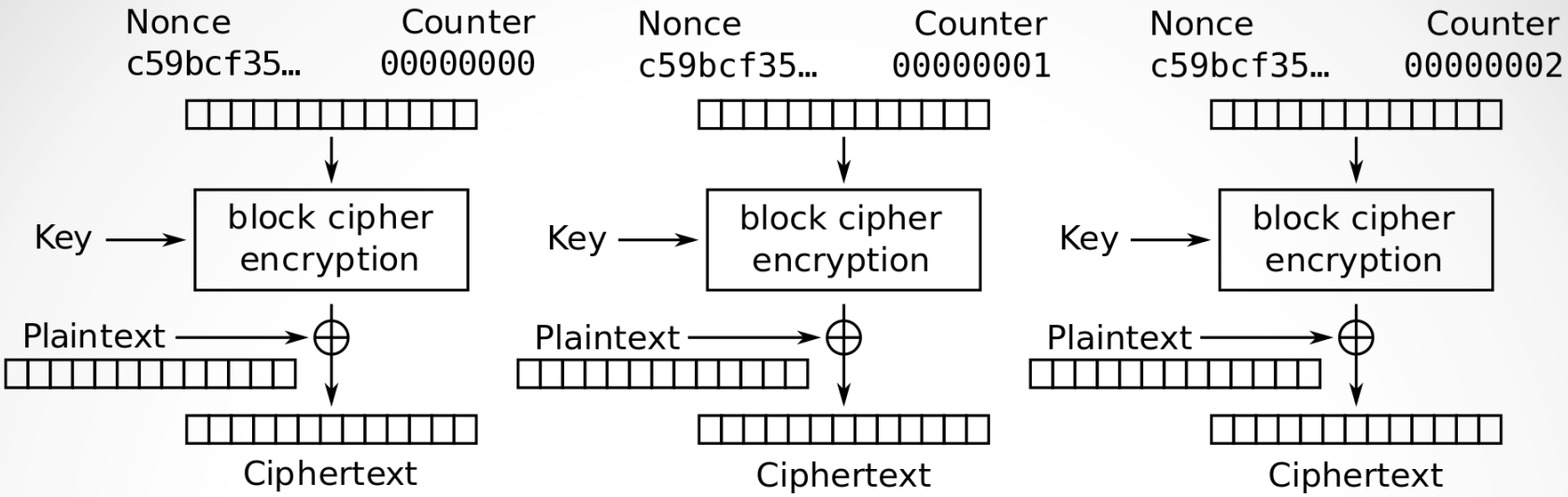
Source: Network Security Essentials (Stallings)

Initialization Vector (IV)

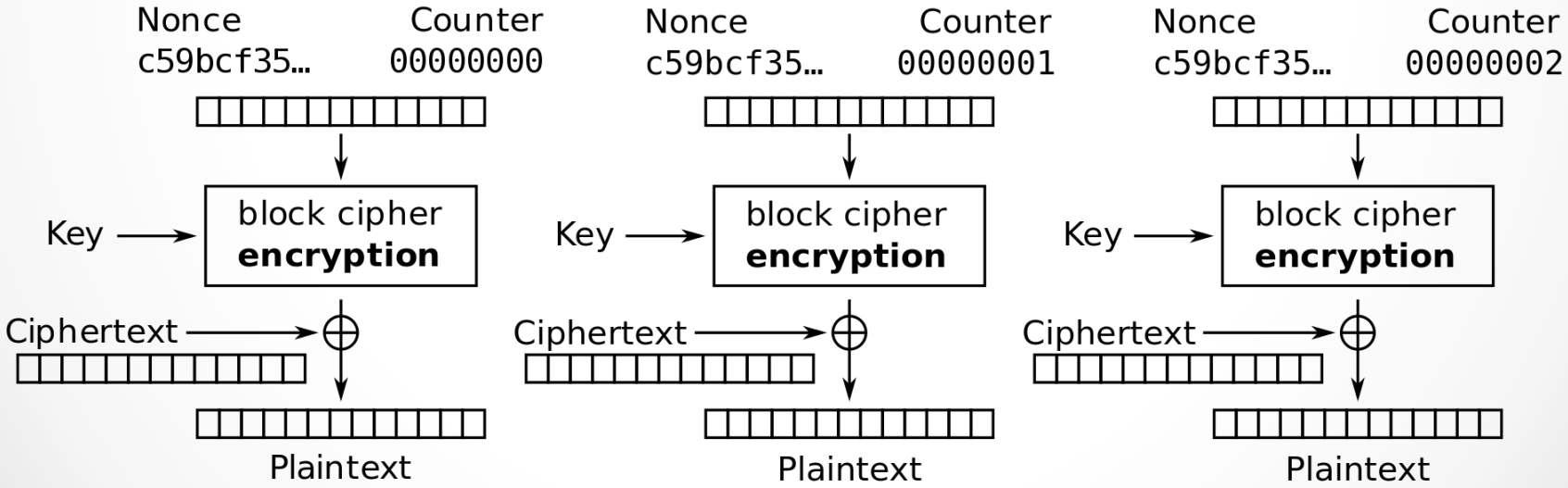
- Must be known to both the sender and recipient
- Ideally both IV and key should protected, but the IV may be public
- Common approach: encrypt IV using ECB and send it with the encrypted data
- Most importantly, an IV should never be reused with the same key. Why?

Block Cipher as a Stream Cipher

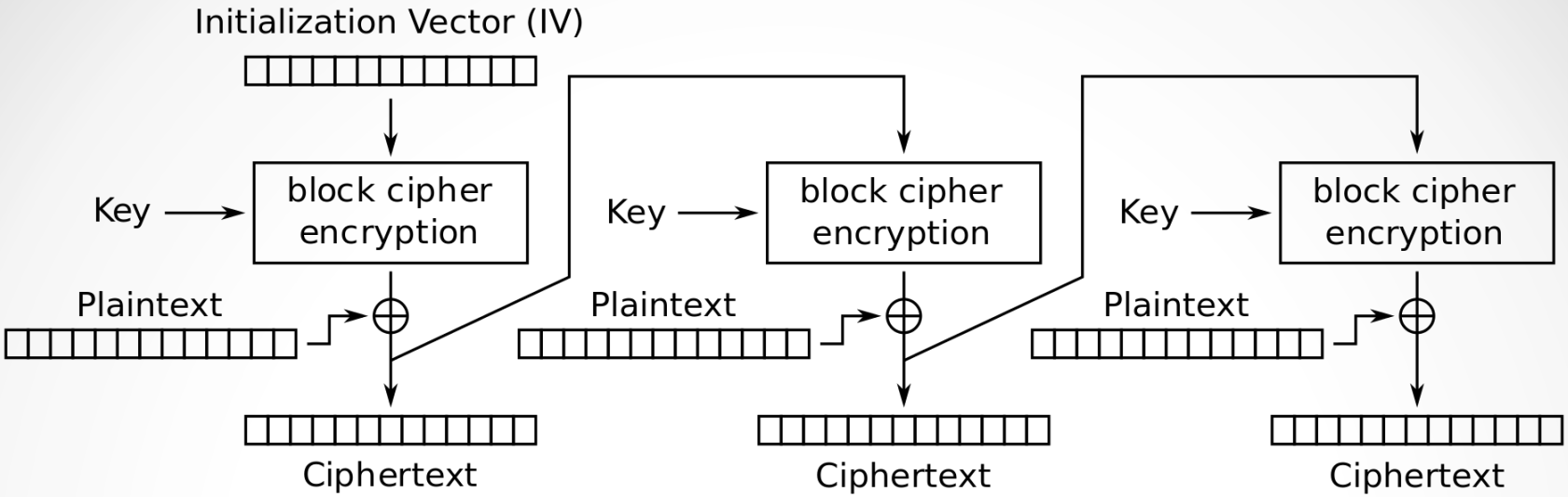
- The following modes create a stream cipher from a block cipher. How is it done?
- Three modes
  - Counter Mode (CTR)
  - Cipher Feedback Mode (CFB)
  - Output Feedback Mode (OFB)



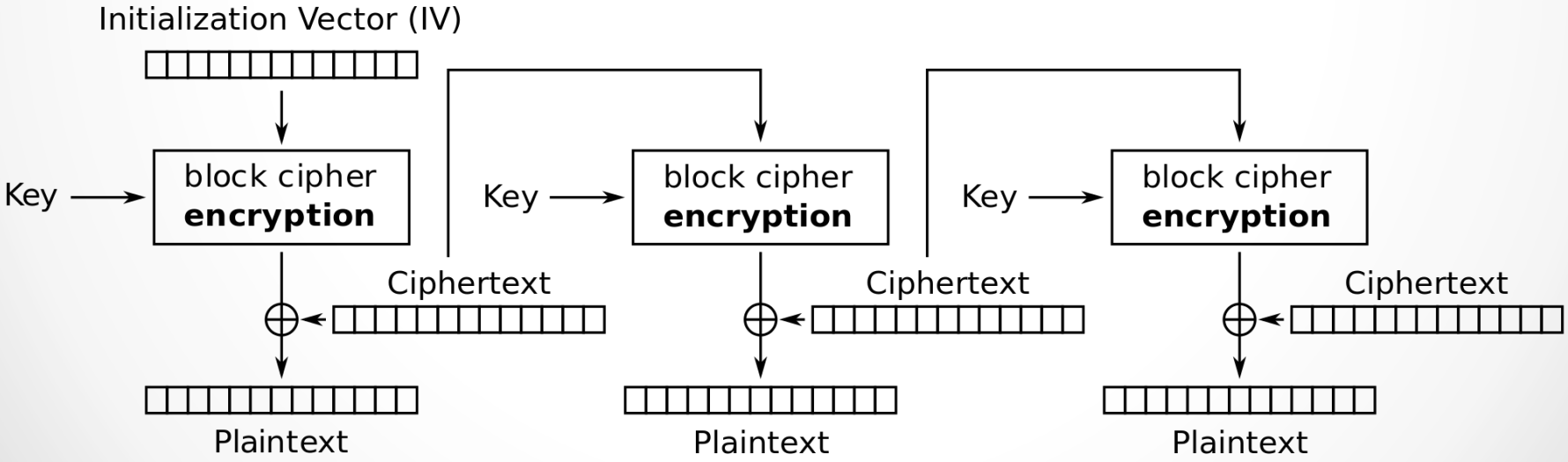
Counter (CTR) mode encryption



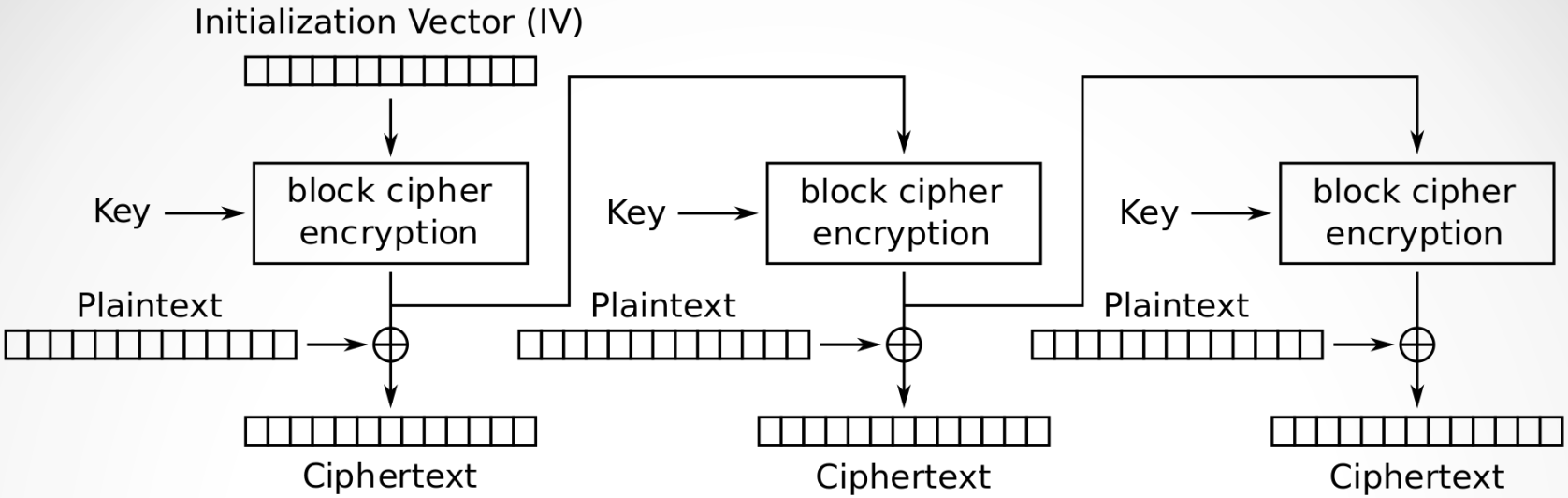
Counter (CTR) mode decryption



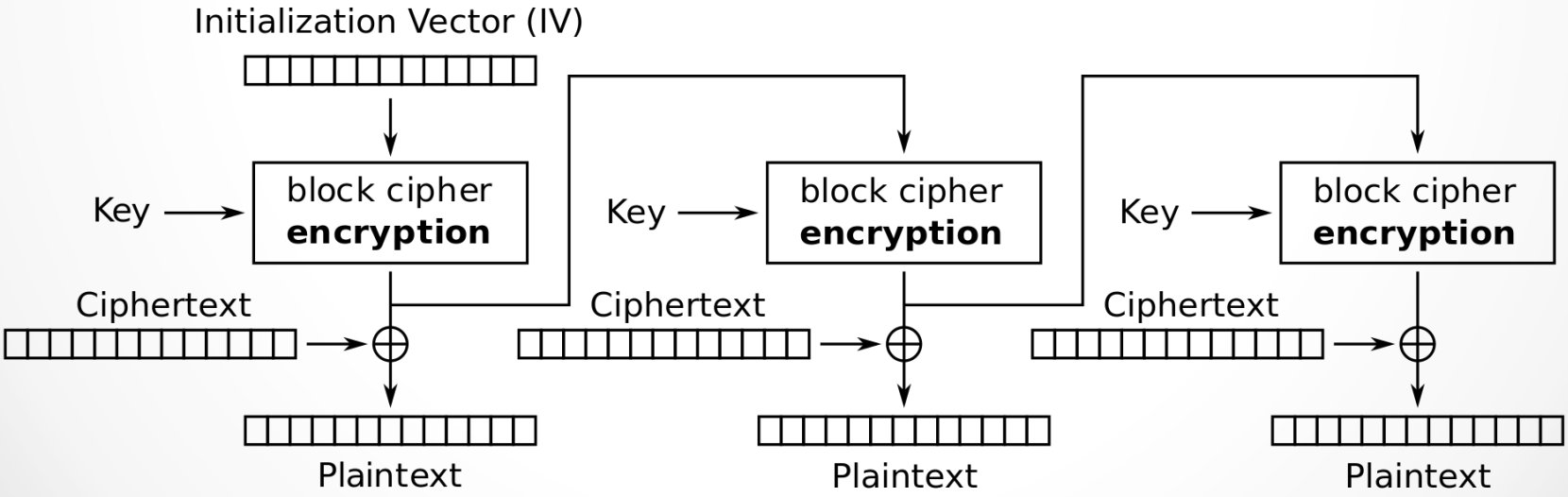
Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption



Output Feedback (OFB) mode encryption



Output Feedback (OFB) mode decryption

Summary

- ECB
  - Simple
  - Don’t have to create/manage an IV
  - Parallel encryption/decryption
  - Reveals patterns in the plaintext – should not use
- CBC
  - Conceals plaintext patterns
  - Requires sequential encryption
  - Parallel decryption

Summary

- Block cipher as stream cipher
  - No need for padding
  - Only have to implement encrypt function
- CTR
  - Preprocessing
  - Parallel encryption/decryption
- CFB
  - Parallel decryption
- OFB
  - Preprocessing
  - Parallel decryption

Padding

Block Ciphers & Padding

- Block ciphers require that the plaintext be a multiple of the block size (ECB and CBC modes)
- Padding is used to make sure that all blocks are “full”
- Both sides need to know the padding scheme

Padding Schemes

What problems can arise?

- Pad with spaces
- Pad with zero (null) characters
- Pad with zero (null) characters
  - Last byte is equal to the number of padding bytes
- Pad with 0x80 followed by zero (null) characters
- Random padding
- Pad with all bytes of the same value

How would this work?

Other Uses for Padding?

- Disguise identical messages
  - Identical messages encrypted with the same key will always produce the same ciphertext
- Disguise message length
  - Pad the message with a random number of bytes to create a random-sized messages
  - All messages are padded to a preset length
- When is padding not required?