CSCE 465 Computer & Network Security

Instructor: Abner Mendoza

Secret Key Cryptography (II)

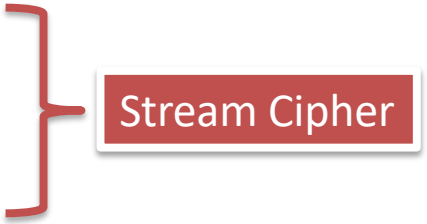
Modes of Operation

Roadmap

- Modes of operation
- Triple DES
- More on Stream cipher

Processing with Block Ciphers

- Most ciphers work on blocks of fixed (small) size
- How to encrypt long messages?
- Modes of operation
  - ECB (Electronic Code Book)
  - CBC (Cipher Block Chaining)
  - OFB (Output Feedback)
  - CFB (Cipher Feedback)
  - CTR (Counter)



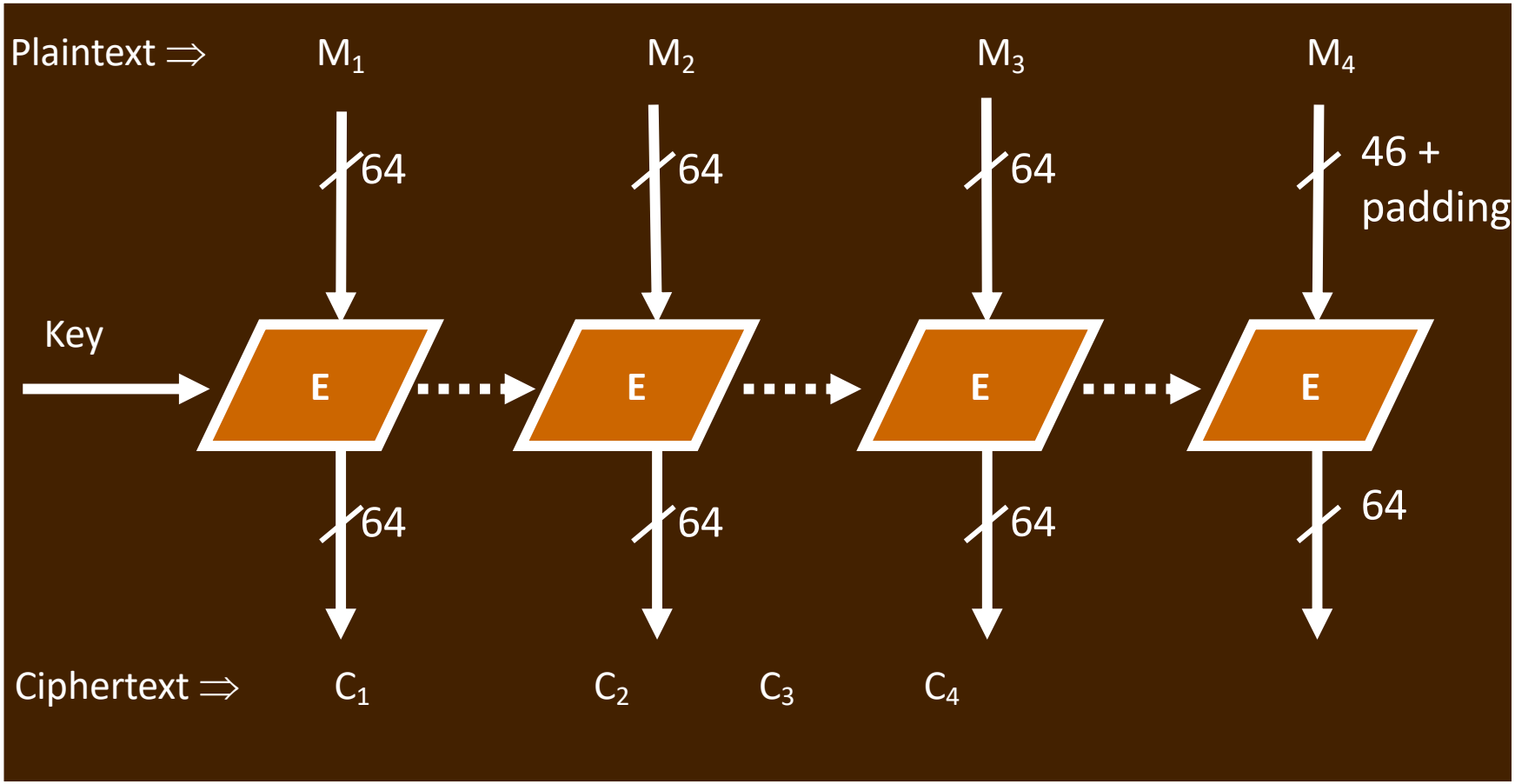
Issues for Block Chaining Modes

- Information leakage
  - Does it reveal info about the plaintext blocks?
- Ciphertext manipulation
  - Can an attacker modify/rearrange ciphertext block(s) in a way that will produce a predictable/desired change in the decrypted plaintext block(s)?
  - Note: assume the structure of the plaintext is known, e.g., first block is employee #1 salary, second block is employee #2 salary, etc.

Issues… (Cont'd)

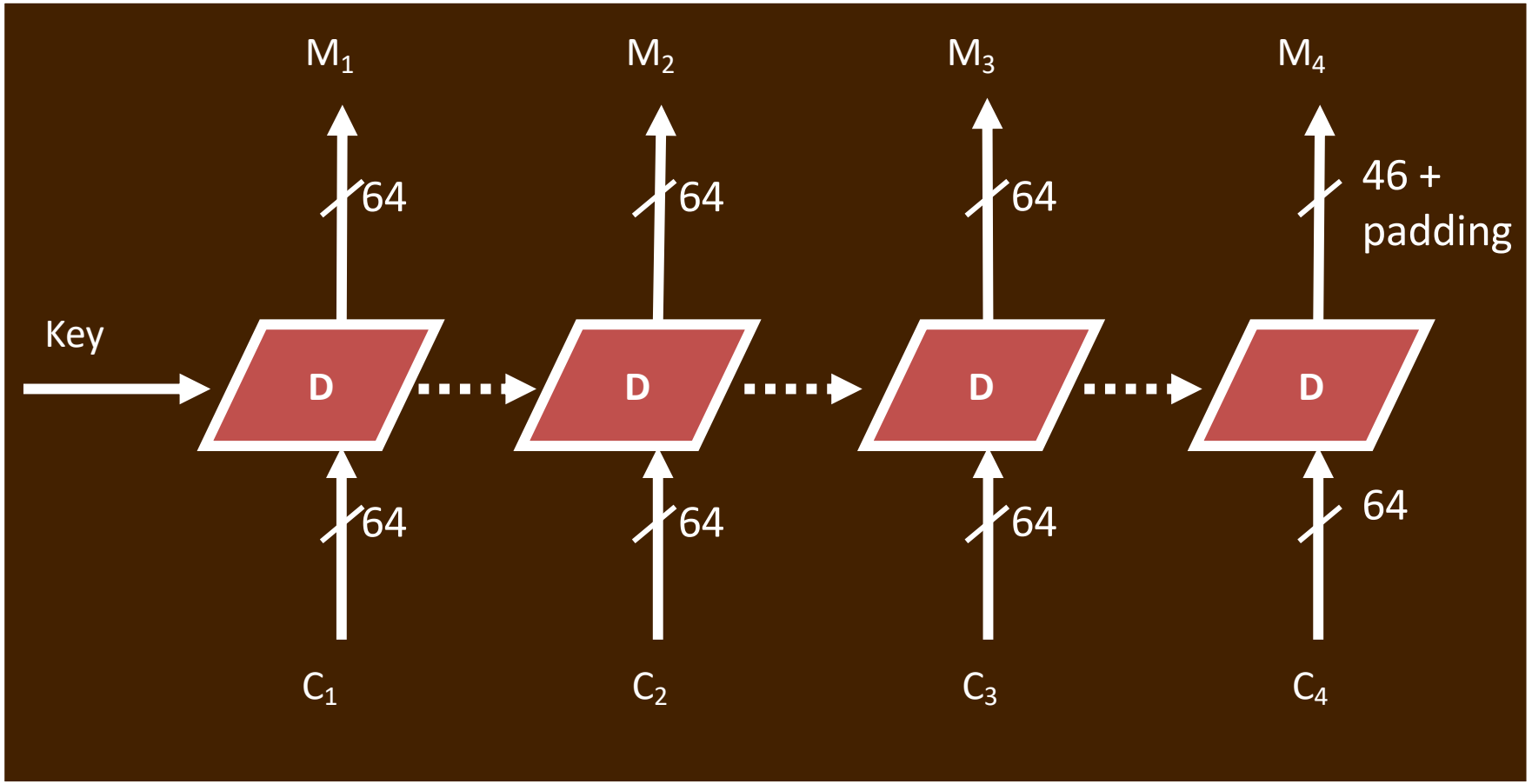
- Parallel/Sequential
  - Can blocks of plaintext (ciphertext) be encrypted (decrypted) in parallel?
- Error propagation
  - If there is an error in a plaintext (ciphertext) block, will there be an encryption (decryption) error in more than one ciphertext (plaintext) block?

Electronic Code Book (ECB)



- The easiest mode of operation; each block is independently encrypted

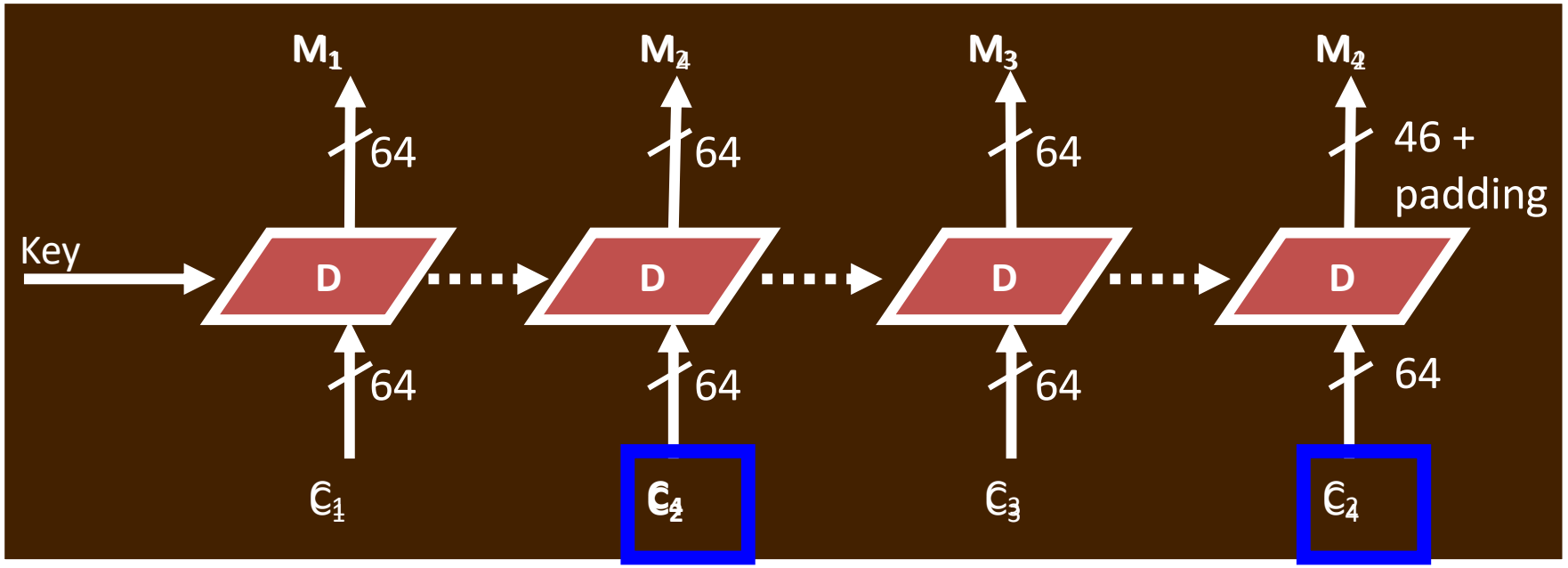
ECB Decryption



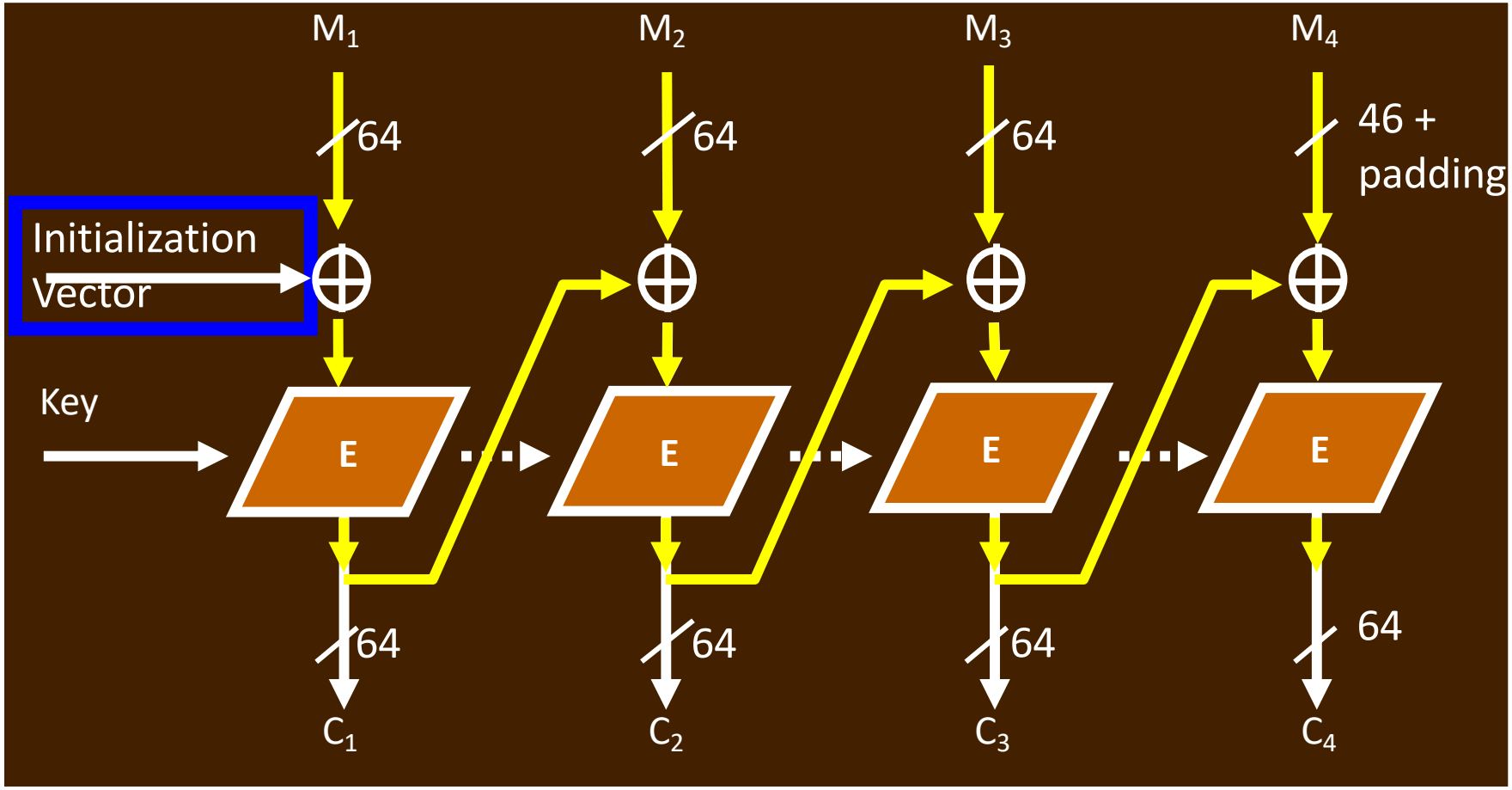
- Each block is independently decrypted

ECB Properties

- Does information leak?
- Can ciphertext be manipulated profitably?
- Parallel processing possible?
- Do ciphertext errors propagate?



Cipher Block Chaining (CBC)

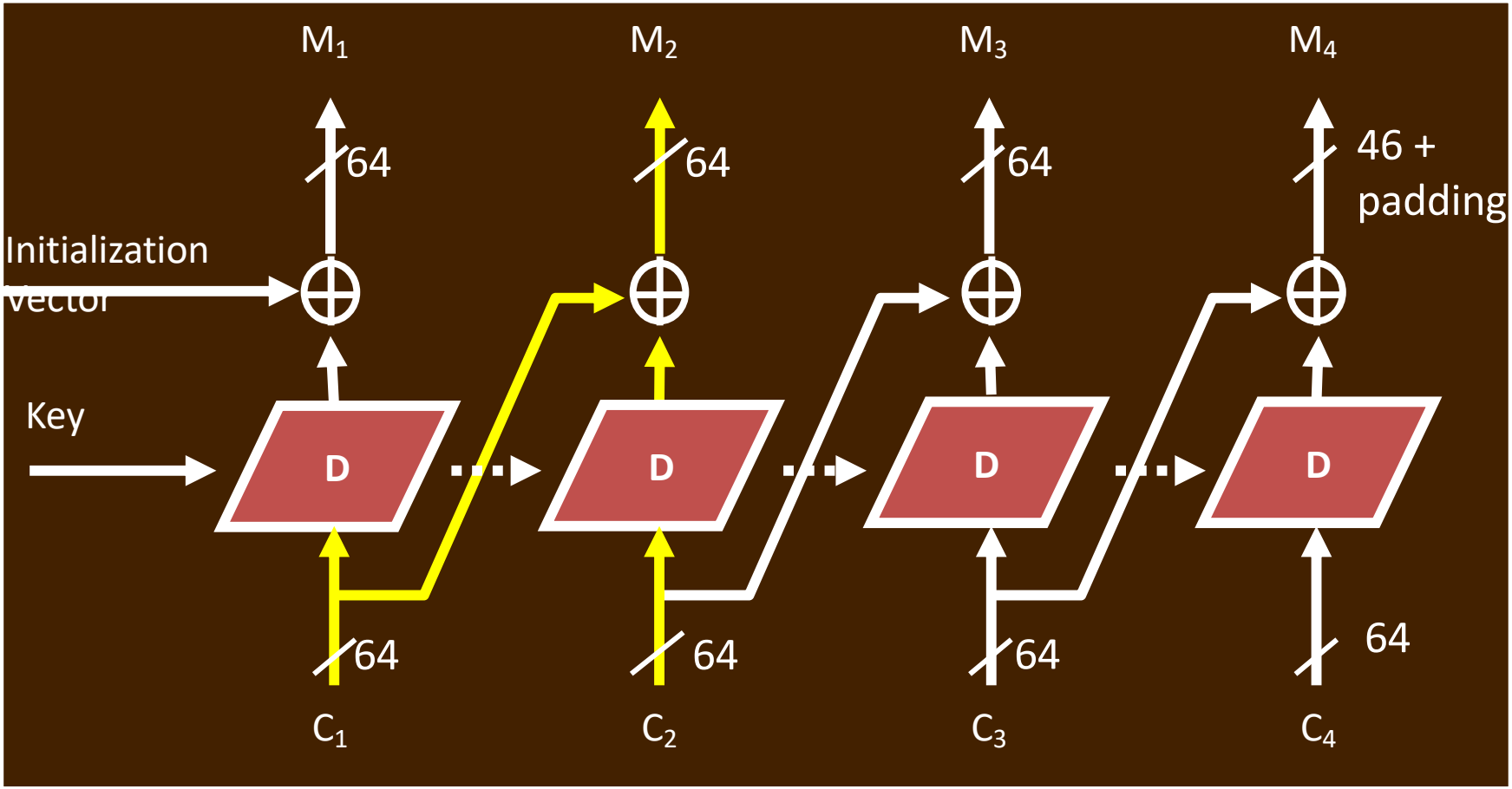


- Chaining dependency: each ciphertext block depends on all preceding plaintext blocks

Initialization Vectors

- Initialization Vector (IV)
  - Used along with the key; not secret
  - For a given plaintext, changing either the key, or the IV, will produce a different ciphertext
  - Why is that useful?
- IV generation and sharing
  - Random; may transmit with the ciphertext
  - Incremental; predictable by receivers

CBC Decryption

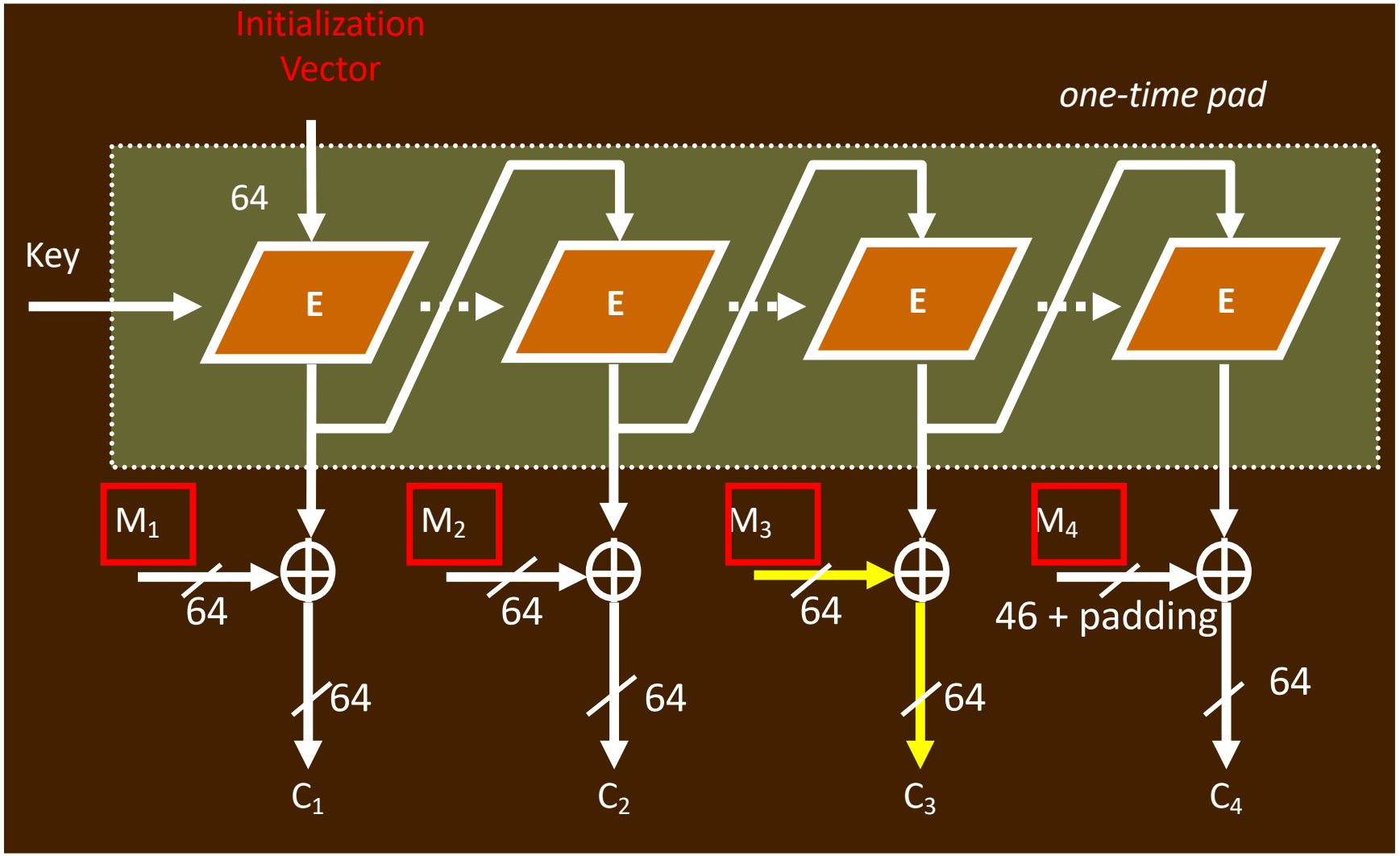


- How many ciphertext blocks does each plaintext block depend on?

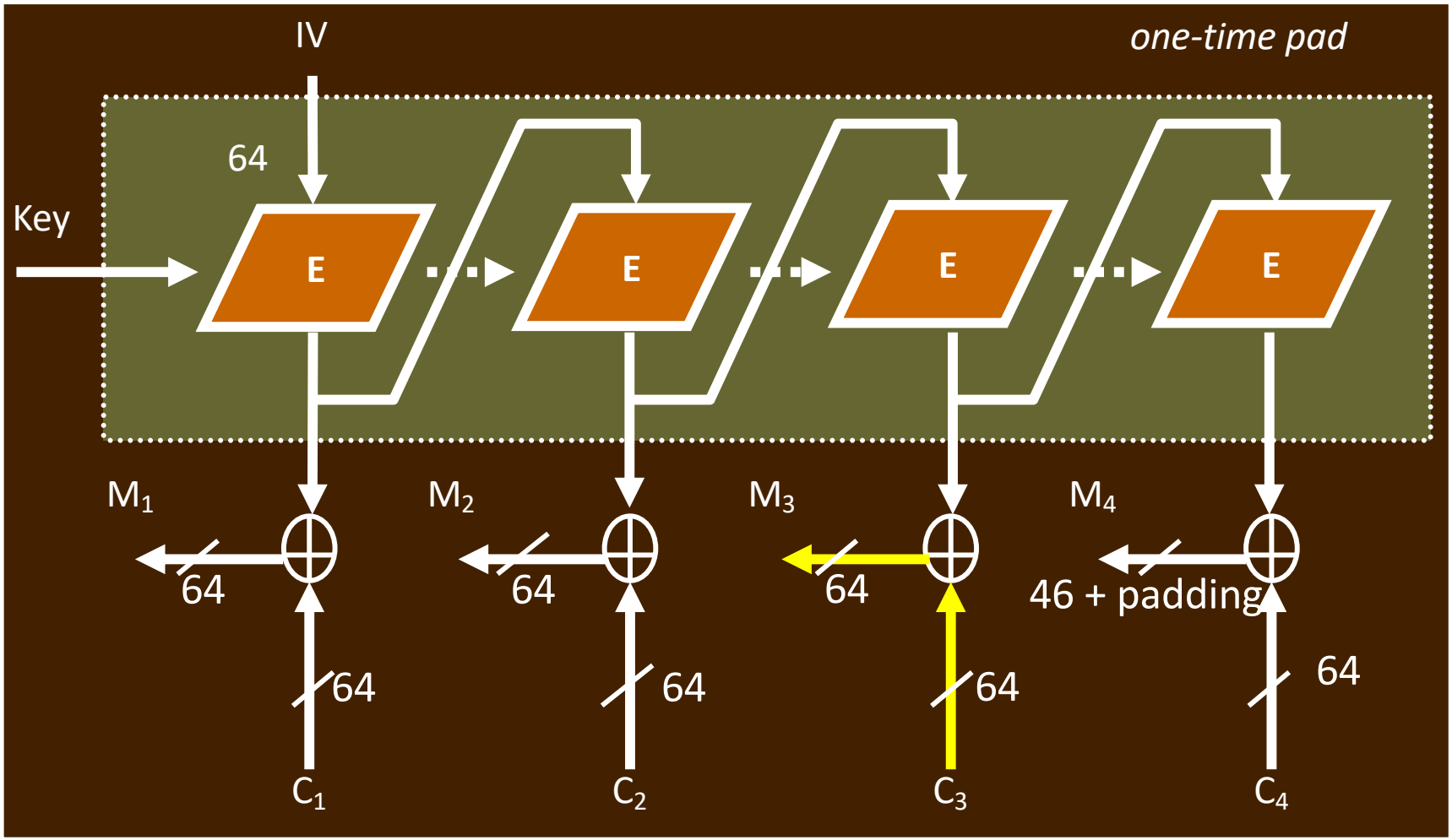
CBC Properties

- Does information leak?
  - Identical plaintext blocks will produce different ciphertext blocks
- Can ciphertext be manipulated profitably?
  - ???
- Parallel processing possible?
  - no (encryption), yes (decryption)
- Do ciphertext errors propagate?
  - yes (encryption), a little (decryption)

Output Feedback Mode (OFB)



OFB Decryption



No block decryption required!

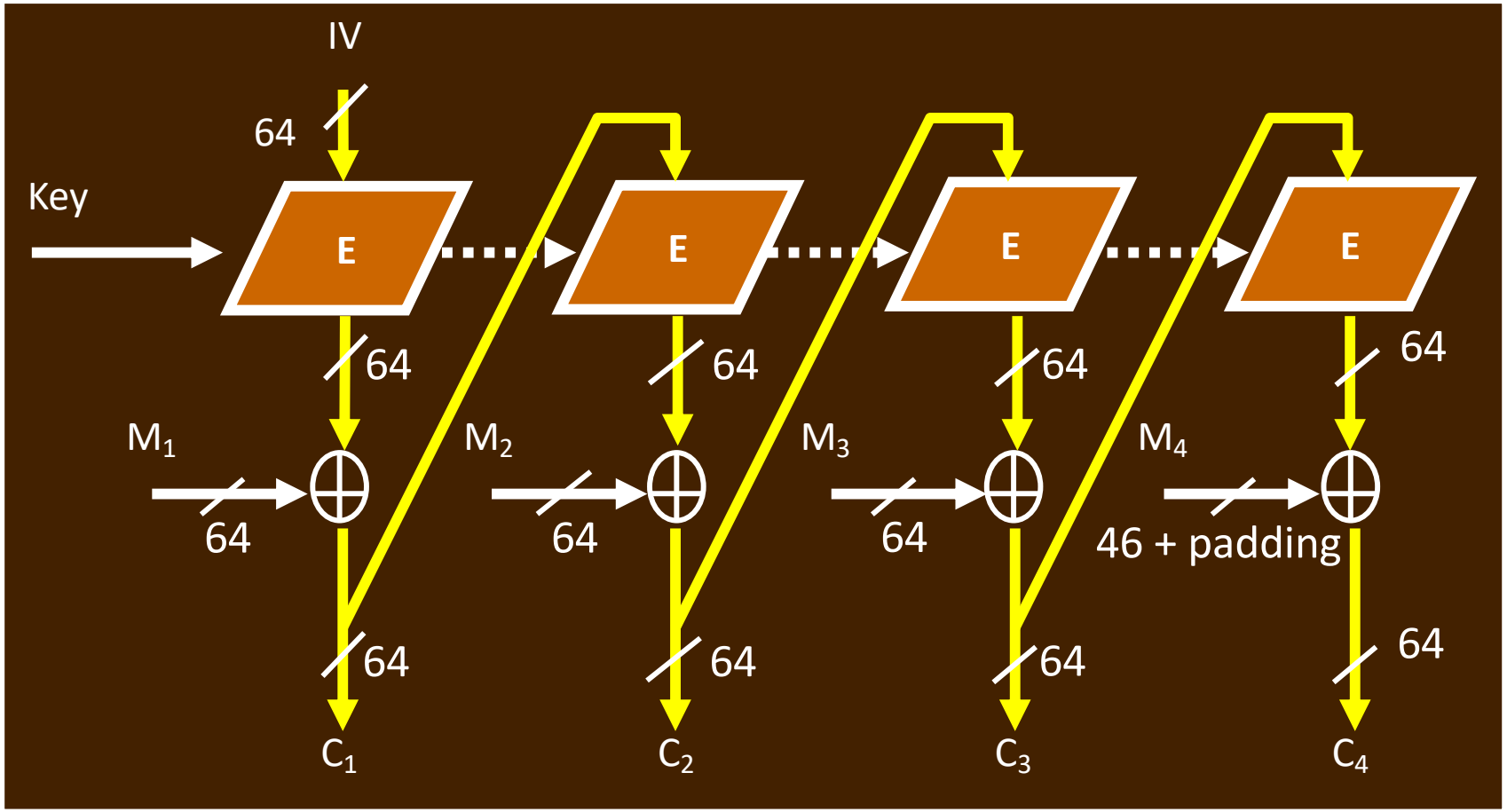
OFB Properties

- Does information leak?
  - identical plaintext blocks produce different ciphertext blocks
- Can ciphertext be manipulated profitably?
  - ???
- Parallel processing possible?
  - no (generating pad), yes (XORing with blocks)
- Do ciphertext errors propagate?
  - ???

OFB … (Cont'd)

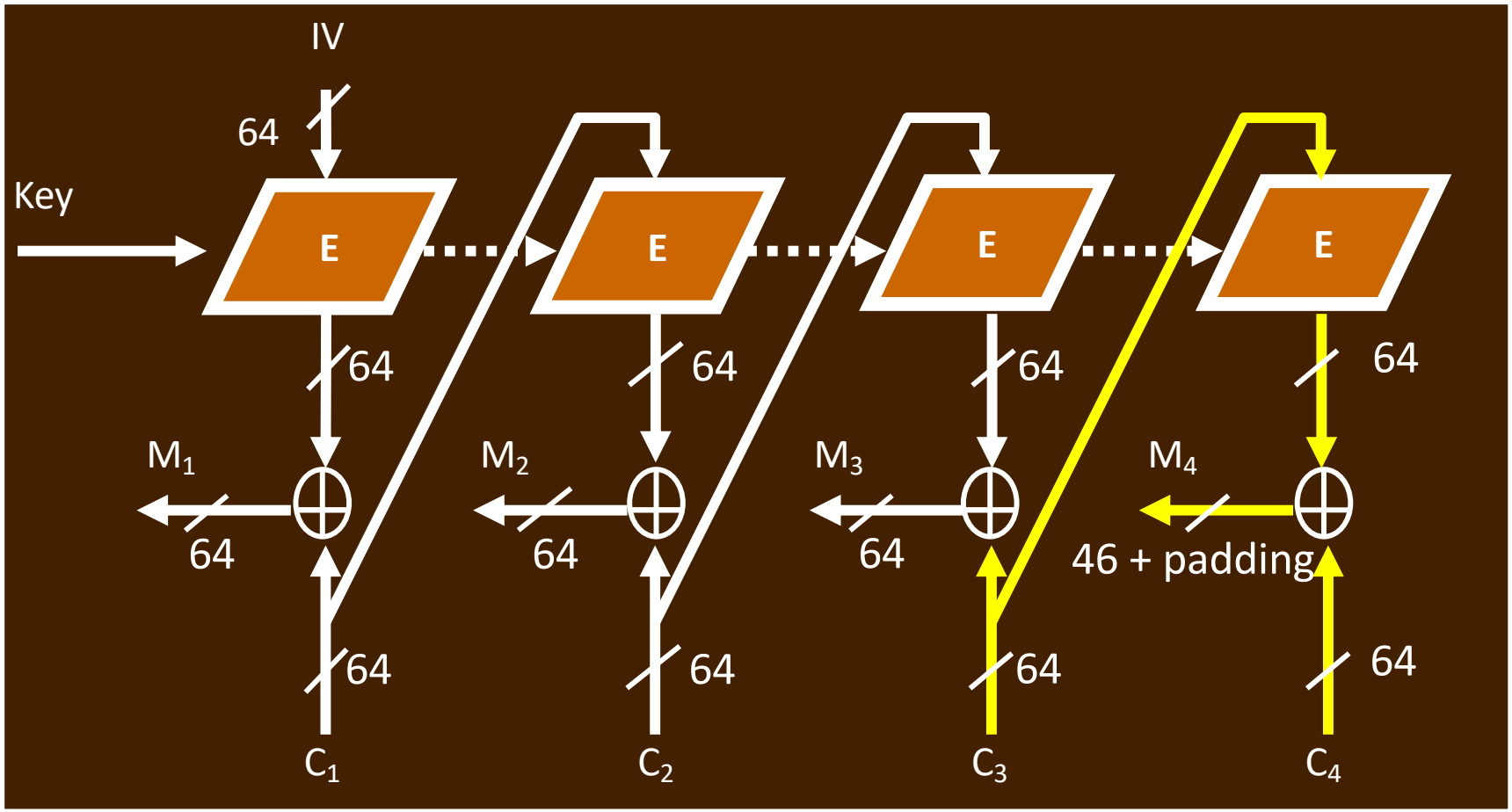
- OFB Advantages
  - Allow pre-computing of pseudo-random stream (One-Time Pad); XOR can be implemented very efficiently
  - Allow in-time encrypt/decrypt due to bit-wise computation (versus the fixed blocks)
- If you know one plaintext/ciphertext pair, can easily derive the one-time pad that was used
  - i.e., should not reuse a one-time pad!
  - Conclusion: IV must be different every time
- Another issue
  - If a bad guy knows the plaintext and ciphertext, can he send arbitrary (valid) messages?

Cipher Feedback Mode (CFB)



- Ciphertext block C_j depends on all preceeding blocks

CFB Decryption

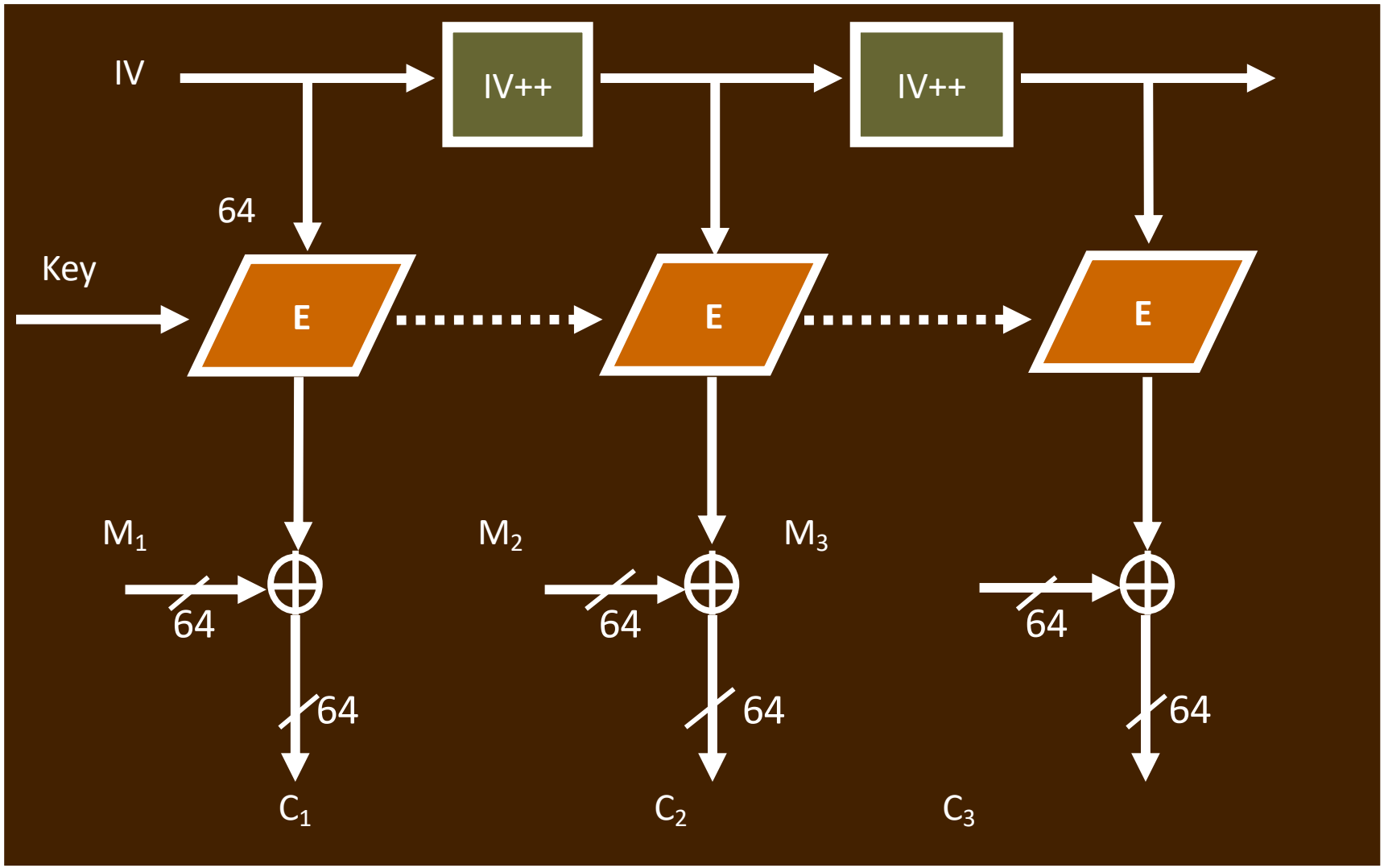


- No block decryption required!

CFB Properties

- Does information leak?
  - Identical plaintext blocks produce different ciphertext blocks
- Can ciphertext be manipulated profitably?
  - ???
- Parallel processing possible?
  - no (encryption), yes (decryption)
- Do ciphertext errors propagate?
  - ???

Counter Mode (CTR)



CTR Mode Properties

- Does information leak?
  - Identical plaintext block produce different ciphertext blocks
- Can ciphertext be manipulated profitably
  - ???
- Parallel processing possible
  - Yes (both generating pad and XORing)
- Do ciphertext errors propagate?
  - ???
- Allow decryption the ciphertext at any location
  - Ideal for random access to ciphertext

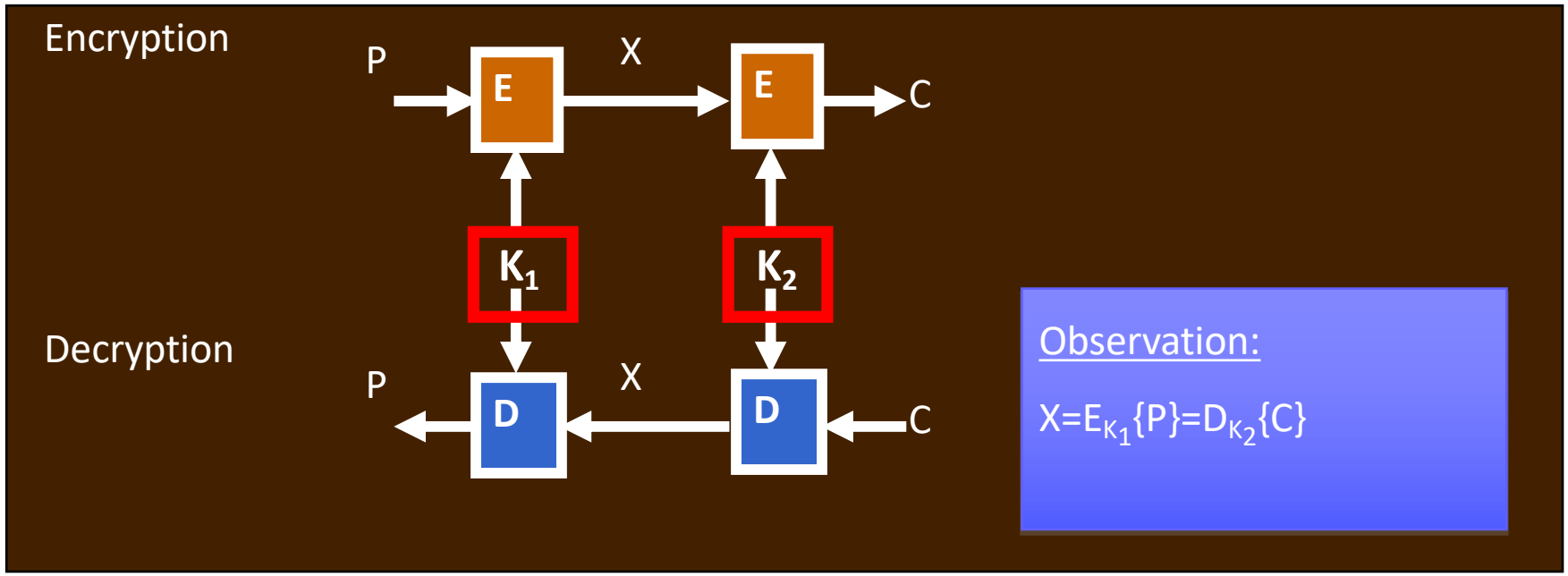
Triple DES

Stronger DES

- Major limitation of DES
  - Key length is too short
- Can we apply DES multiple times to increase the strength of encryption?

Double Encryption with DES

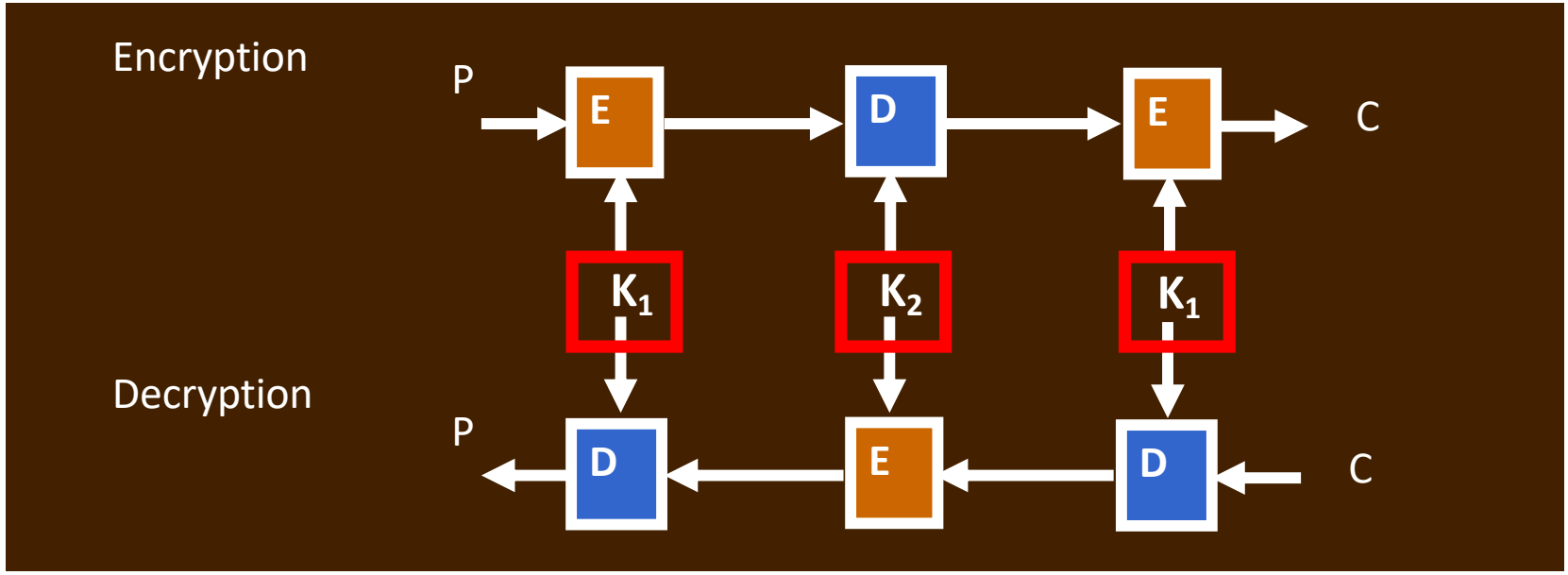
- Encrypt the plaintext twice, using two different DES keys
- Total key material increases to 112 bits
  - is that the same as key strength of 112 bits?



Concerns About Double DES

- Wasn't clear at the time if DES was a group (it's not)
  - If it were, then $E_{k2}(E_{k1}(P)) \equiv E_{k3}(P)$, for all P
  - Not good?
- Possible attack (better than brute force): meet-in-the-middle
  - A known-plaintext attack (check the textbook for detail)

Triple Encryption (Triple DES-EDE)



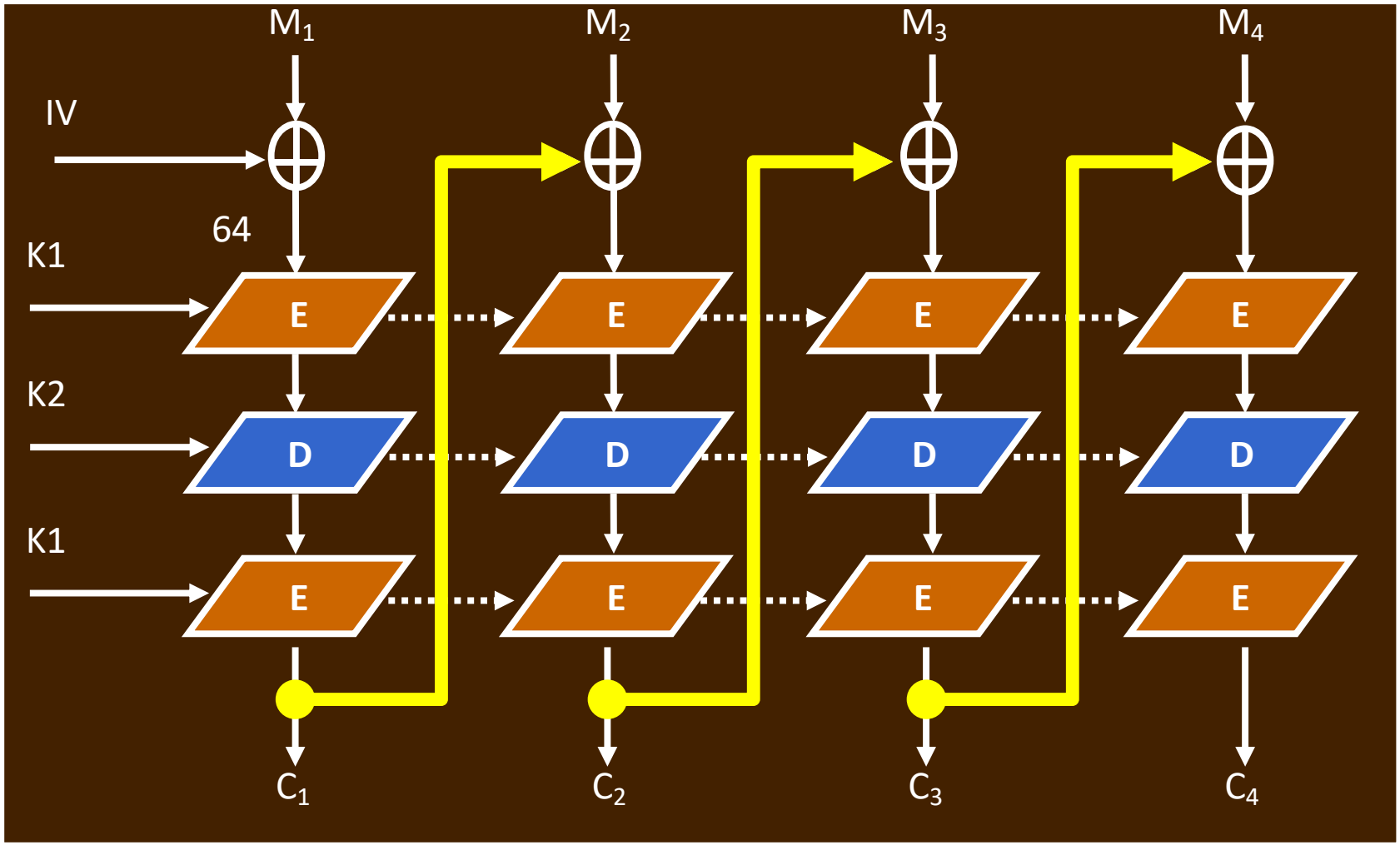
- Why not E-E-E?
  - again, wasn't clear if DES was a group
- Apply DES encryption/decryption three times
  - why not 3 different keys?
  - why not the same key 3 times?

Triple DES (Cont'd)

- Widely used
  - equivalent strength to using a 112 bit key
  - strength about 2^{110} against M-I-T-M attack
- However: inefficient / expensive to compute
  - one third as fast as DES on the same platform, and DES is already designed to be slow in software

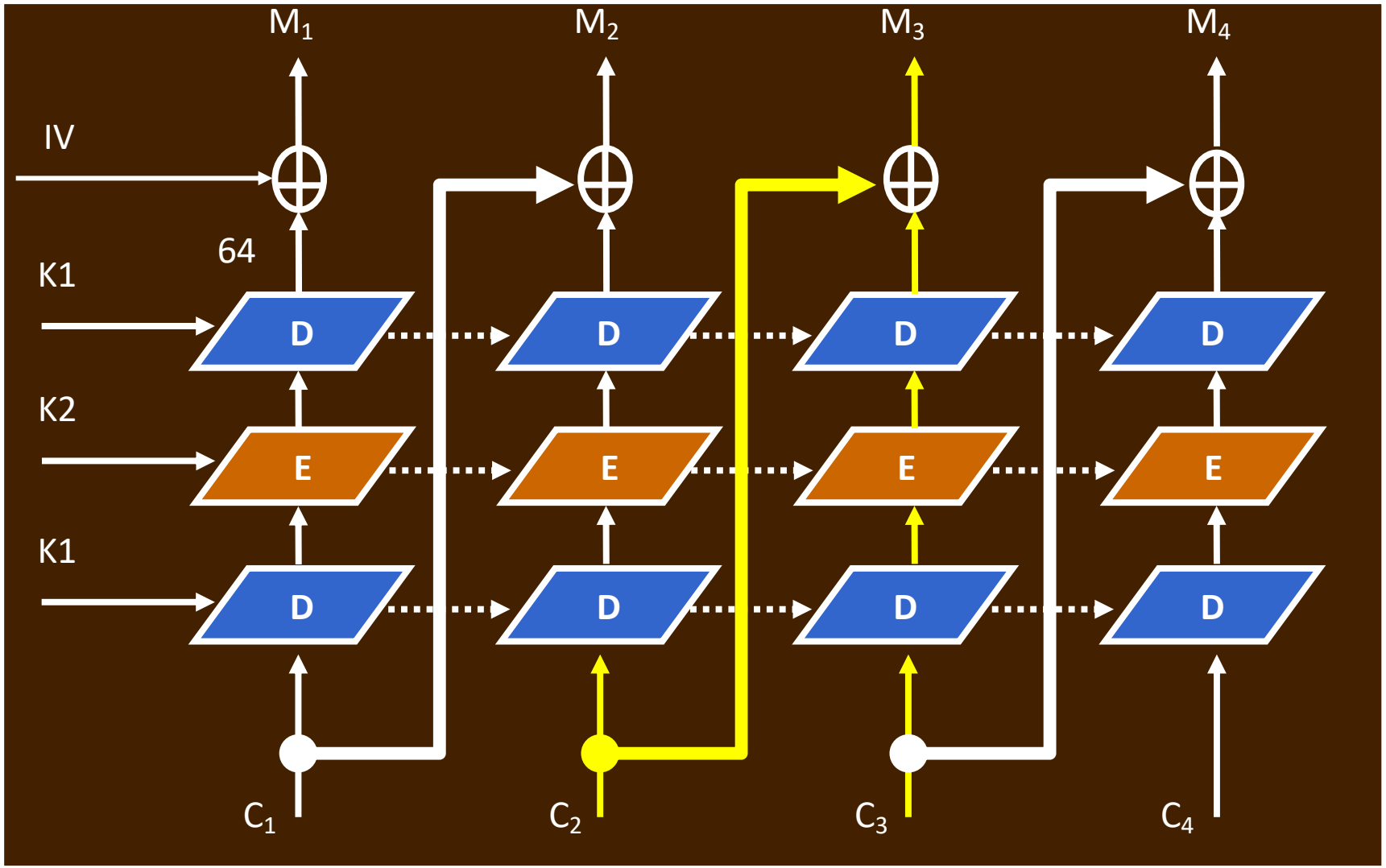
- Next question: how is block chaining used with triple-DES?

3DES-EDE: Outside Chaining Mode

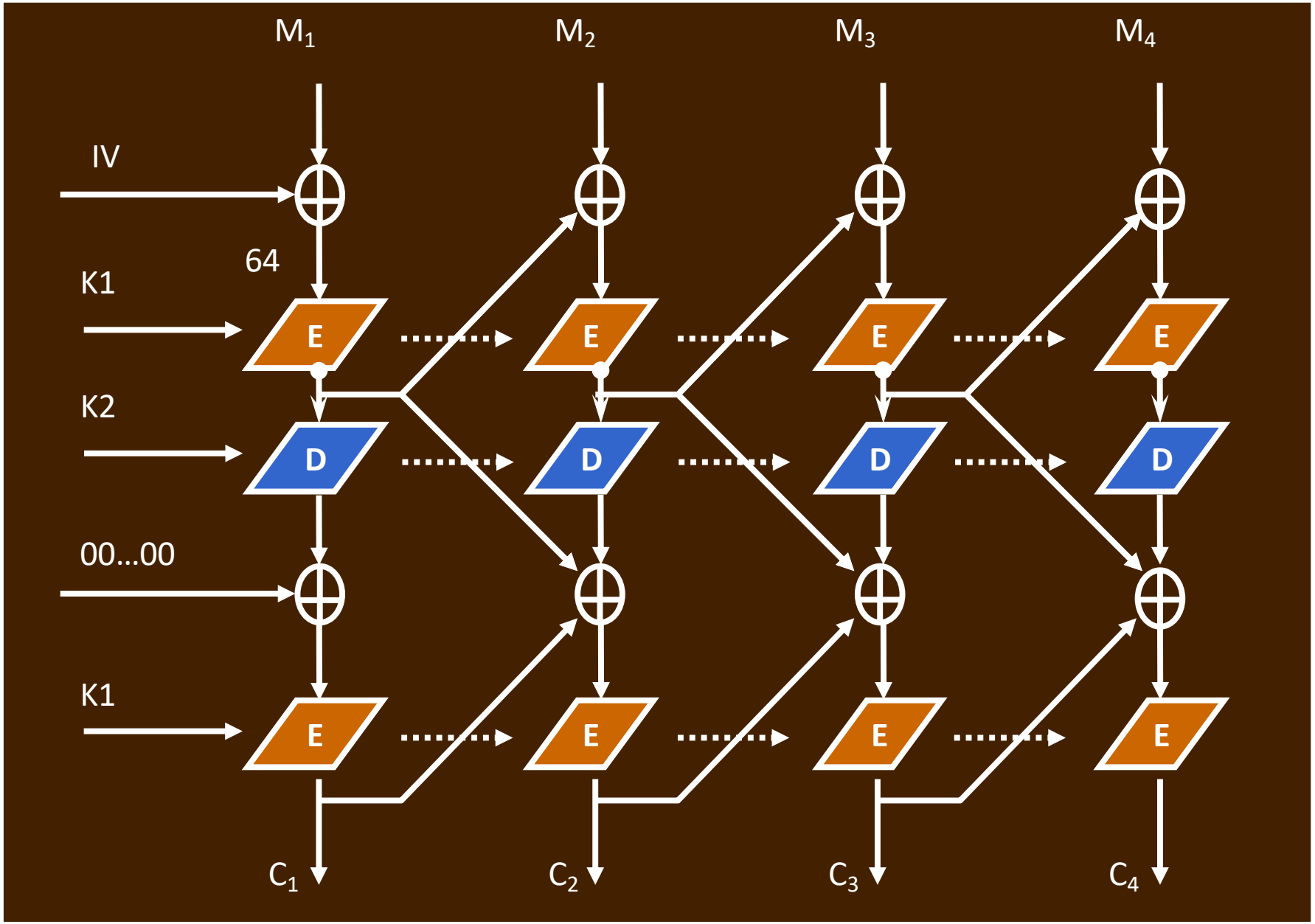


- What basic chaining mode is this?

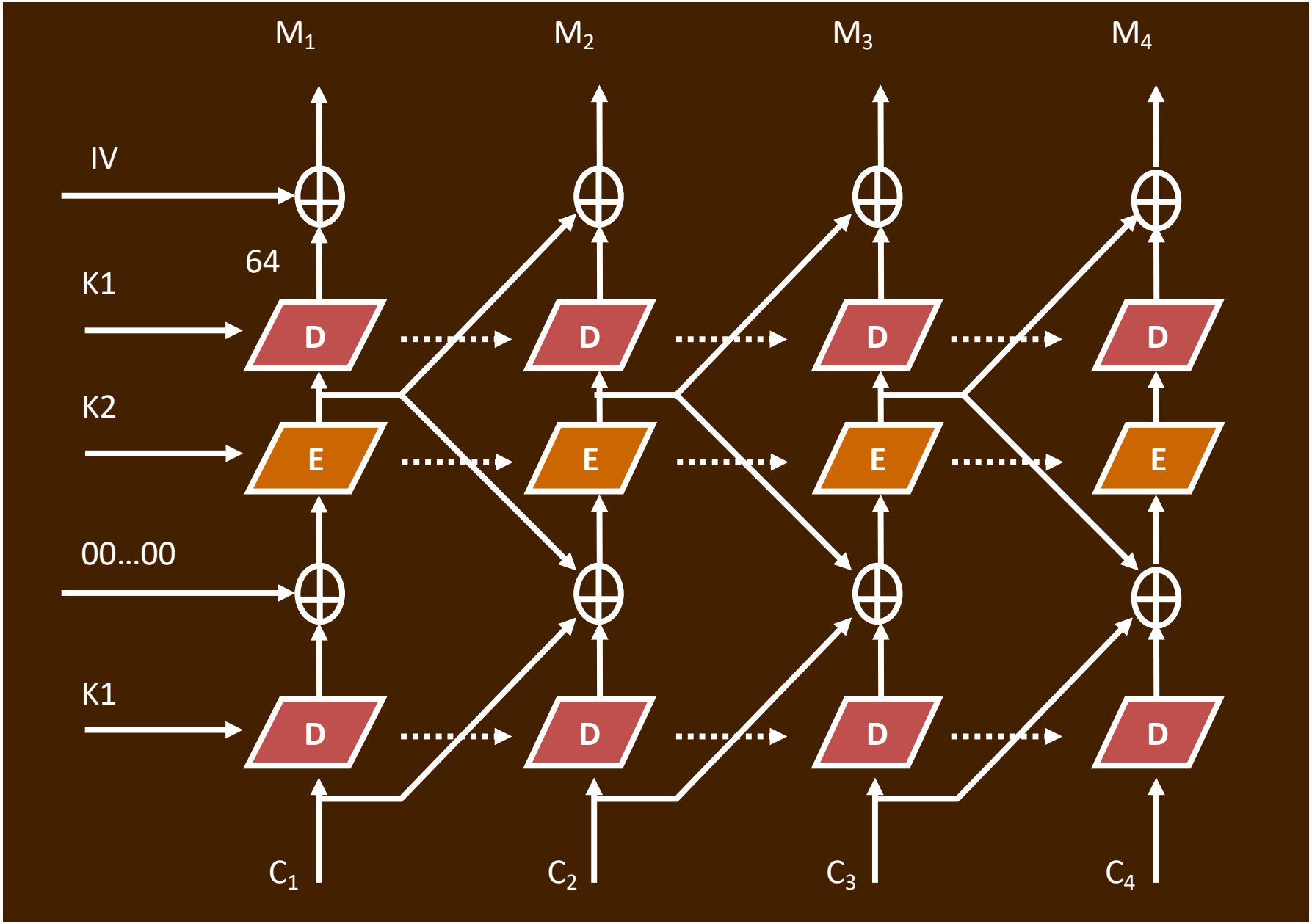
3DES-EDE: OCM Decryption



3DES-EDE: Inside Chaining Mode



3DES-EDE: ICM Decryption



More on Stream Cipher

Stream Ciphers

- Remember one-time pad?

  Ciphertext(Key,Message)=Message⊕Key
  - Key must be a random bit sequence as long as message
- Idea: replace "random" with "pseudo-random"
  - Encrypt with pseudo-random number generator (PRNG)
  - PRNG takes a short, truly random secret seed and expands it into a long "random-looking" sequence
    - E.g., 128-bit seed into a 10^6-bit pseudo-random sequence

No efficient algorithm can tell this sequence from truly random

- Ciphertext(Key,Msg)=IV, Msg⊕PRNG(IV,Key)
  - Message processed bit by bit, not in blocks

How Random is “Random?”

Properties of Stream Ciphers

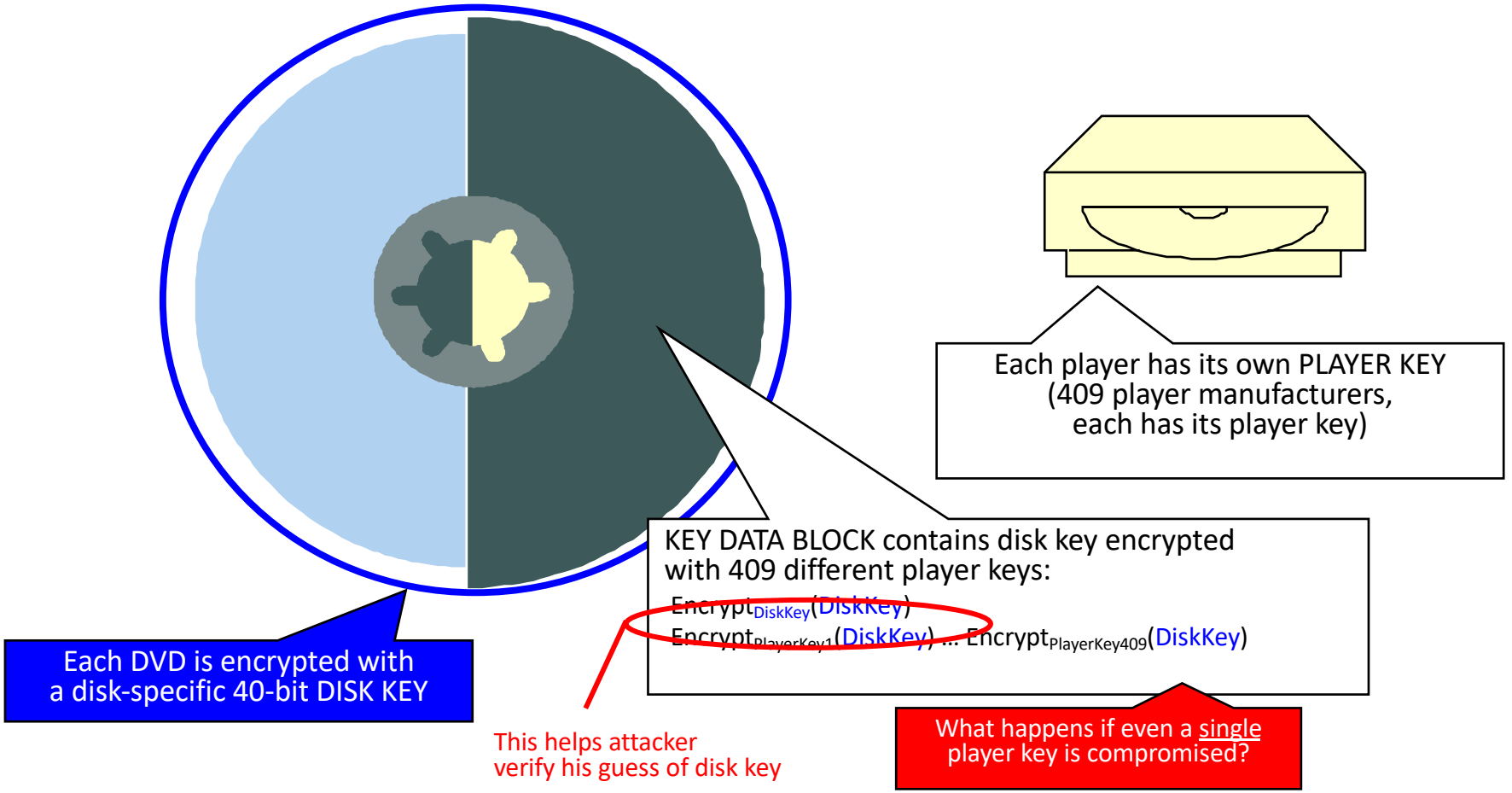
- Usually very fast (faster than block ciphers)
  - Used where speed is important: WiFi, DVD, RFID, VoIP
- Unlike one-time pad, stream ciphers do not provide perfect secrecy
  - Only as secure as the underlying PRNG
  - If used properly, can be as secure as block ciphers
- PRNG is, by definition, unpredictable
- Most widely used stream cipher: RC4
  - SSL/TLS for Web security, WEP for wireless

Weaknesses of Stream Ciphers

- No integrity
  - Associativity & commutativity: $(X \oplus Y) \oplus Z = (X \oplus Z) \oplus Y$
  - $(M_1 \oplus \text{PRNG(seed)}) \oplus M_2 = (M_1 \oplus M_2) \oplus \text{PRNG(seed)}$
- Known-plaintext attack is very dangerous if keystream is ever repeated
  - Self-cancellation property of XOR: $X \oplus X = 0$
  - $(M_1 \oplus \text{PRNG(seed)}) \oplus (M_2 \oplus \text{PRNG(seed)}) = M_1 \oplus M_2$
  - If attacker knows M_1, then easily recovers M_2
    - Most plaintexts contain enough redundancy that knowledge of M_1 or M_2 is not even necessary to recover both from $M_1 \oplus M_2$

Content Scrambling System (CSS)

- DVD encryption scheme from Matsushita and Toshiba



DeCSS

- In CSS, disk key is encrypted under hundreds of different player keys… including Xing, a software DVD player
- Reverse engineering the object code of Xing revealed its decryption key
  - Recall that every CSS disk contains the master disk key encrypted under Xing's key
  - One bad player $\Rightarrow$ entire system is broken!
- Easy-to-use DeCSS software

DeCSS Aftermath

- DVD CCA sued Jon Lech Johansen, one of DeCSS authors (eventually dropped)
- Publishing DeCSS code violates copyright
  - Underground distribution as haikus and T-shirts
  - “Court to address DeCSS T-Shirt: When can a T-shirt become a trade secret? When it tells you how to copy a DVD.” - From Wired News

Fundamental Weakness of CSS

- CSS utilizes a proprietary 40-bit stream cipher algorithm
  - Structural flaws in CSS reduce the effective key length to only around 16 bits
  - CSS can be compromised in less than a minute by brute-force with a 450 MHz processor
- Since CSS is broken, new stand is proposed: Advanced Access Content System (AACS)
  - Used in HD DVD and Blu-ray Disc (BD)
  - 128-bit AES (CBC)

Summary

- ECB mode is not secure
  - CBC most commonly used mode of operation
- Triple-DES (with 2 keys) is much stronger than DES
  - usually uses EDE in Outer Chaining Mode
- Stream cipher is simple, fast
  - Key size needs to be large enough
  - Other weakness