Fundamentals of Information & Network Security
ECE 471/571

Lecture #15-16: Modes of Operation

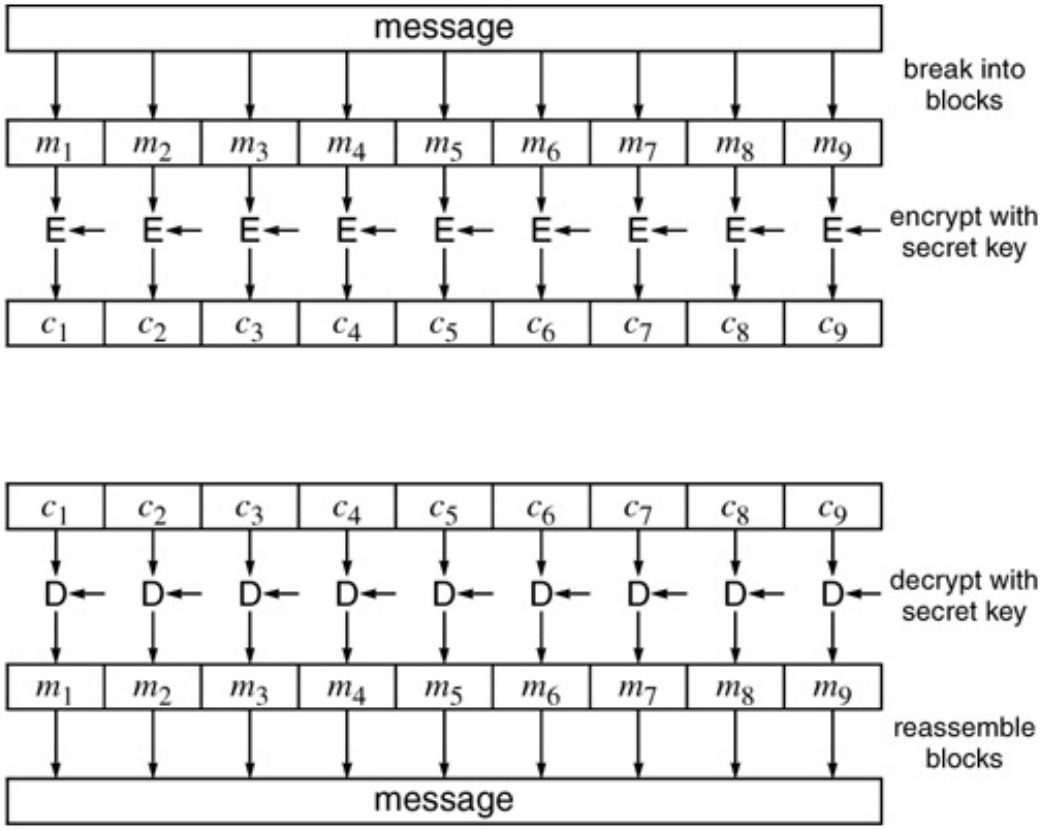
Instructor: Ming Li
Dept of Electrical and Computer Engineering
University of Arizona

Modes of Operation

How to encrypt a message > 64 bits?

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Output Feedback Mode (OFB)
- Cipher Feedback Mode (CFB)
- Counter Mode (CTR)

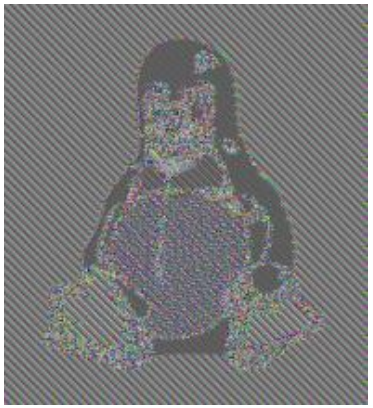
ECB Mode



- Message is broken into 64-bit blocks
- Each block is independently encoded with the same secret key

ECB Mode: Pros and Cons

- Simple
- Error does not propagate
- Identical plaintext → identical ciphertext blocks
- Suitable for?



- Ciphertext blocks can be easily rearranged or modified

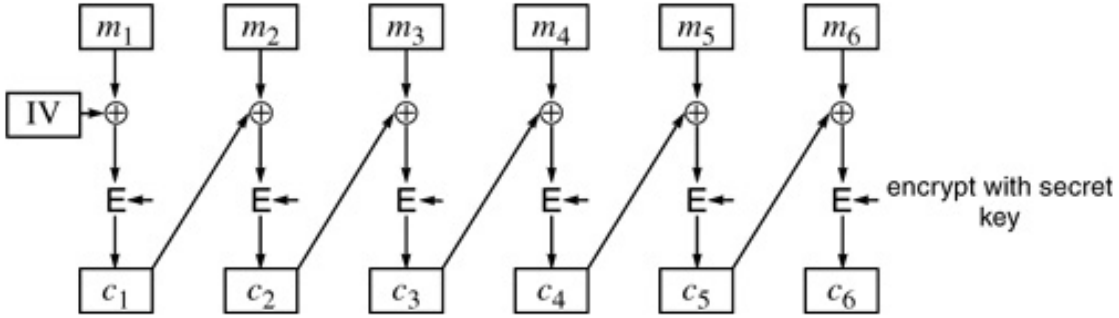
Criteria and properties for evaluating and constructing block cipher modes of operation that are superior to ECB:

- Security
- Diffusion
- Overhead
- Error propagation
- Error recovery

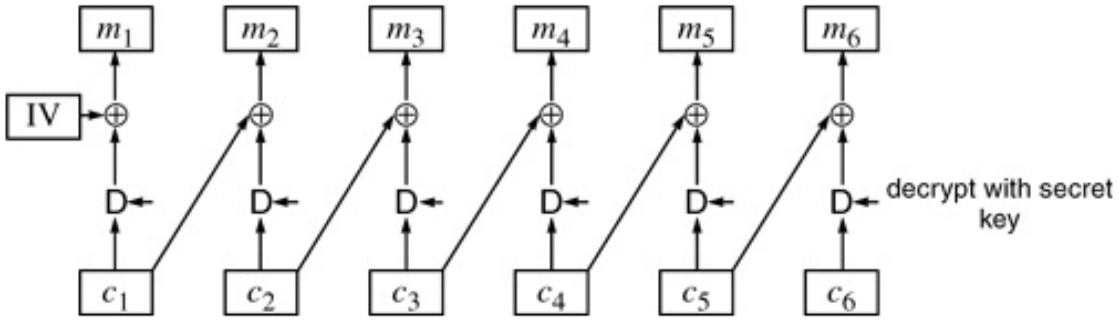
Cipher Block Chaining (CBC) mode

- Identical plaintext → different ciphertext
- Initialization Vector (IV) (Why?)

encryption

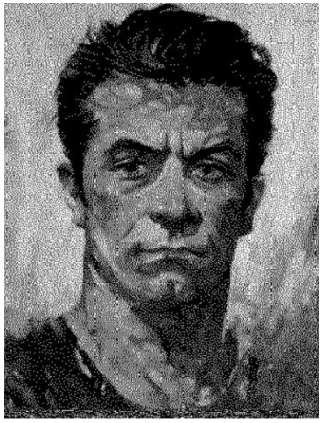


decryption

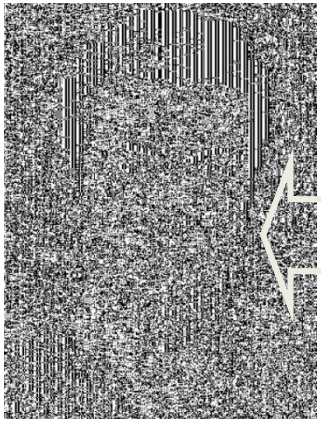


Pros & Cons of CBC

- Identical plaintext → non-identical ciphertext

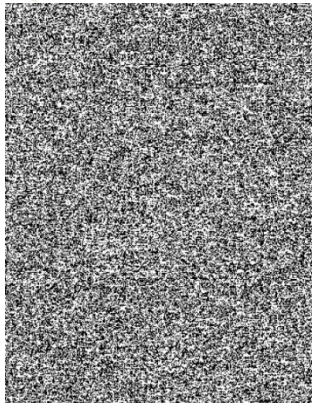


AES in ECB mode



Similar plaintext blocks produce similar ciphertext blocks (not good!)

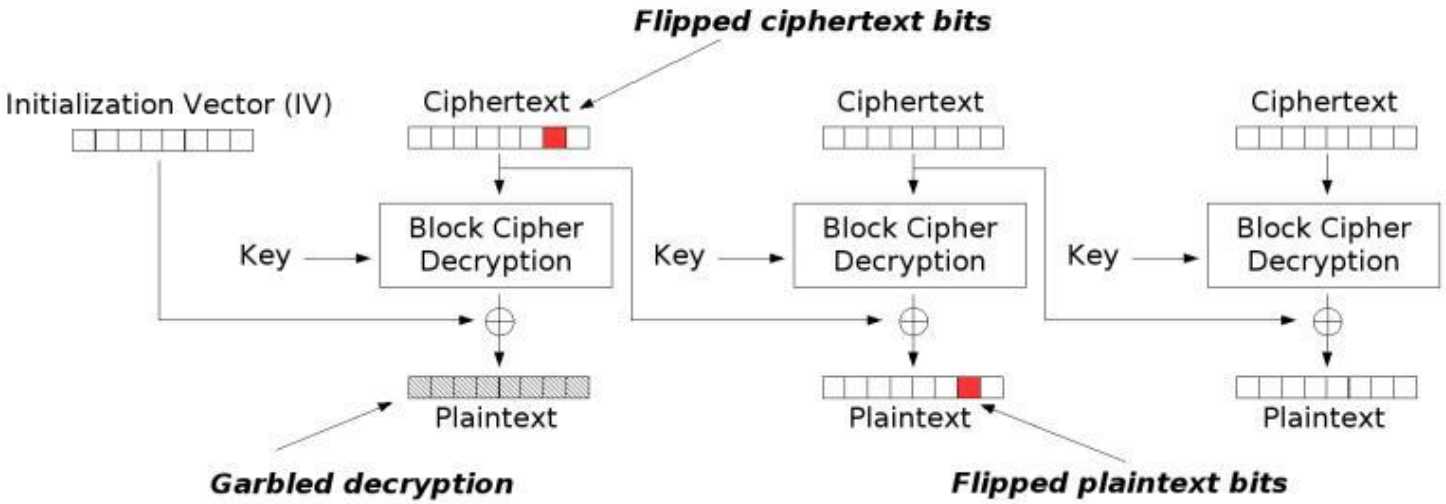
AES in CBC mode



Pros & Cons of CBC

- Identical plaintext → non-identical ciphertext
- Can be used to construct message authentication codes
- IV: needs to be shared between sender and receiver (how?)
  - Does it need to be a secret?
  - Never reuse: why?
- Modification attack

CBC Modification attack



Modification attack on CBC

CBC Modification Attack

Original message

```
Tacker, Jo A        System Security Officer      54,122.10
|       |      |       |        |        |      |      |        |
```

Decrypted message
after modification

```
Tacker, Jo A        System Security Of#f8Ts9(*   74,122.10
|       |      |       |        |        |      |      |        |
```

Output FeedBack Mode (OFB)

$$z_i = e_K(z_{i-1}), y_i = x_i \oplus z_i$$

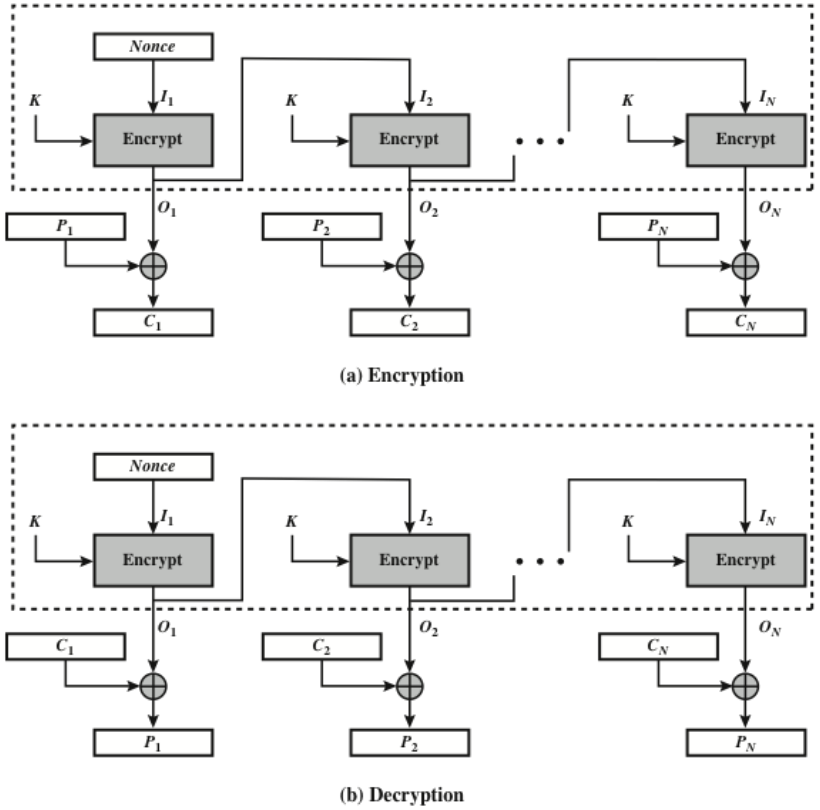


Figure 7.6 Output Feedback (OFB) Mode

- Stream cipher, like a one-time pad: z_0 is a random 64-bit IV

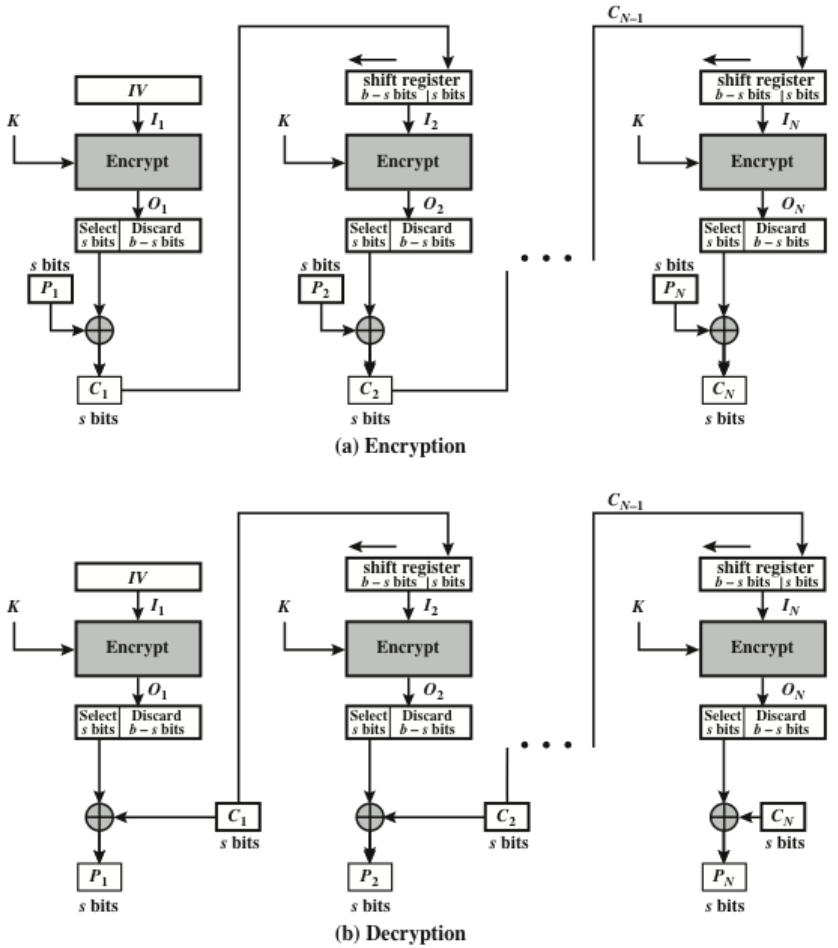
Pros and Cons of OFB

- One-time pad can be generated in advance, only XOR operations are performed in real-time
- Can be used as a pseudorandom number generator
- Bit errors do not propagate: error in one ciphertext block only garbles the corresponding plaintext block
- Suitable for use in stream-oriented transmission over noisy channel (e.g., satellite communication)

- Plaintext modification attack: if attacker knows <plaintext, ciphertext>, he can XOR the plaintext and ciphertext, and XOR the result with any message of his choosing
- Must not reuse the same IV (Why?)

Cipher Feedback Mode (CFB)

$$z_i = e_K(y_{i-1}), \quad y_i = x_i \oplus z_i,$$



- Also a stream cipher

Figure 7.5 *s*-bit Cipher Feedback (CFB) Mode

Pros and Cons of CFB

- Self-resynchronization:
  If the number of lost bits is a multiple integer of *k* then *b* / *k* additional blocks are distorted before synchronization is re-established.

- Less subject to tampering

- One-time pad cannot be pre-computed, encryption needs to be done in real-time

- Error in a ciphertext block propagates: it garbles the next plaintext block

- Not used in practice

Counter Mode (CTR)

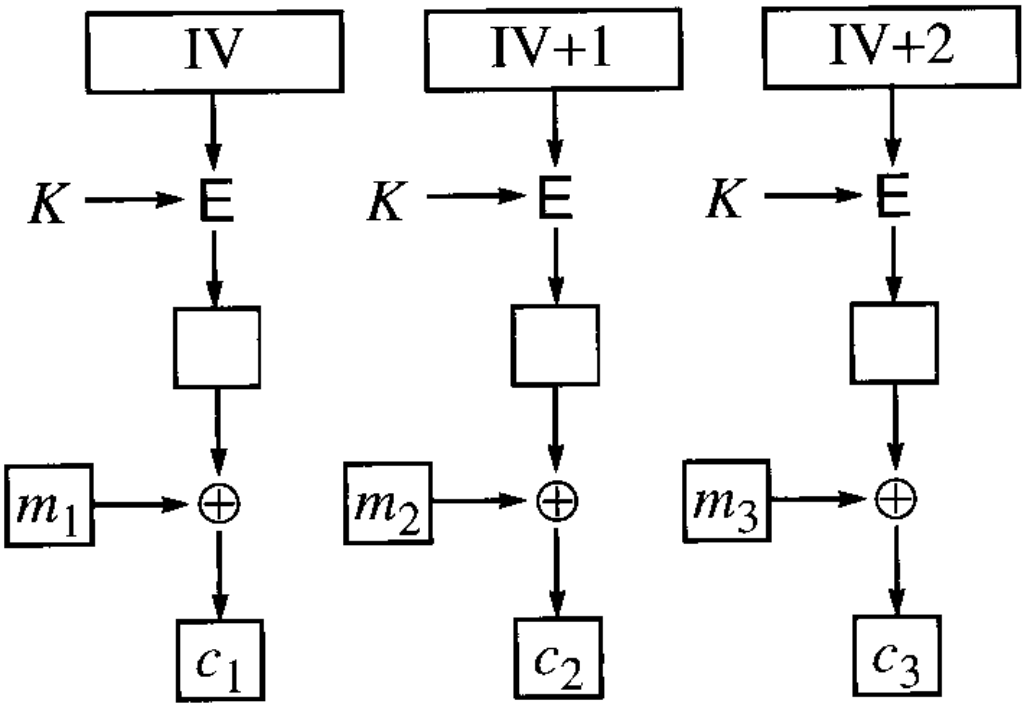


Figure 4-10. Counter Mode (CTR)

- Instead of chaining the encryption of one-time pad, the IV is incremented and encrypted to get successive blocks of the one-time pad

Pros and Cons of CTR

- One-time pad can be pre-computed
- Can start encrypting/decrypting at any point
  - Ideal for applications requiring random access
- Efficiency:
  - No chaining, encryption/decryption can be done in parallel on multiple blocks, without depending on other blocks
  - Simple, use only encryption

- Must not reuse the same IV or key, same as OFB

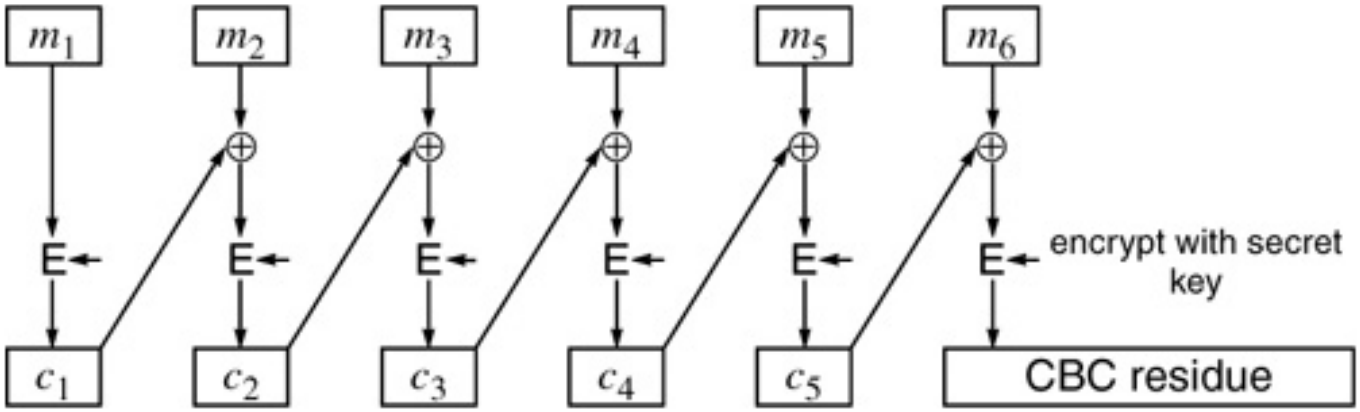
(Why?)

Summary

- Use of modes
  - ECB : key management, useless for file encryption
  - CBC : File encryption, useful for MAC
  - OFB : stream cipher, pseudorandom number generator
  - CFB : stream cipher, more secure
  - CTR : support random access, parallel computation

Integrity: Generating MACs

- Protect against undetected modifications
- Plaintext + CBC residue (when message not secret)



Privacy and Integrity Together (1)

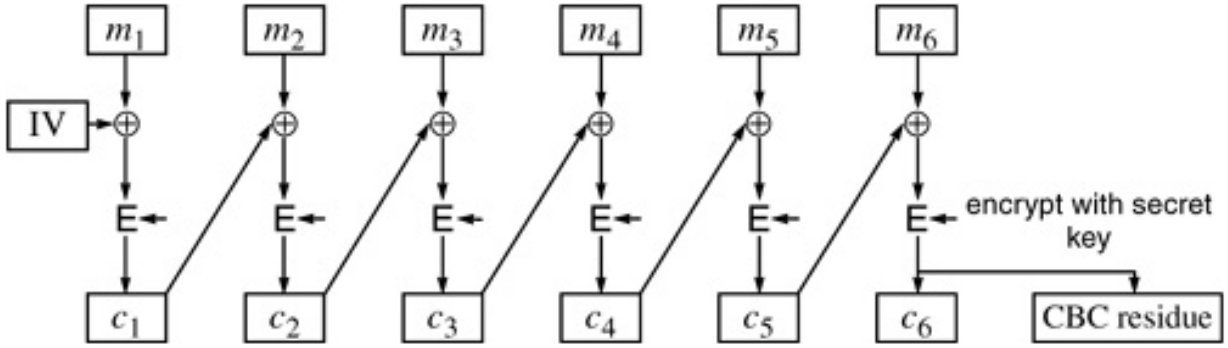
- Privacy: CBC encryption

  Integrity: CBC residue

- Ciphertext + CBC residue ?
- Encrypt {plaintext + CBC residue} ?
- Encrypt {plaintext + CRC} ?

Ciphertext + CBC Residue

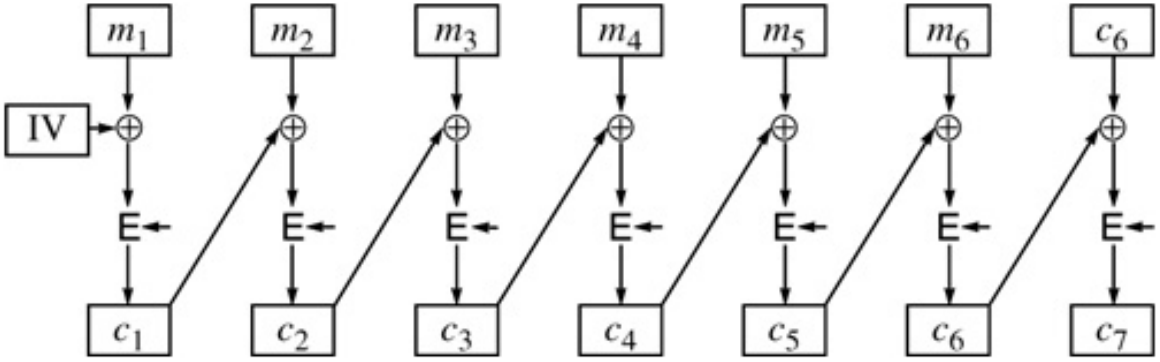
Figure 4-12. Cipher Block Chaining Encryption plus CBC Residue



- Problem?

Encrypt {plaintext + CBC residue}

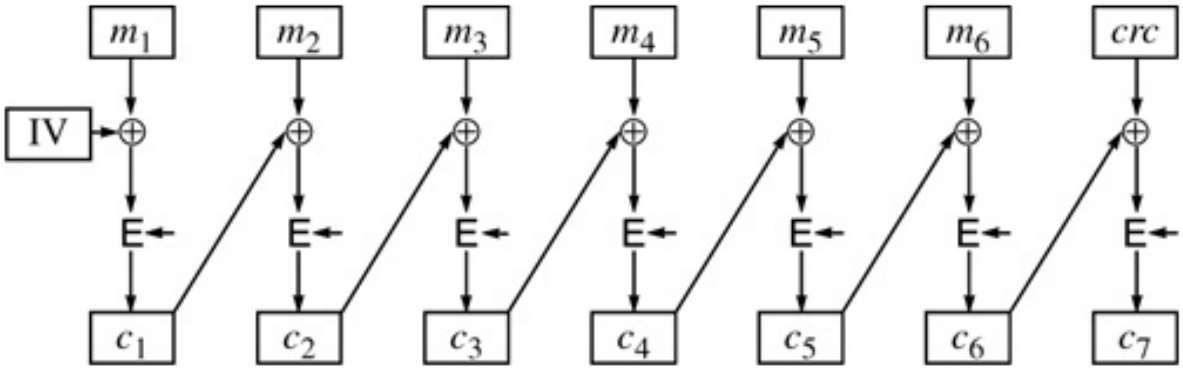
Figure 4-13. Cipher Block Chaining Encryption of Message with CBC Residue



- Problem?

Encrypt {plaintext + CRC}

Figure 4-14. Cipher Block Chaining Encryption of Message with CRC



- Longer CRC maybe Okay

Privacy and Integrity: The Do's

- Privacy: CBC encryption + Integrity: CBC residue, but with different keys
- CBC + weak cryptographic checksum
- CBC + CBC residue with related keys
- CBC + cryptographic hash: keyed hash preferred
- ……