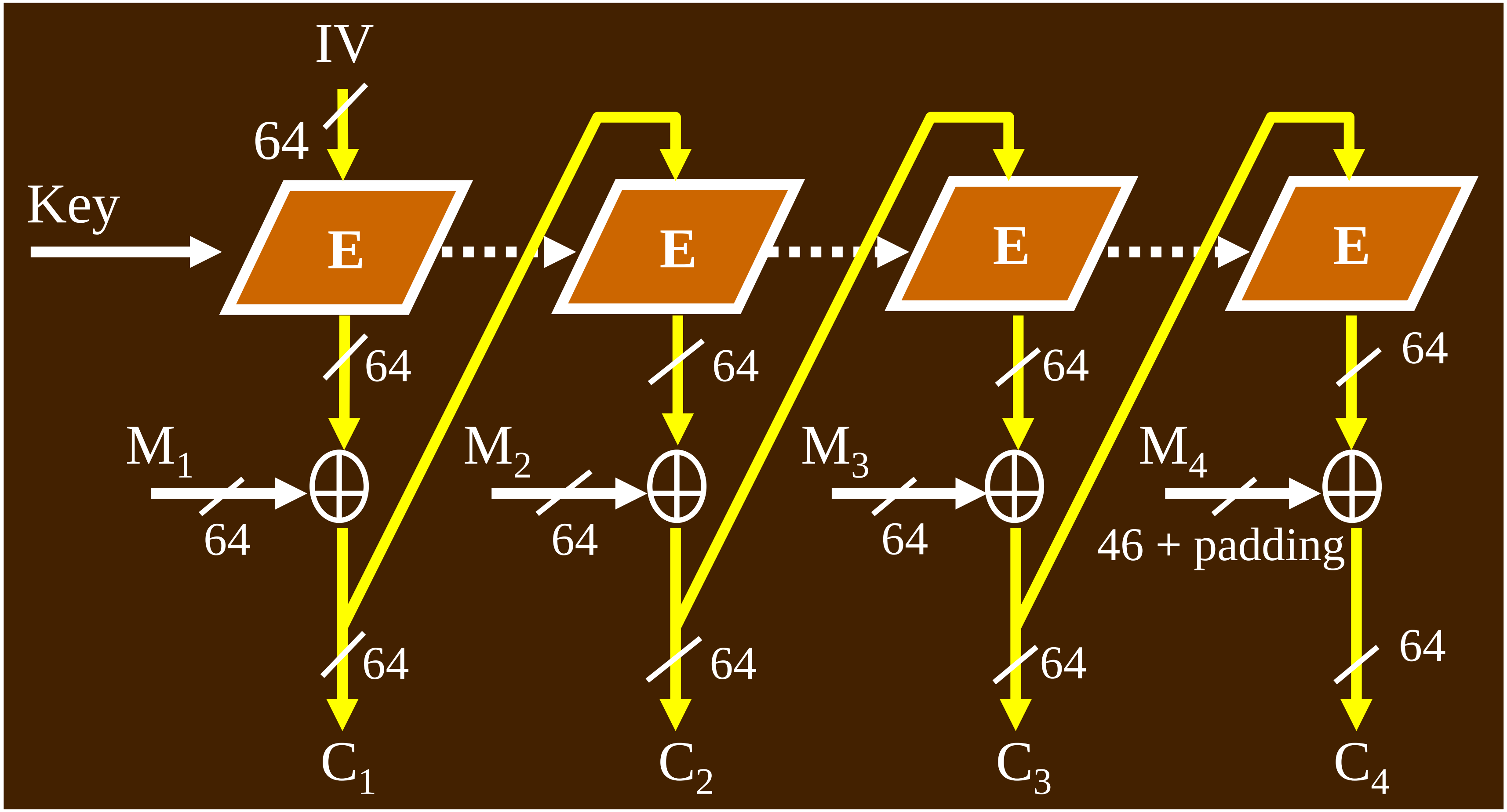
CIS 6930/4930 Computer and Network Security

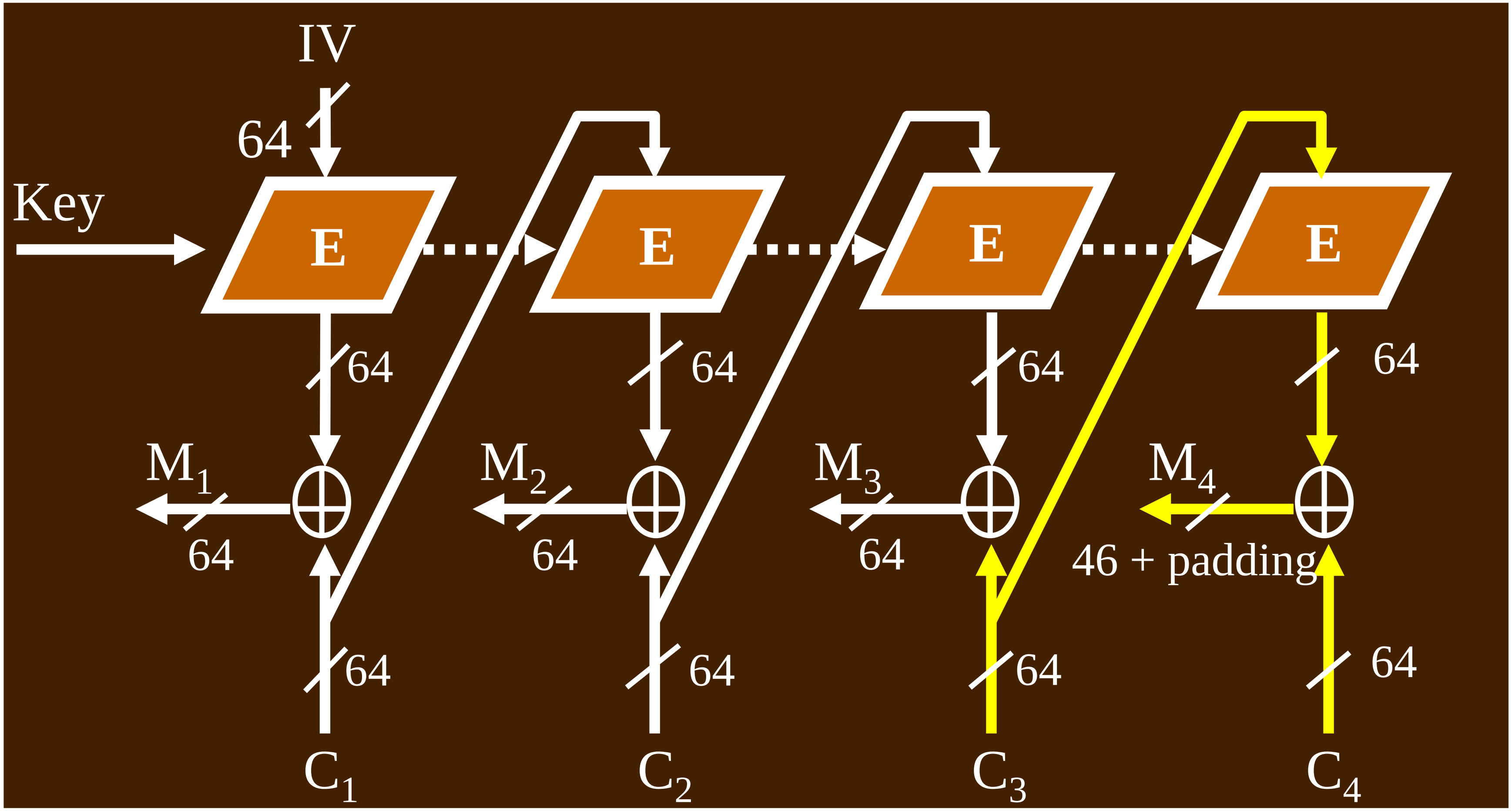
Topic 3.2 Secret Key Cryptography – Modes of Operation

Cipher Feedback Mode (CFB)



- Ciphertext block C_j depends on all preceding plaintext blocks

CFB Decryption

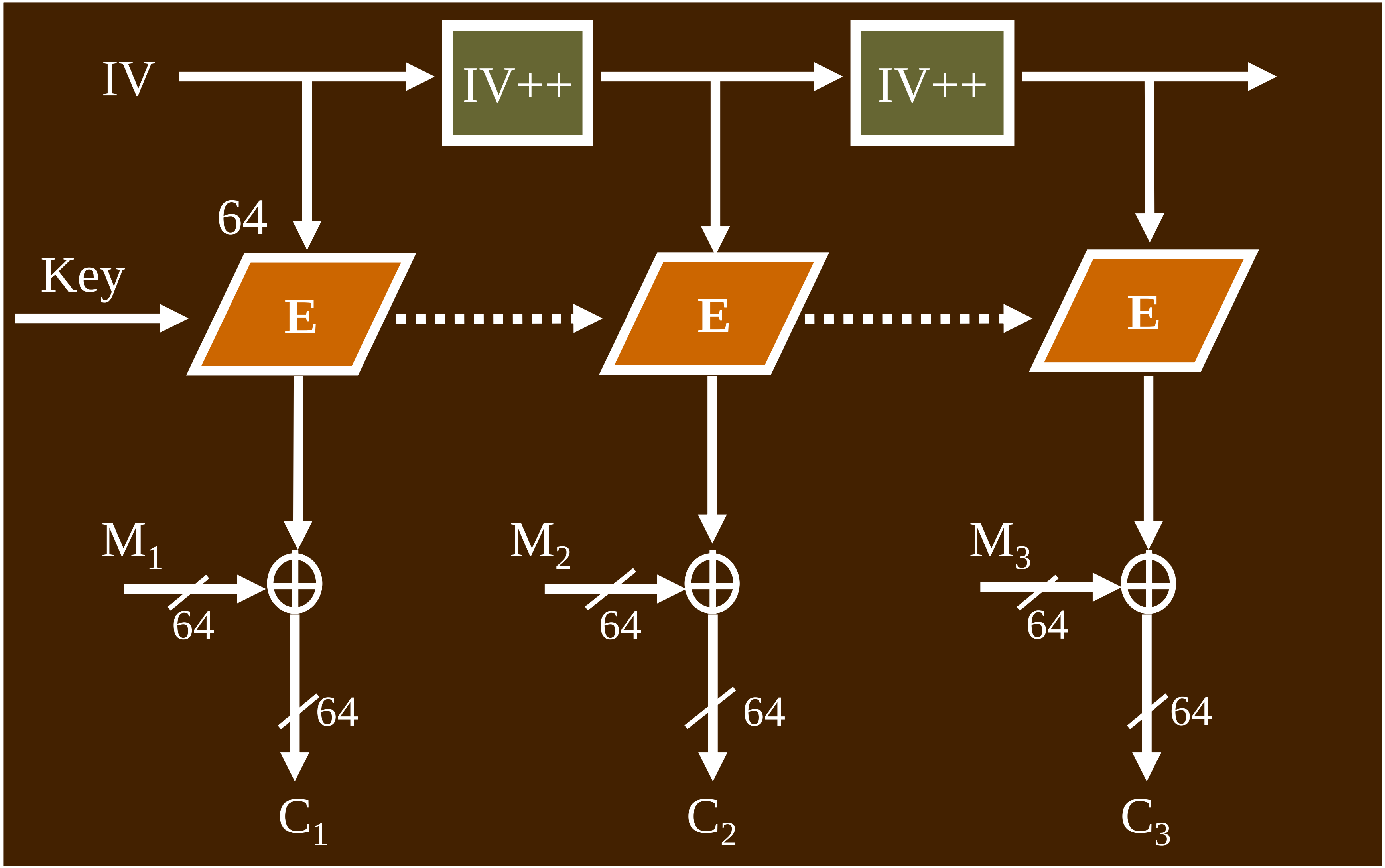


- No block decryption required!

CFB Properties

- Does information leak?
  - Identical plaintext blocks produce different ciphertext blocks
- Can ciphertext be manipulated predictably?
  - ???
- 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 predictably
  - ???
- Parallel processing possible
  - Yes (both generating pad and XORing)
- Do ciphertext errors propagate?
  - ???

CIS 6930/4930 Computer and Network Security

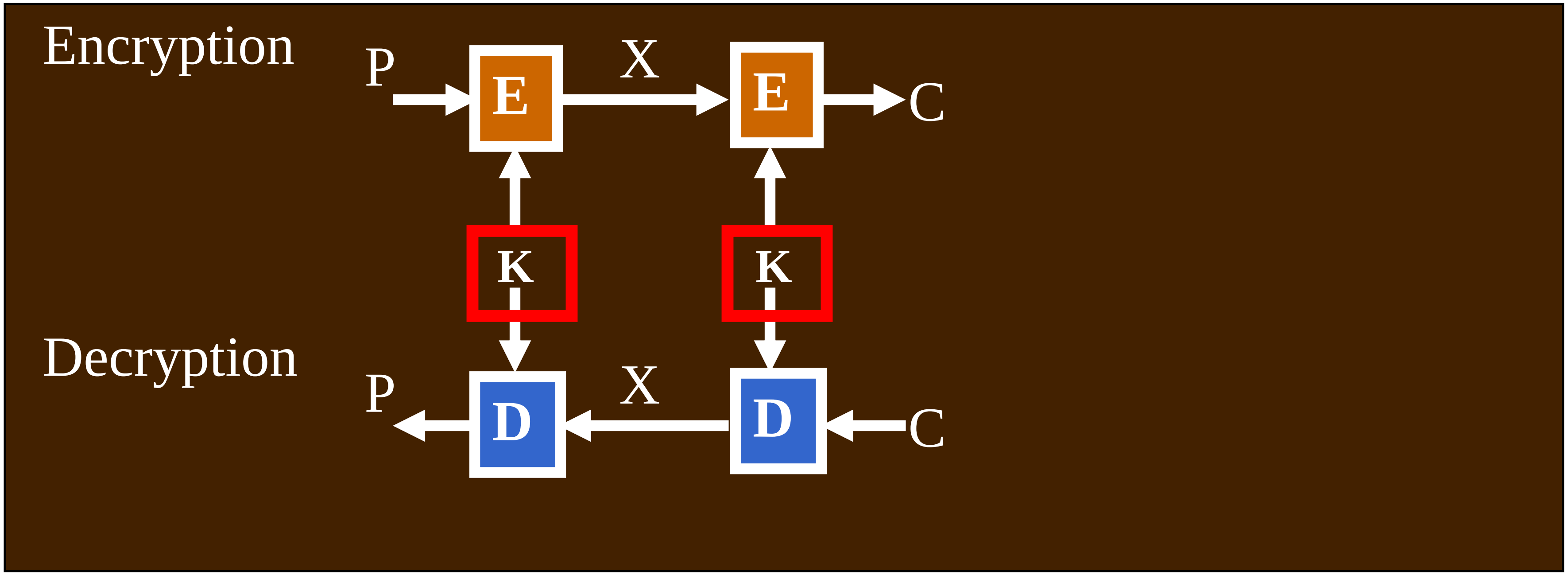
Topic 3.3 Secret Key Cryptography – Triple DES

Stronger DES

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

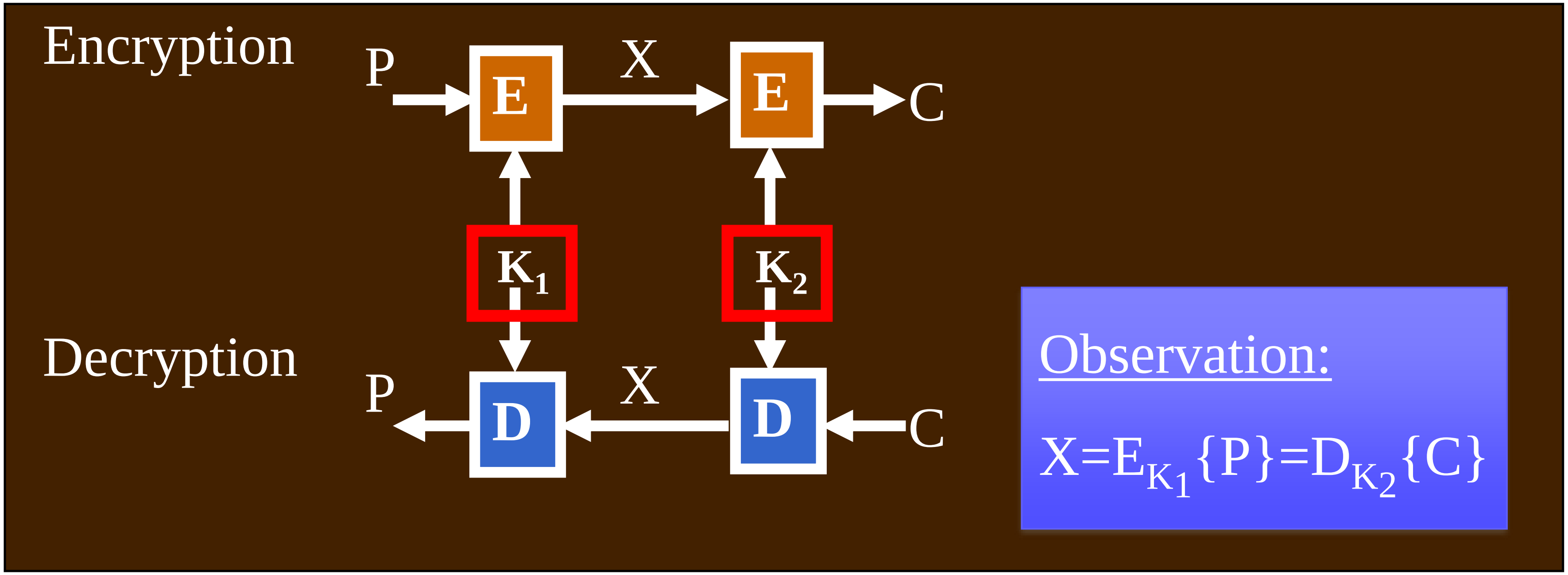
Double Encryption with DES

- Does encrypting using the same key make things more secure?



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?



The Meet-in-the-Middle Attack

1. Choose a plaintext P and generate ciphertext C, using double-DES with $\mathcal{K}1+\mathcal{K}2$
2. Then…
   a. encrypt P using single-DES for all possible 2^{56} values K_1 to generate all possible single-DES ciphertexts for P:
   $X_1, X_2, \ldots, X_{2^{56}}$;
   store these in a table indexed by ciphertex values
   b. decrypt C using single-DES for all possible 2^{56} values K_2 to generate all possible single-DES plaintexts for C:
   $Y_1, Y_2, \ldots, Y_{2^{56}}$;
   for each value, check the table

Steps … (Cont'd)

3. Meet-in-the-middle:
   - Each match ($X_i = Y_j$) reveals a *candidate key pair* K_i+K_j
   - There are 2^{112} pairs but there are only 2^{64} X's
4. On average, how many pairs have identical X and Y?
   - For any pair (X, Y), the probability that X = Y is $1/2^{64}$
   - There are 2^{112} pairs.
   - The expected number of pairs that result in identical X and Y is $2^{112} / 2^{64} = 2^{48}$

Steps … (Cont'd)

5. The attacker uses a second pair of plaintext and ciphertext to try the 2^{48} Key pairs

– There are 2^{48} key pairs and 2^{64} X's (Y's)

– The probability that a false key pair results in identical X and Y is $2^{48} / 2^{64} = 2^{-16}$

— The correct key pair always leads to identical X and Y

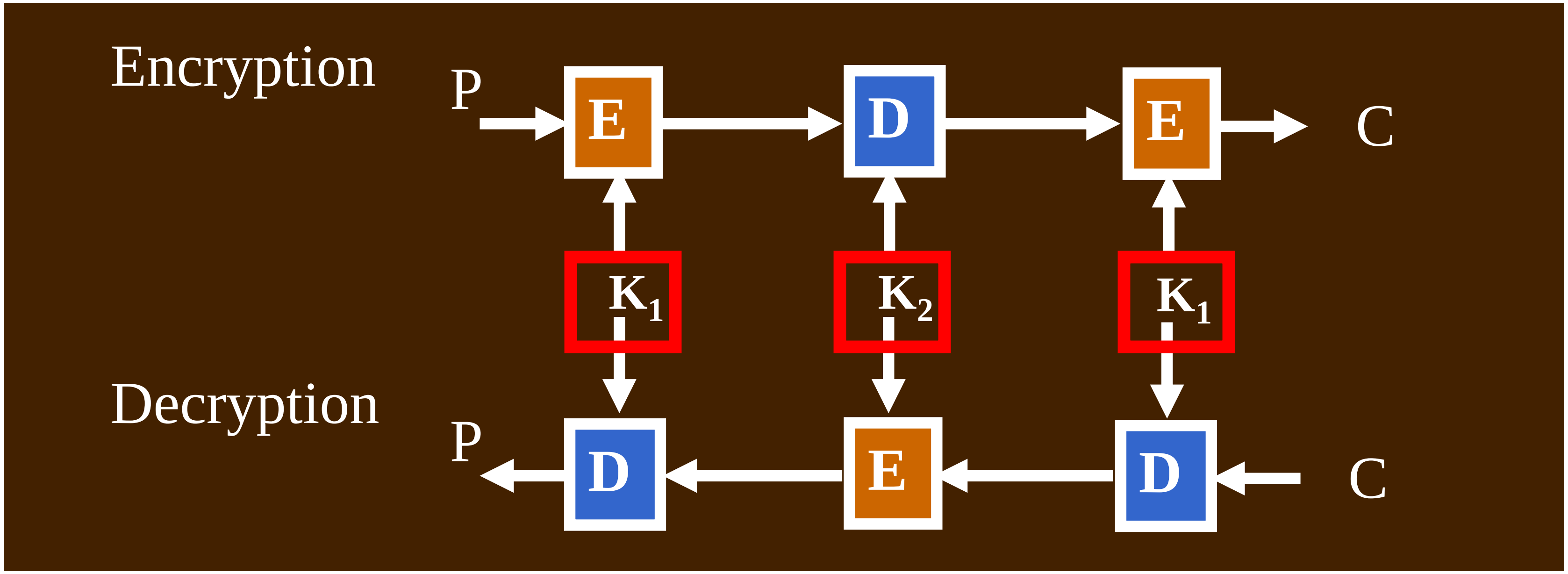
— A false key pair leads to identical X and Y at the probability of 2^{-16} (i.e., 1/65536)

— Hence, after examine two pairs of plaintext and ciphtertext, the attacker can normally identify the key

Attack Complexity

- How many DES encryptions and decryptions the attacker need to compute?
  - $2 \times 2^{56} + 2 \times 2^{48}$
- An expensive attack (computation + storage)
  - still, enough of a threat to discourage use of double-DES

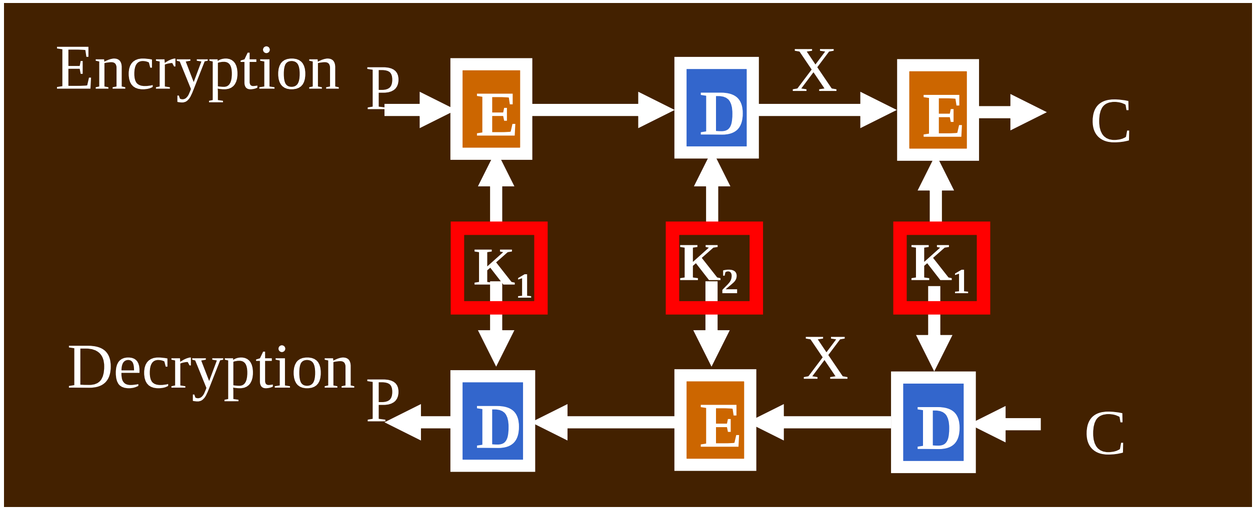
Triple Encryption (Triple DES-EDE)



- Apply DES encryption/decryption three times
  - why EDE?
  - One reason might be that by taking k1 = k2 = key, 3DES becomes single DES with key. 3DES can communicate with single DES.

Triple DES (Cont'd)

- Widely used
  - equivalent strength to using a 112 bit key
  - strength about 2^{112} against M-I-T-M attack



Observation:

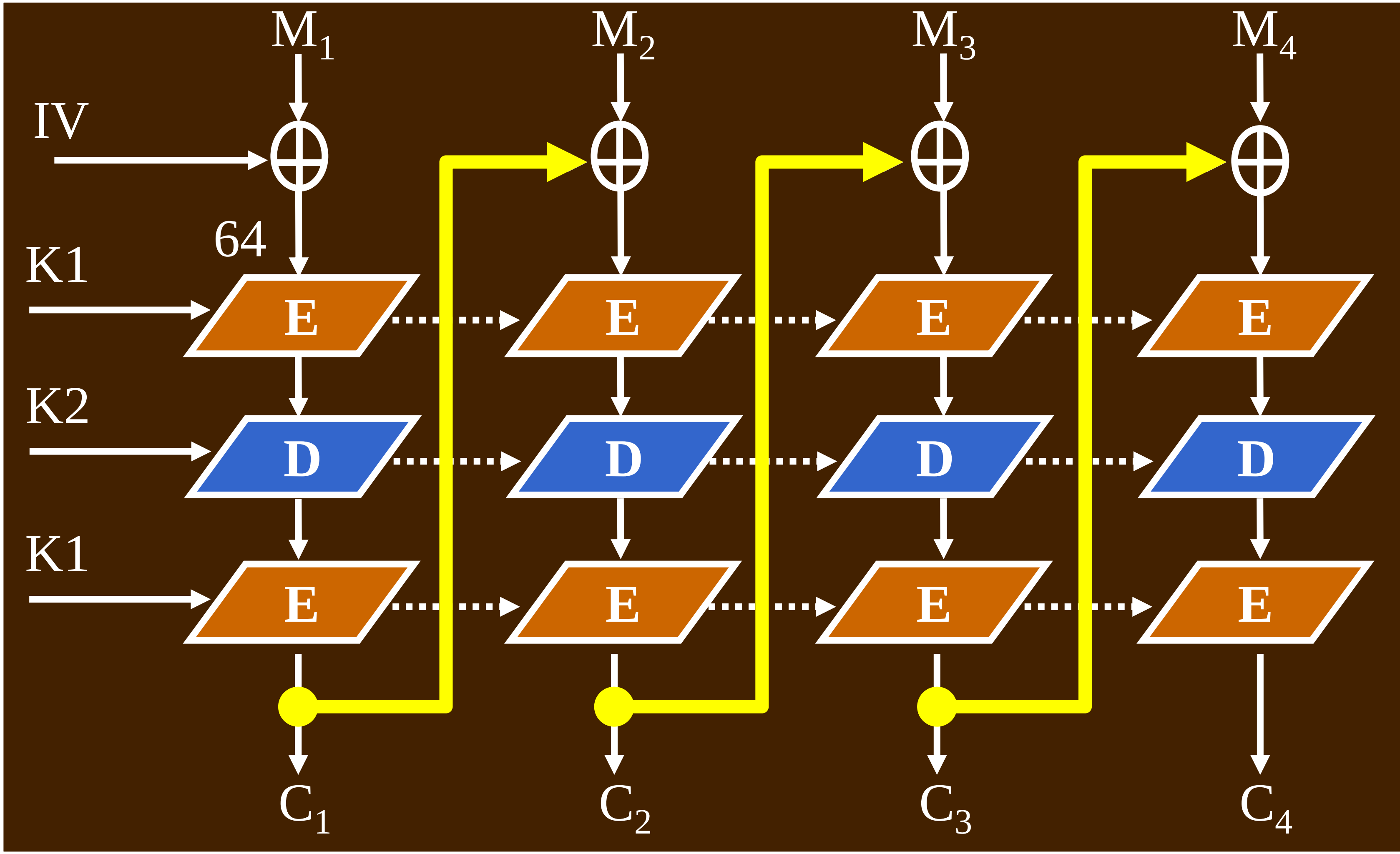
$X = D_{K_2}\{E_{K_1}\{P\}\} = D_{K_1}\{C\}$

Triple DES (Cont'd)

- 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

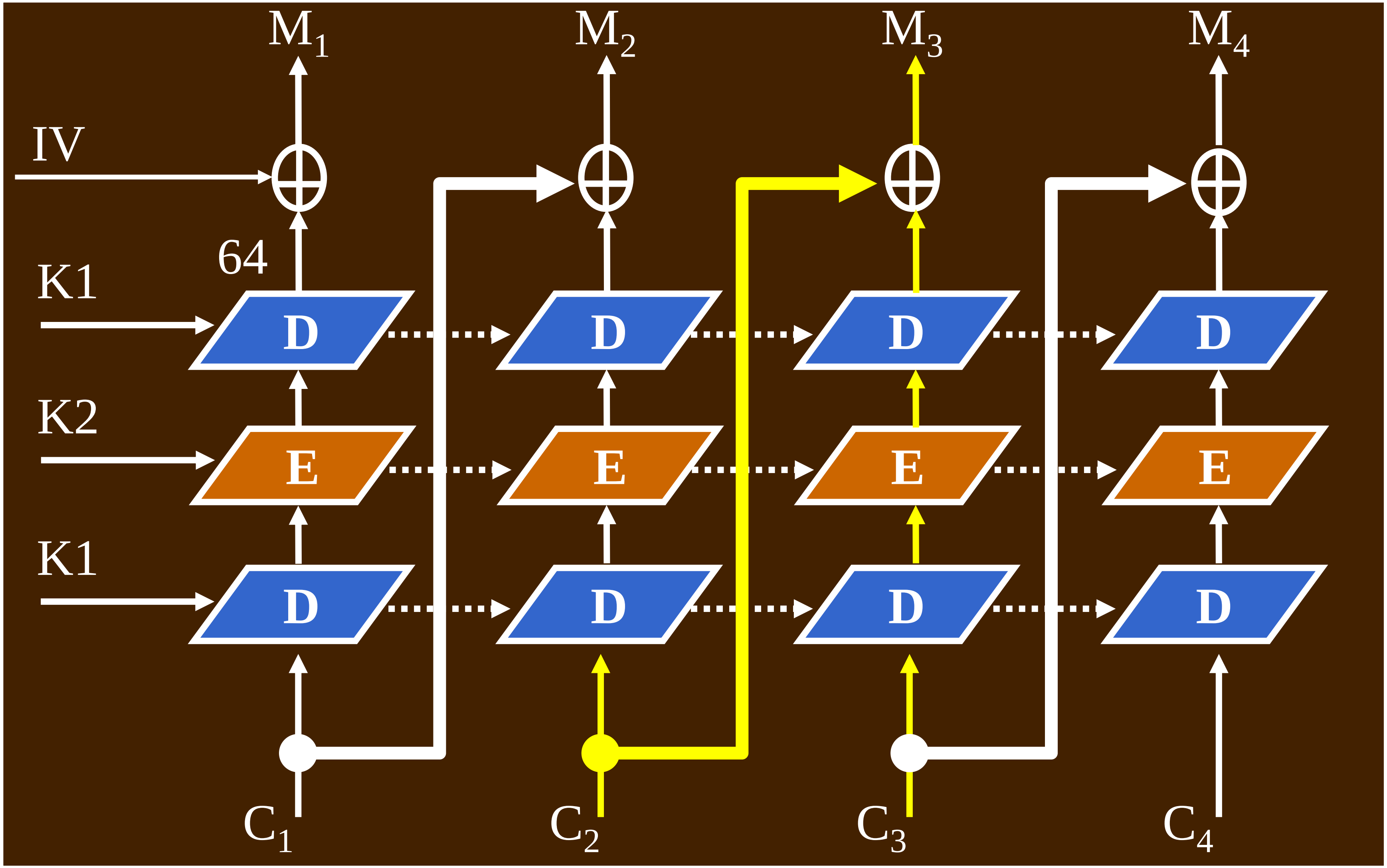
- Next question: how is block chaining used with triple-DES?

3DES-EDE: Outside Chaining Mode



- What basic chaining mode is this?

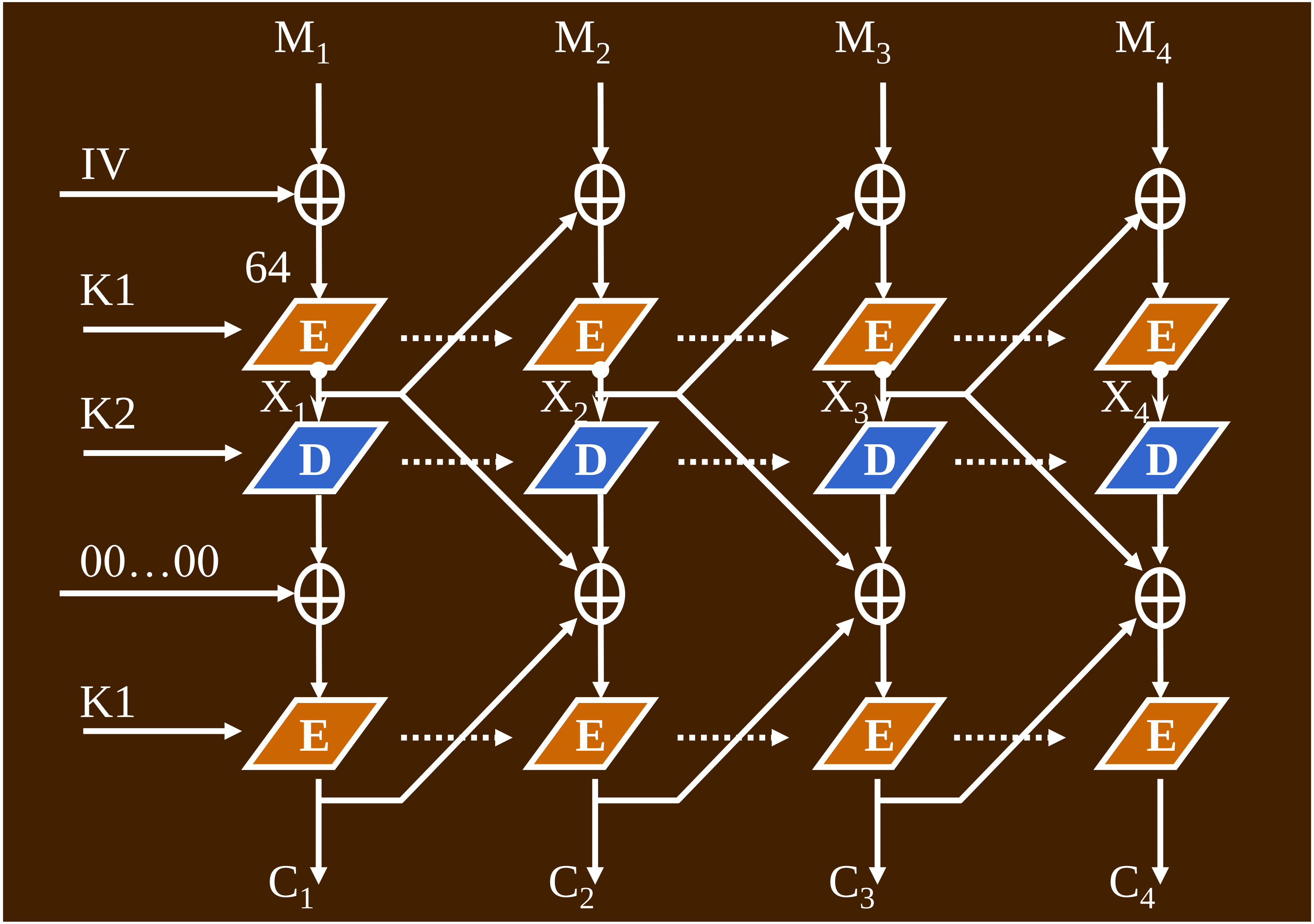
3DES-EDE: OCM Decryption



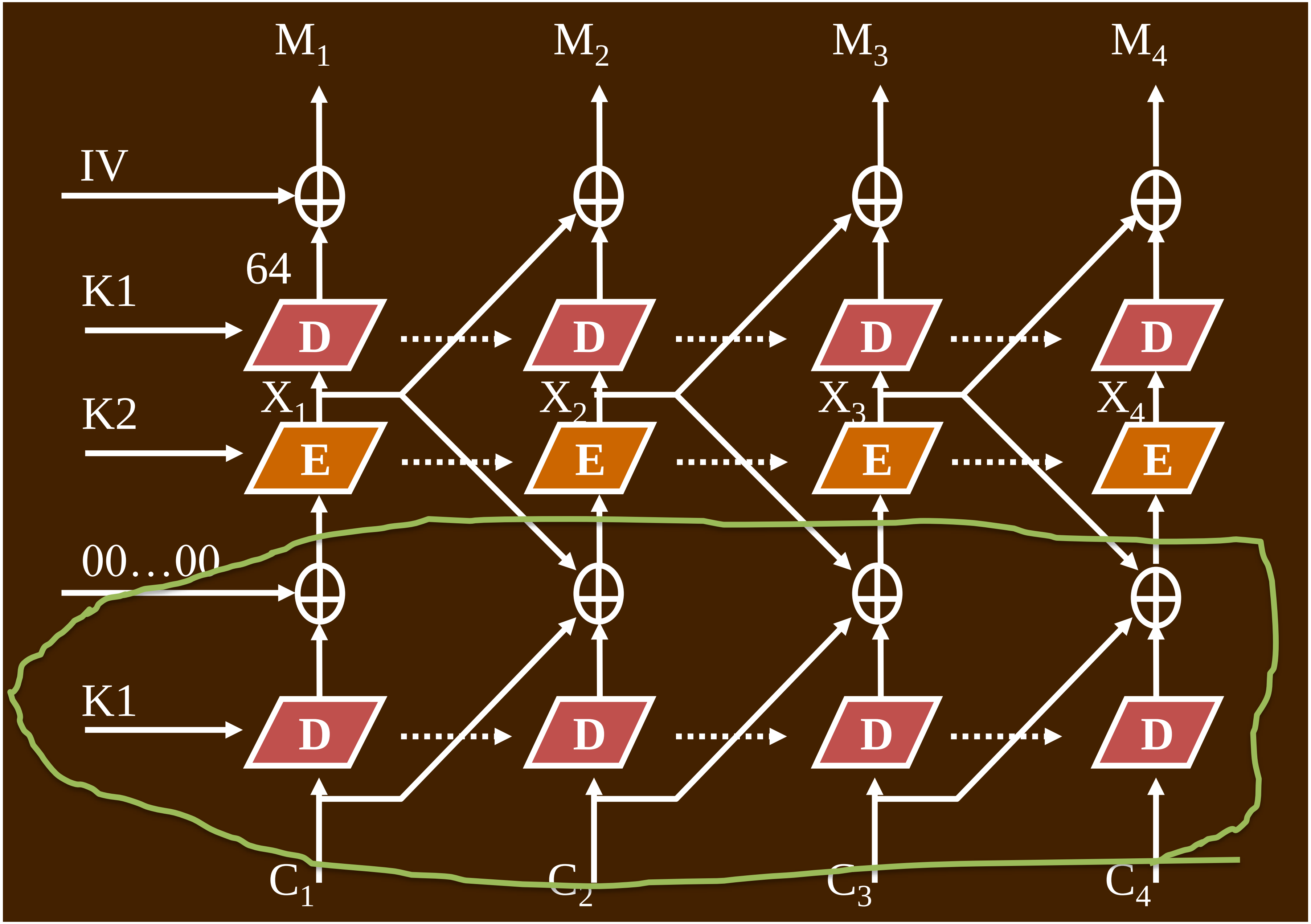
OCM Properties

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

3DES-EDE: Inside Chaining Mode



3DES-EDE: ICM Decryption



ICM Properties

- Does information leak?
  - identical plaintext blocks produce different ciphertext blocks
- Can ciphertext be manipulated predictably?
  - ???
- Parallel processing possible?
  - no (encryption), yes (partial of the decryption)
- Do ciphertext errors propagate?
  - ???

CIS 6930/4930 Computer and Network Security

Topic 3.4 Secret Key Cryptography – MAC with Secret Key Ciphers

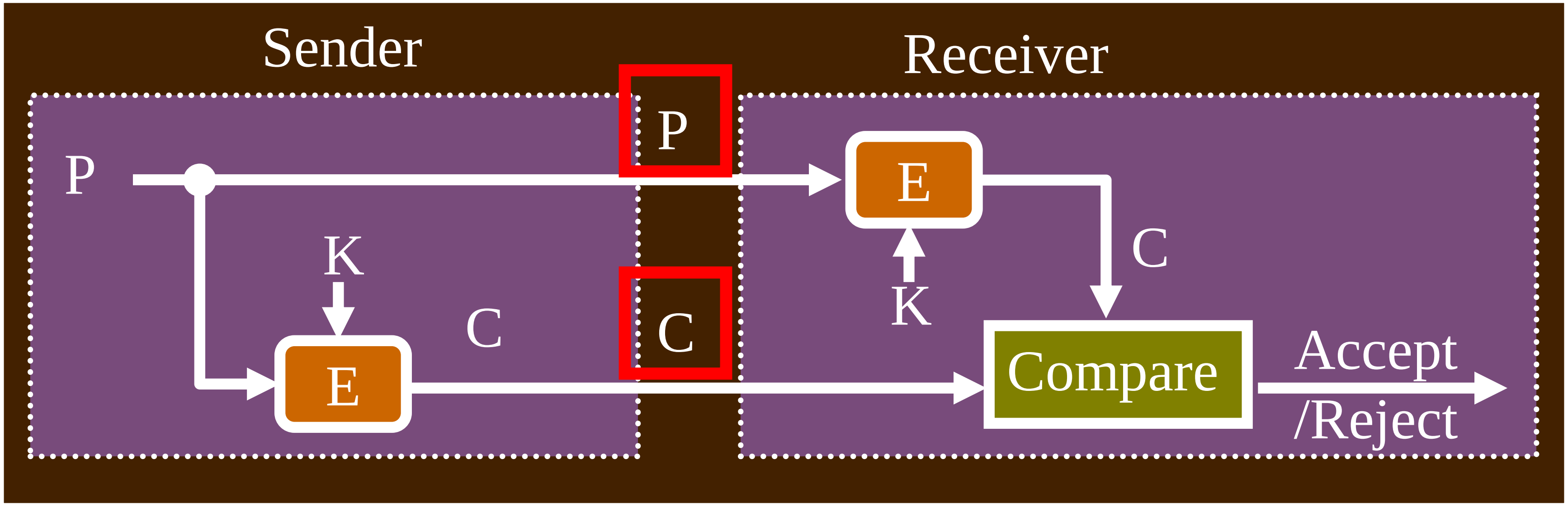
Message Authentication/Integrity

- Encryption easily provides confidentiality of messages
  - only the party sharing the key (the “key partner”) can decrypt the ciphertext
- How to use encryption to authenticate messages and verify the integrity? That is,
  - prove the message was created by the key partner
  - prove the message wasn’t modified by someone other than the key partner

Approach #1

- If the decrypted plaintext “looks plausible”, then conclude ciphertext was produced by the key partner
  - i.e., illegally modified ciphertext, or ciphertext encrypted with the wrong key, will probably decrypt to random-looking data
- But, is it easy to verify data is “plausible-looking”?

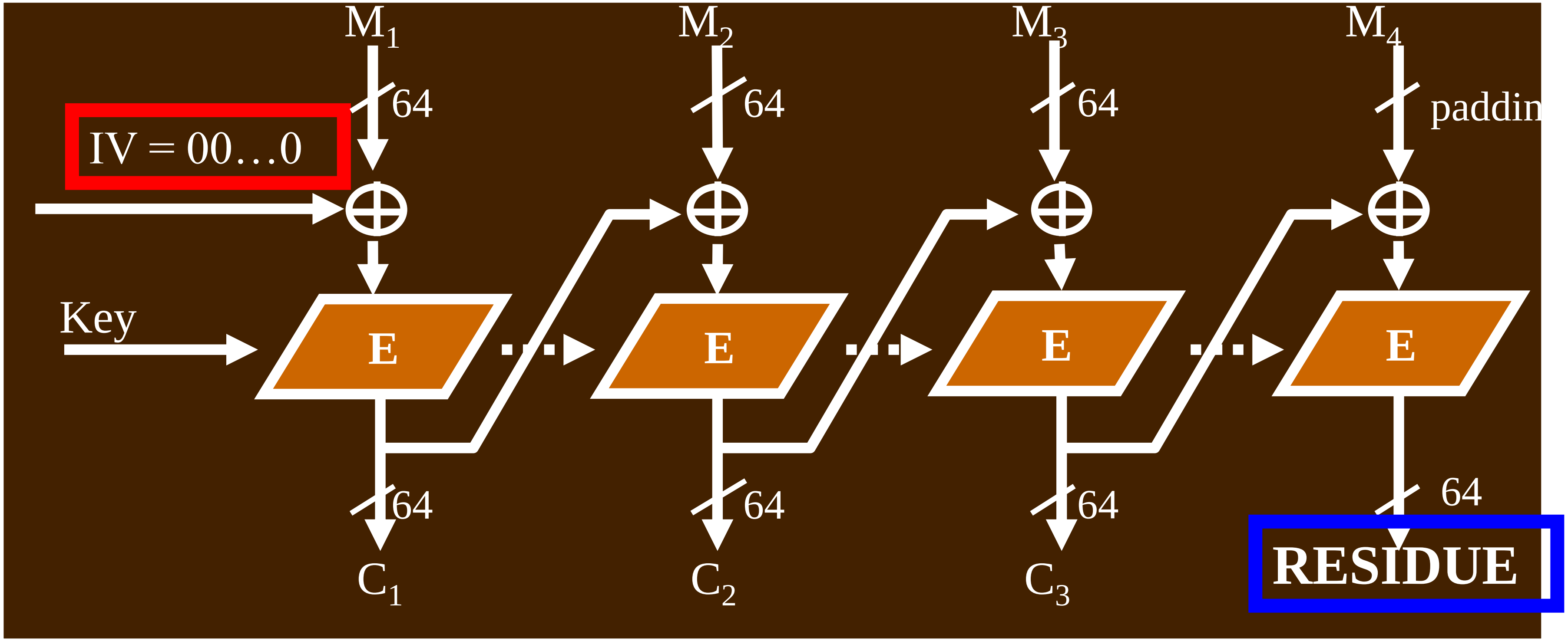
Approach #2: Plaintext+Ciphertext



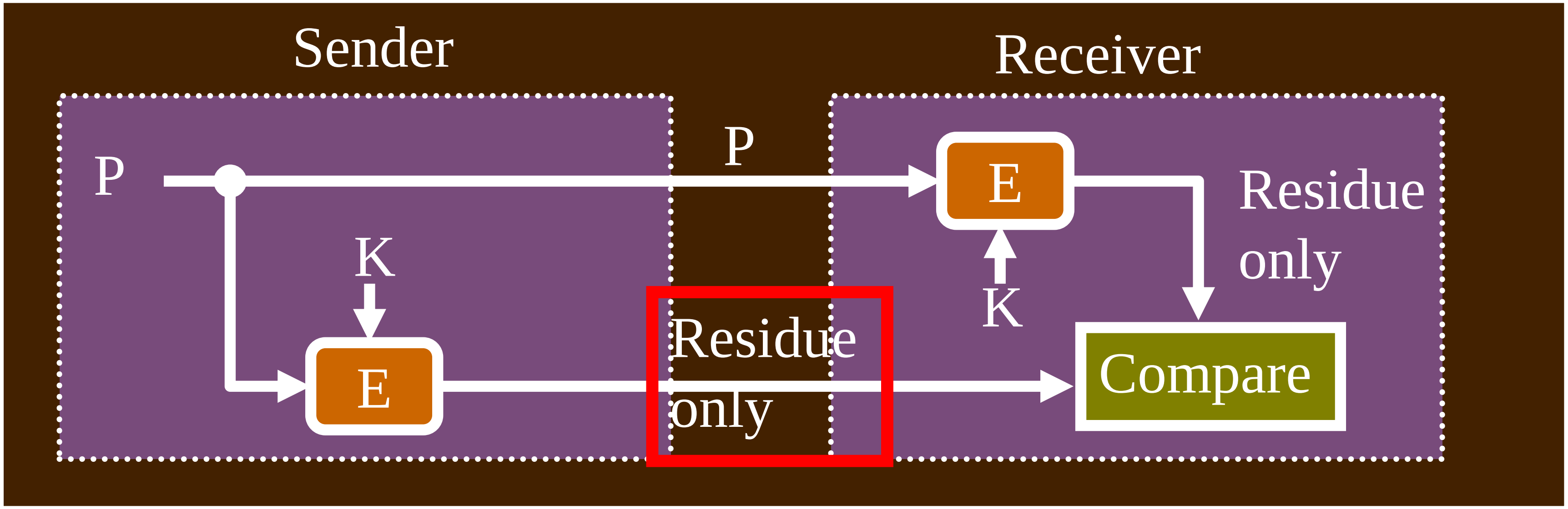
- Send plaintext and ciphertext
  - receiver encrypts plaintext, and compares result with received ciphertext
  - forgeries / modifications easily detected
  - any problems / drawbacks?

Approach #3: Use Residue

- Encrypt plaintext using DES CBC mode, with IV set to zero
  - the last (final) ciphertext output block is called the *residue*



Approach #3… (Cont'd)



- Transmit the plaintext and this residue
  - receiver computes same residue, compares to the received residue
  - forgeries / modifications highly likely to be detected

Message Authentication Codes

- *MAC*: a small fixed-size block (i.e., independent of message size) generated from a message using secret key cryptography
  - also known as *cryptographic checksum*

Requirements for MAC

1. Given M and MAC(M), it should be computationally infeasible (expensive) to construct (or find) another message M’ such that MAC(M’) = MAC(M)
2. MAC(M) should be uniformly distributed in terms of M
   – for randomly chosen messages M and M’, P(MAC(M)=MAC(M’)) = 2^{-k}, where *k* is the number of bits in the MAC

Requirements … (cont'd)

3. Knowing MAC(M), it should be computationally infeasible for an attacker to find M.

S.K. Crypto for Confidentiality AND Authenticity?

- So far we've got
  – confidentiality (encryption),

  or…

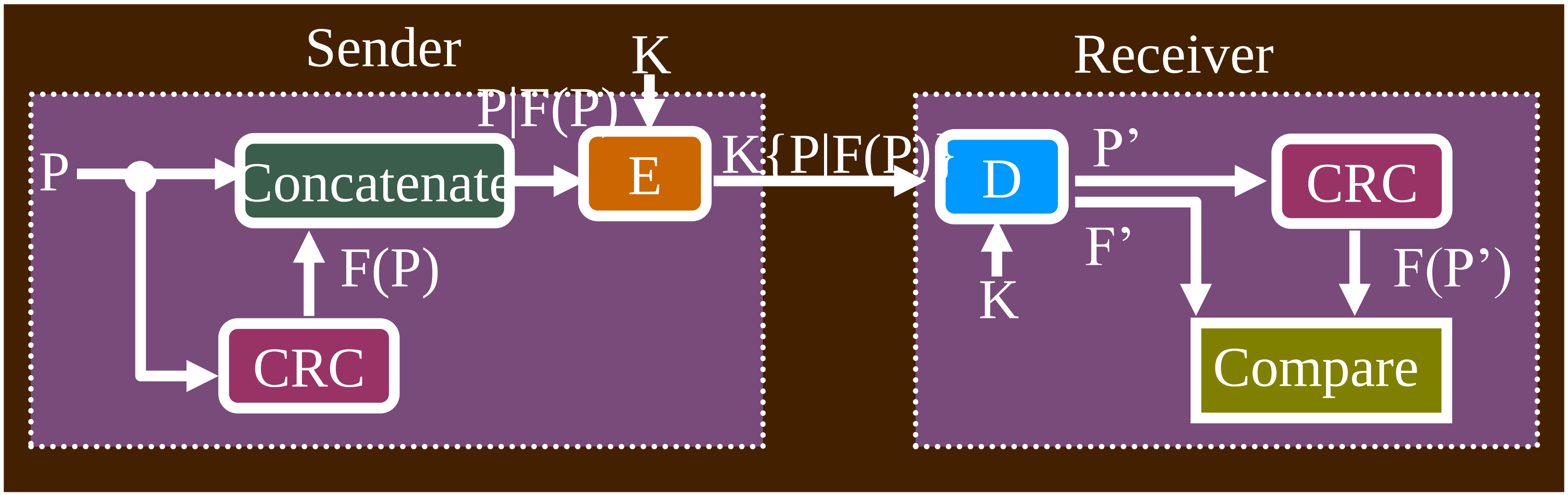
  – authenticity (MACs)
- Can we get both at the same time with one cryptographic operation?

Attempt #1

1. Sender computes an error-detection code F(P) of the plaintext P
2. Sender concatenates P and F(P) and encrypts
   - i.e., $C = E_K(P \mid F(P))$
3. Receiver decrypts received ciphertext C' using K, to get P'|F'
4. Receiver computes F(P') and compares to F' to authenticate received message P' = P

- How does this authenticate P?

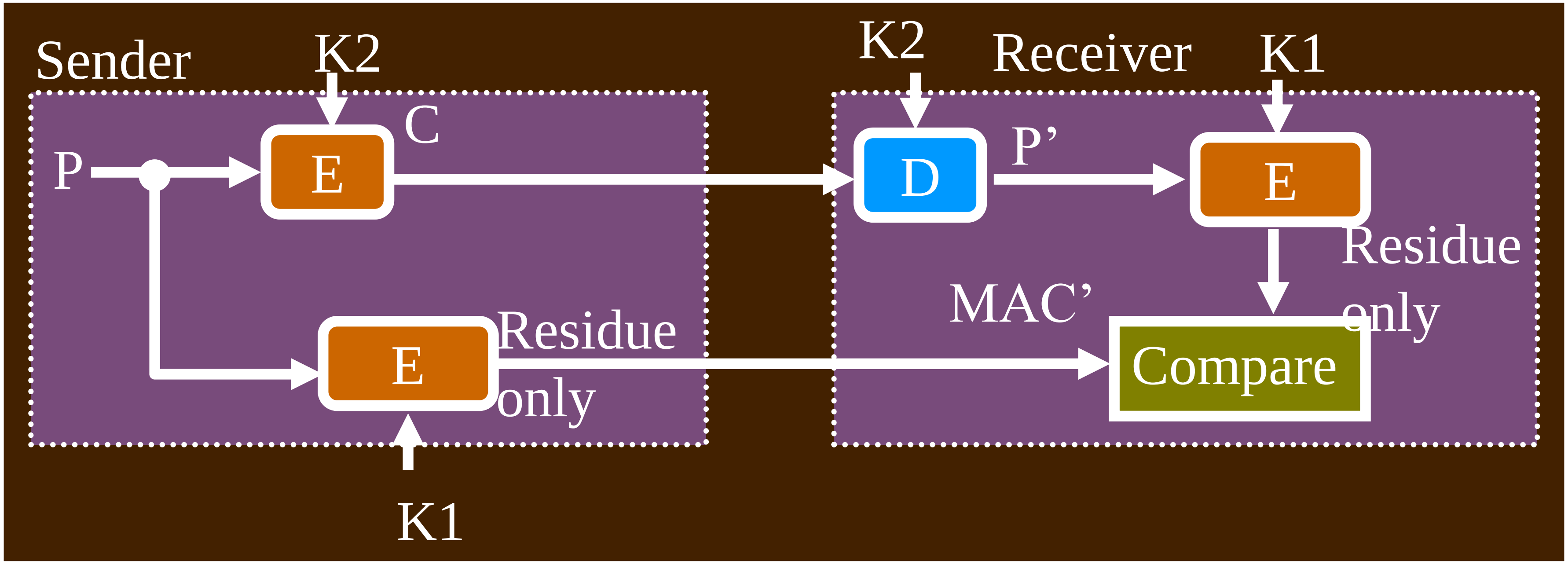
Attempt #1… (Cont'd)



Attempt #2

1. Compute residue (MAC) using key K1
2. Encrypt plaintext message M using key K2 to produce C
3. Transmit MAC | C to receiver
4. Receiver decrypts received C’ with K2 to get P’
5. Receiver computes MAC(P’) using K1, compares to received MAC’

Attempt #2… (cont'd)



- Good (cryptographic) quality, but…
- Expensive! Two separate, full encryptions with different keys are required

Summary

1. ECB mode is not secure
   – CBC most commonly used mode of operation
2. Triple-DES (with 2 keys) is much stronger than DES
   – usually uses EDE in Outer Chaining Mode
3. MACs use crypto to authenticate messages at a small cost of additional storage / bandwidth
   – but at a high computational cost