CIS 4360: Computer Security Fundamentals

Authenticated Encryption

Viet Tung Hoang

Agenda

1. AE and Its Security Definitions

2. Failed Ways To Build AE

3. Generic Compositions

Authenticated Encryption

K

Transfer $5 to account 12345

K

K

Privacy **Authenticity**

Encryption scheme

Authenticated Encryption
Achieve **both** of these aims

MAC

Authenticated Encryption (AE)

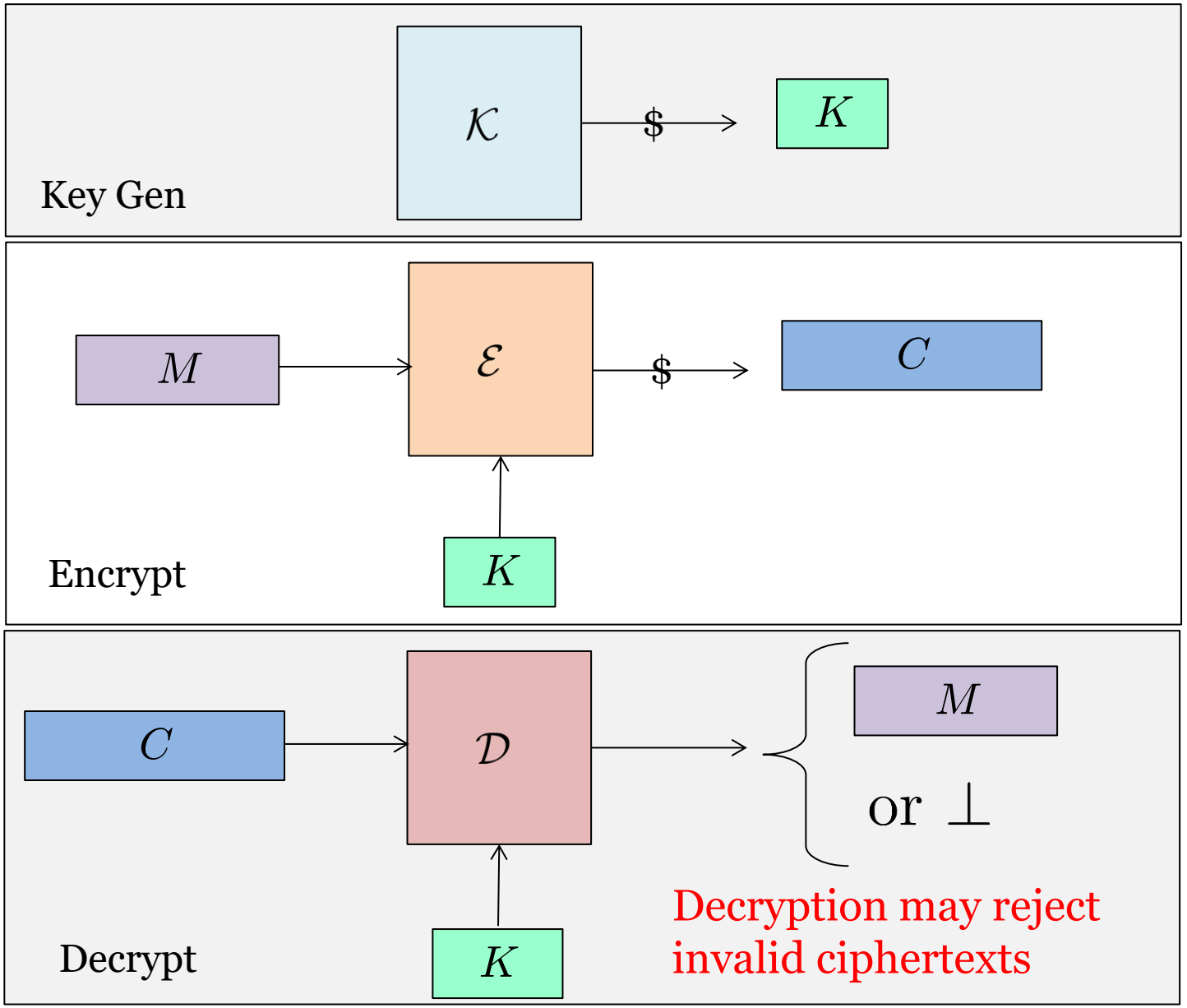
Emerged ~ 2000

Begin with two **realizations**

1. Authenticity is routinely needed/assumed
2. "Standard" privacy mechanisms don't provide it

Provide an easier-to-correctly-use abstraction boundary

AE Syntax


$\mathcal{K}$
$
K
Key Gen
M
$\mathcal{E}$
$
C
K
Encrypt
C
$\mathcal{D}$
M
or $\perp$
Decryption may reject invalid ciphertexts
K
Decrypt

Defining Security for AE

Authenticity

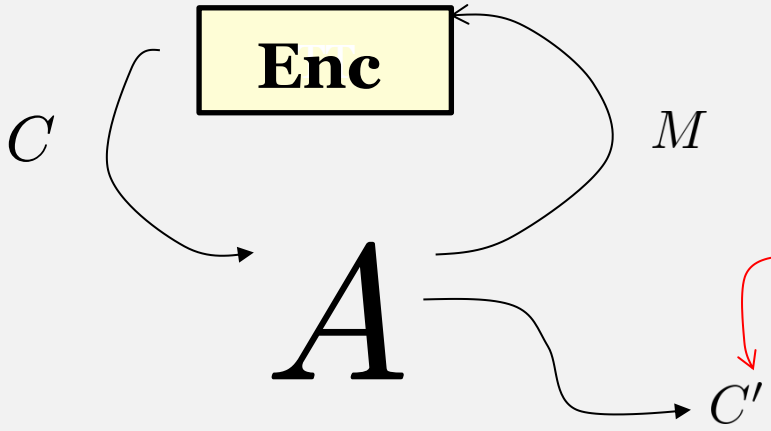
-Use Left-or-Right security for privacy

Auth $_{\mathcal{E}}$

procedure Initialize()
$K \xleftarrow{\$} \mathcal{K}$

procedure Enc(M)
Return $\mathcal{E}_K(M)$

procedure Finalize(C')
Return $(\mathcal{D}_K(C') \neq \perp)$



Must never receive from Enc

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathcal{T}}(A) = \Pr[\mathrm{Auth}^{A}_{\mathcal{E}} \Rightarrow 1]$$

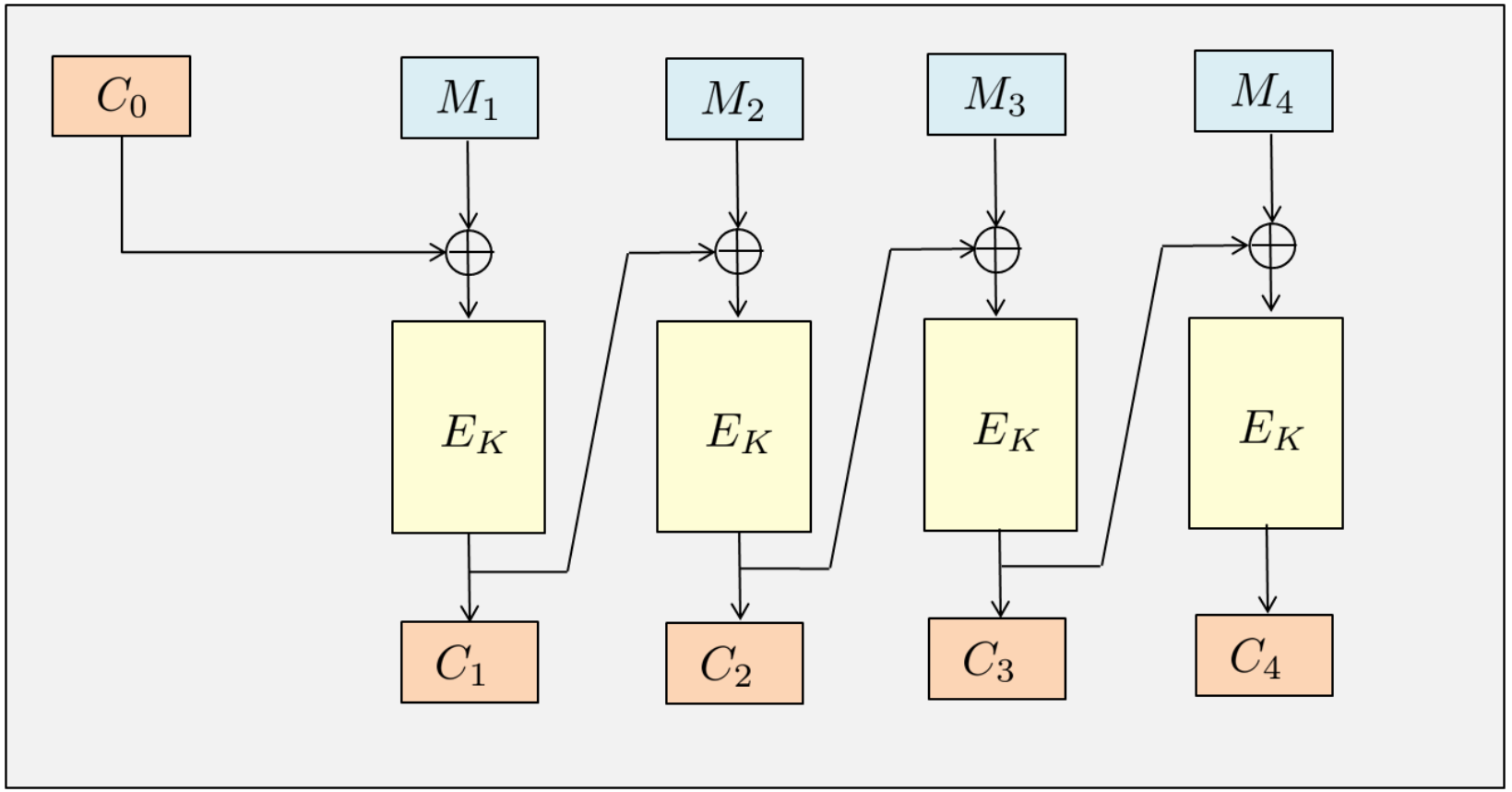
Agenda

1. AE and Its Security Definitions

2. Failed Ways To Build AE

3. Generic Compositions

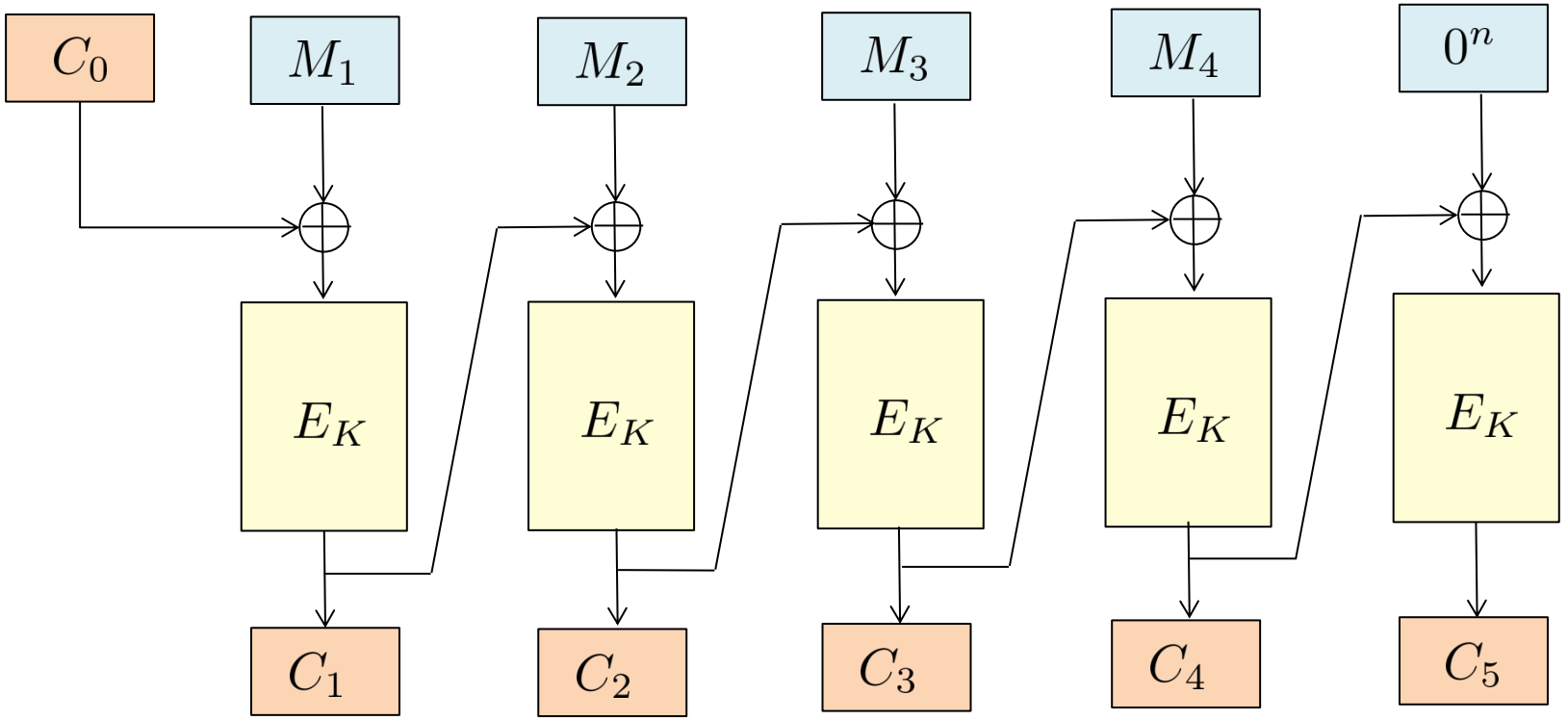
Plain Encryption Doesn't Provide Authenticity



Question: Does CBC provide authenticity?

Answer: No, because any ciphertext has valid decryption

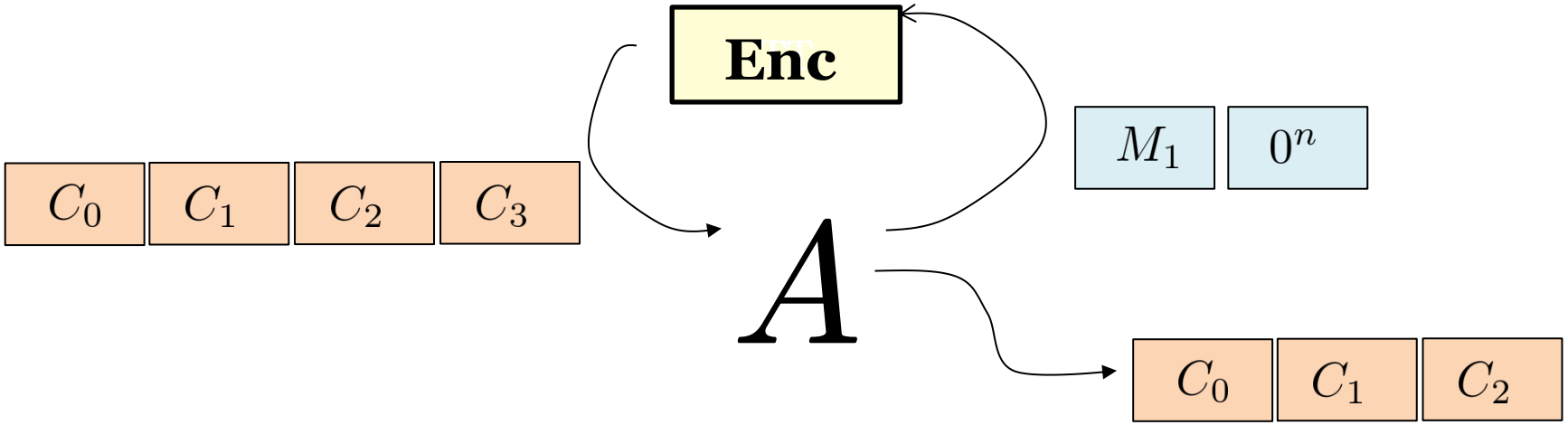
A Bad Fix: CBC with Redundancy

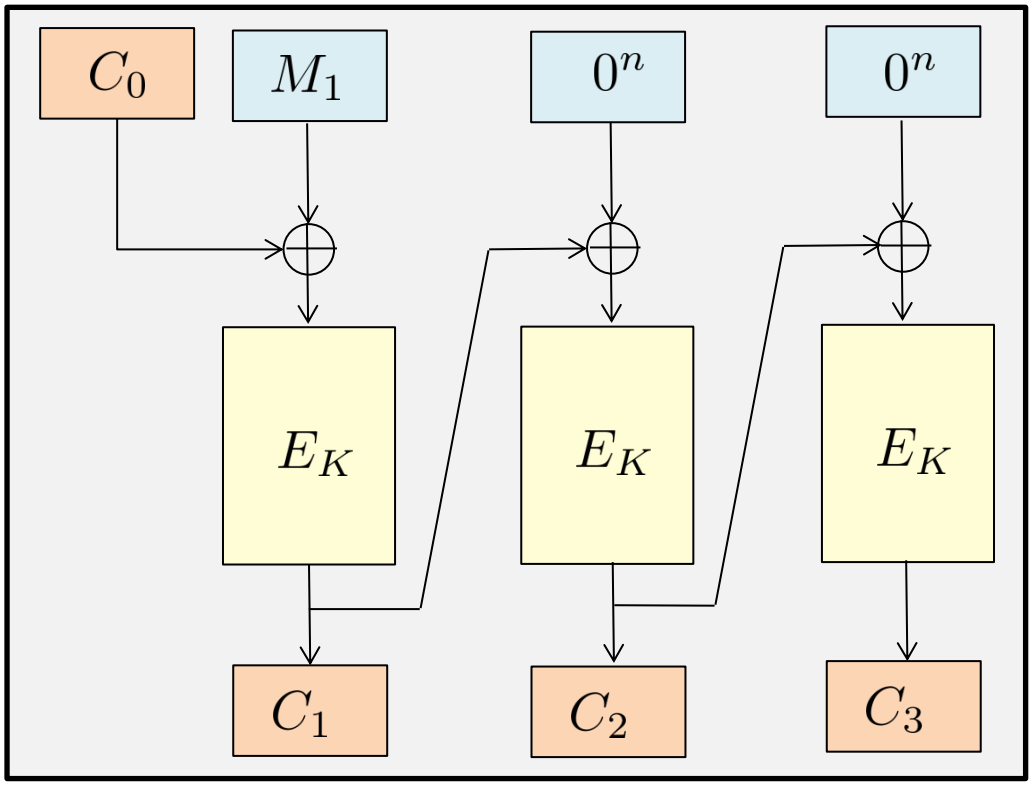


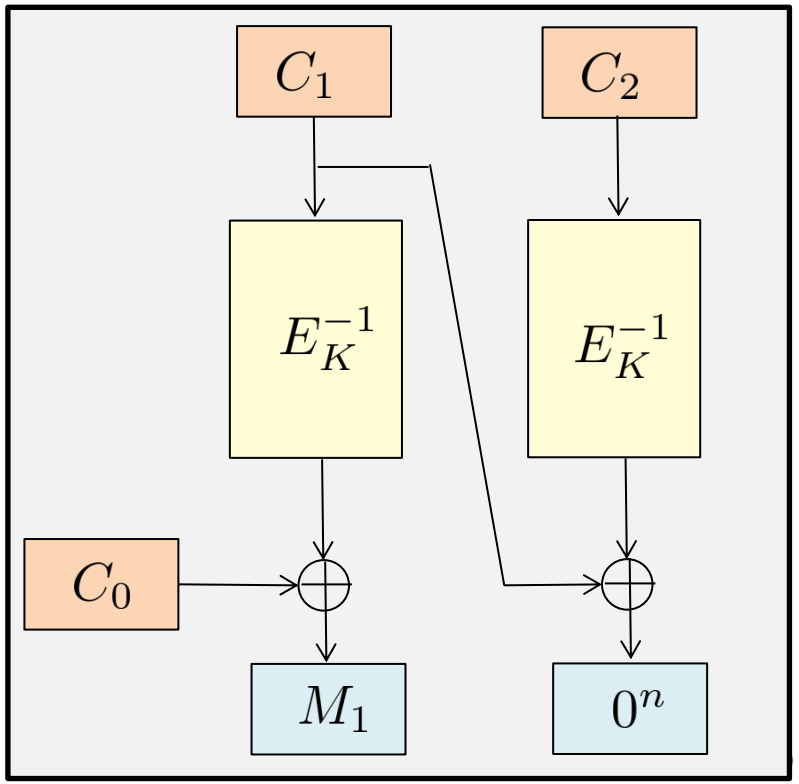
On decryption, verify the decrypted last block is zero.

Question: Break the authenticity of this scheme with a single Enc query

An Attack

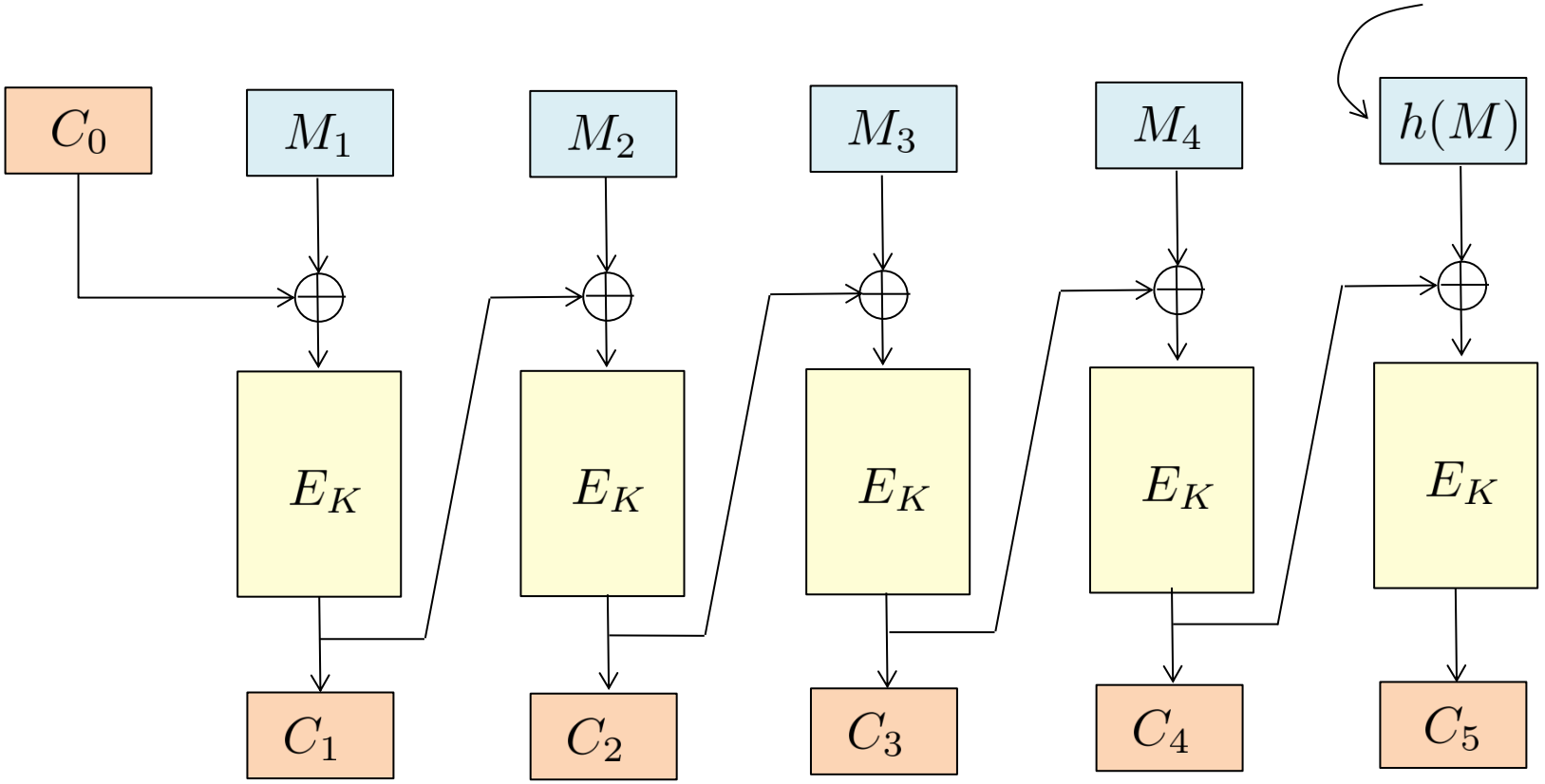

Enc
C_0 C_1 C_2 C_3
M_1 0^n
A
C_0 C_1 C_2


C_0 M_1 0^n 0^n
E_K E_K E_K
C_1 C_2 C_3


C_1 C_2
E_K^{-1} E_K^{-1}
C_0
M_1 0^n

Complex Redundancy Doesn't Help

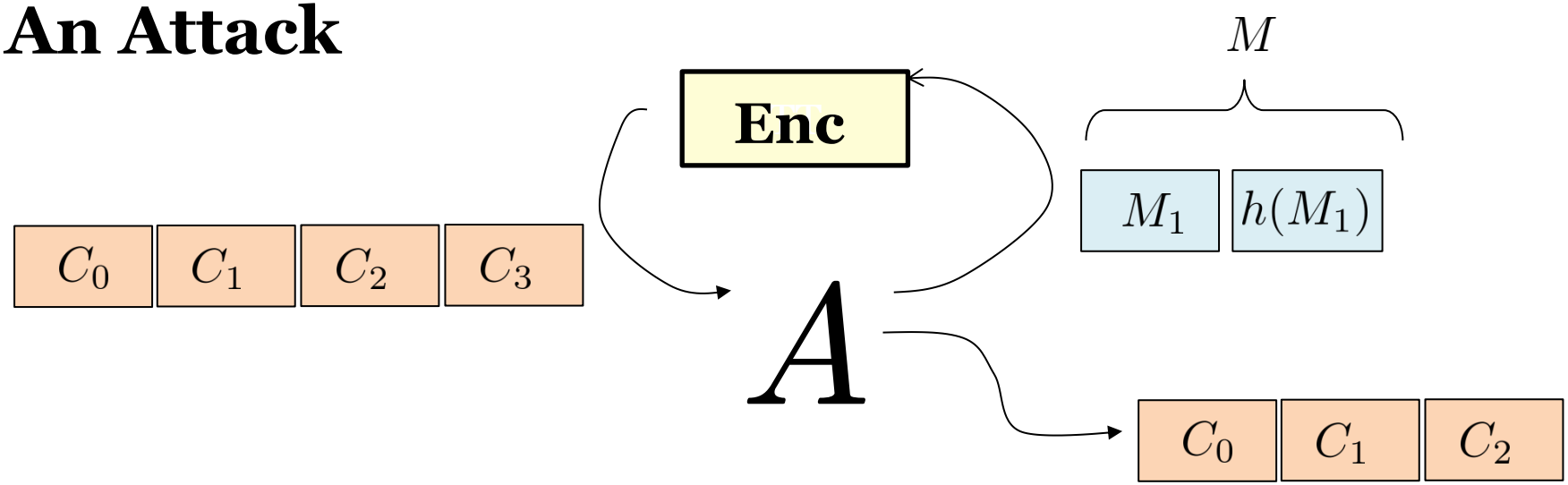
Some (unkeyed) "redundancy" function, such as checksum

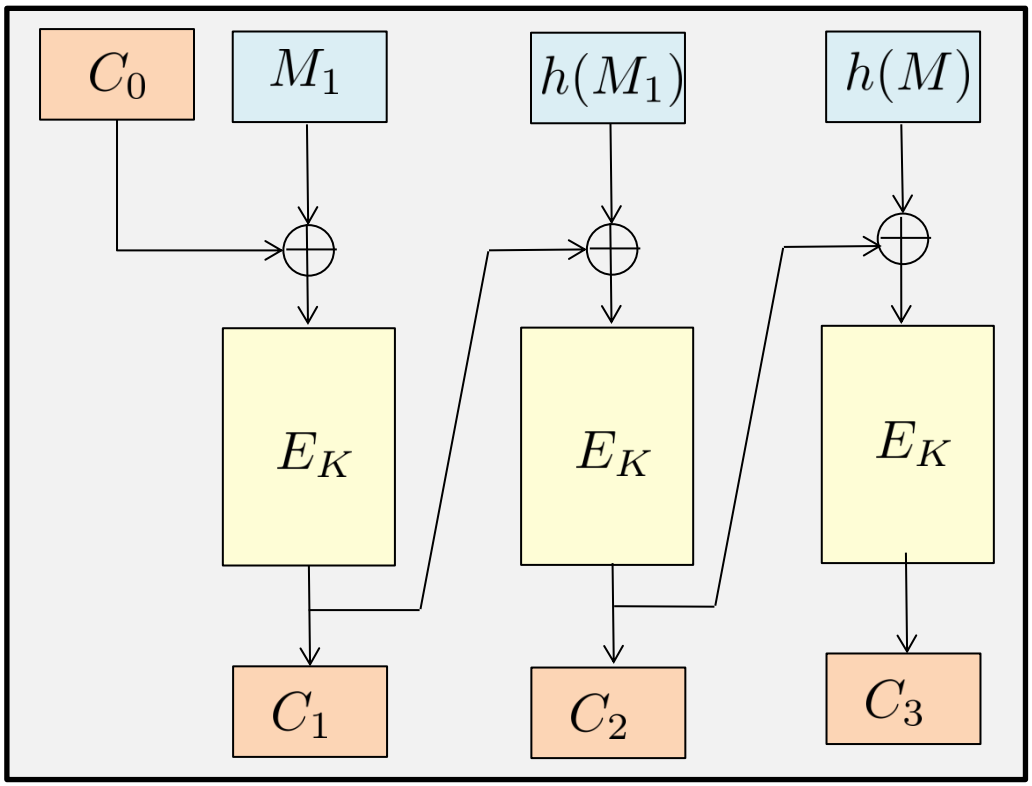


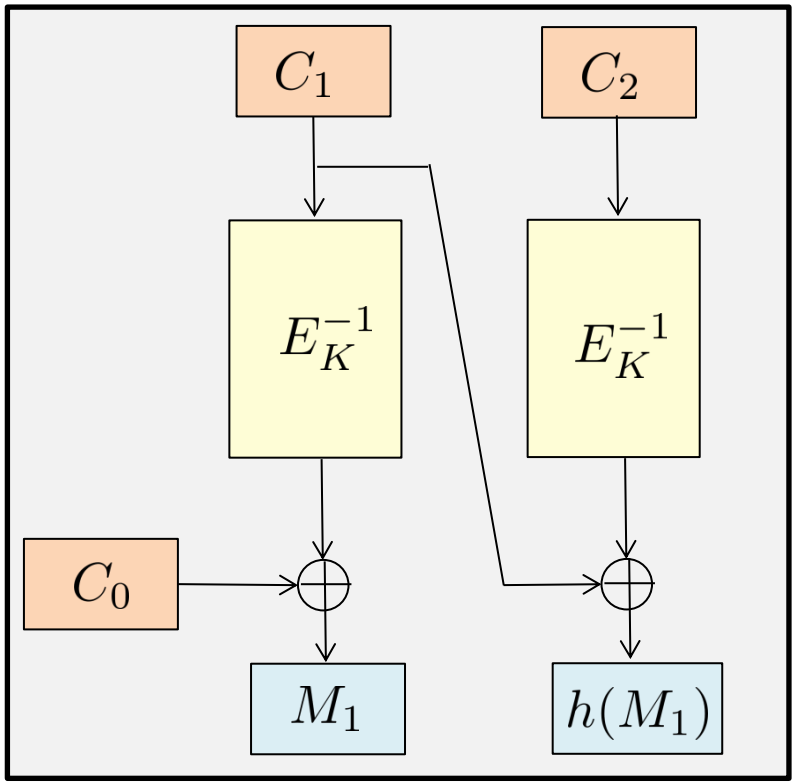
The redundancy is verified upon decryption

Question: Break the authenticity of this scheme with a single Enc query

An Attack


M
Enc
M_1
$h(M_1)$
C_0
C_1
C_2
C_3
A
C_0
C_1
C_2

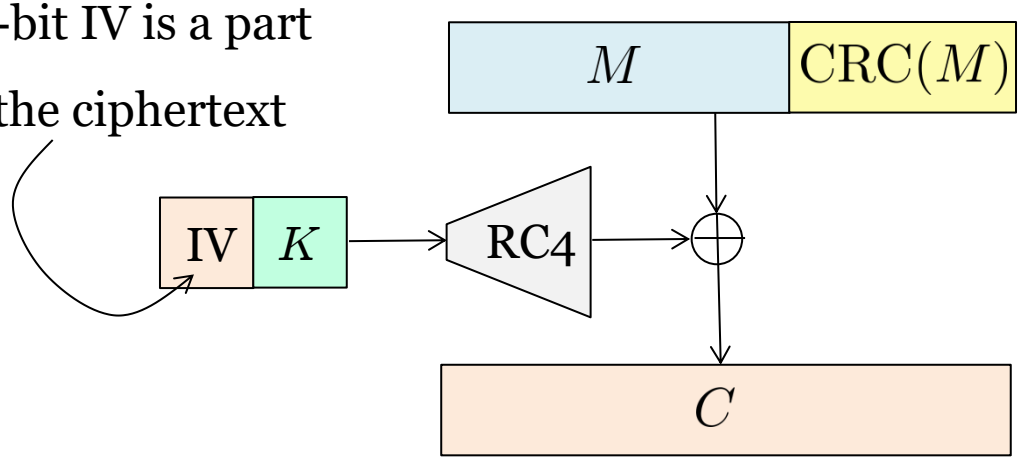

C_0
M_1
$h(M_1)$
$h(M)$
E_K
E_K
E_K
C_1
C_2
C_3


C_1
C_2
E_K^{-1}
E_K^{-1}
C_0
M_1
$h(M_1)$

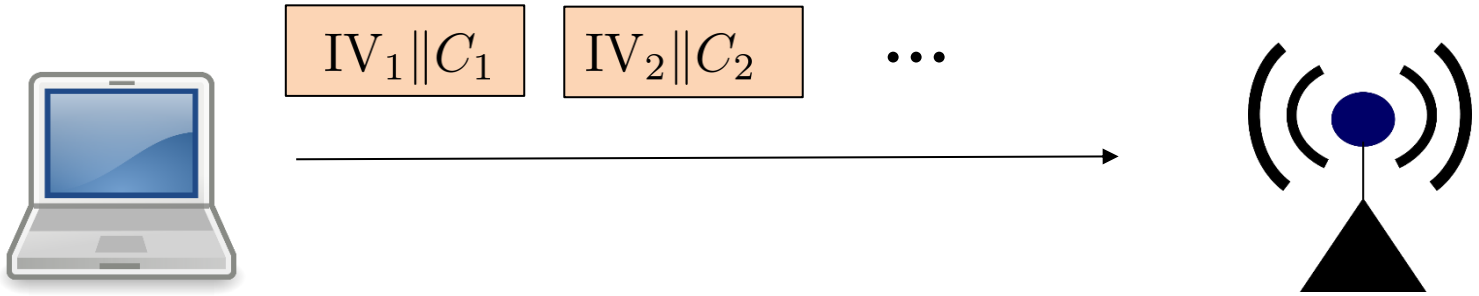
A Case Study: WEP

Used in IEEE WiFi standard

24-bit IV is a part of the ciphertext



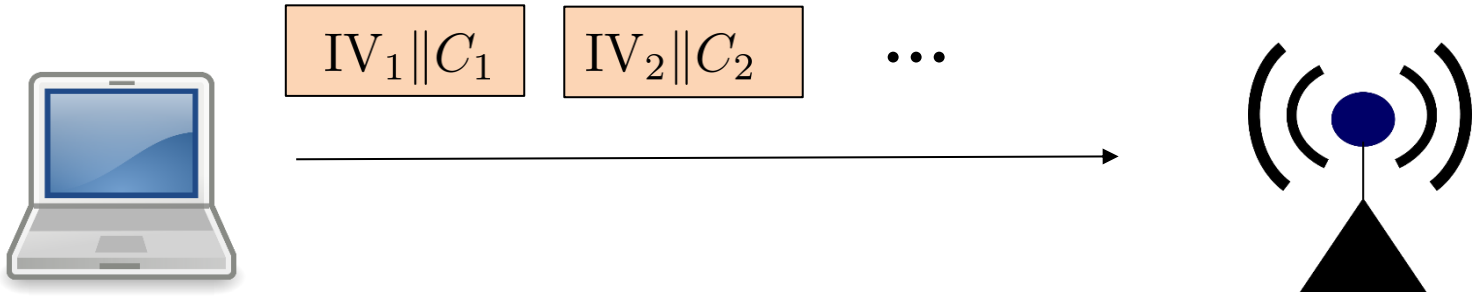
Attack 1: Exploiting Short IV



Assume all messages are of the same length, and fairly long

Goal: recover at least one message

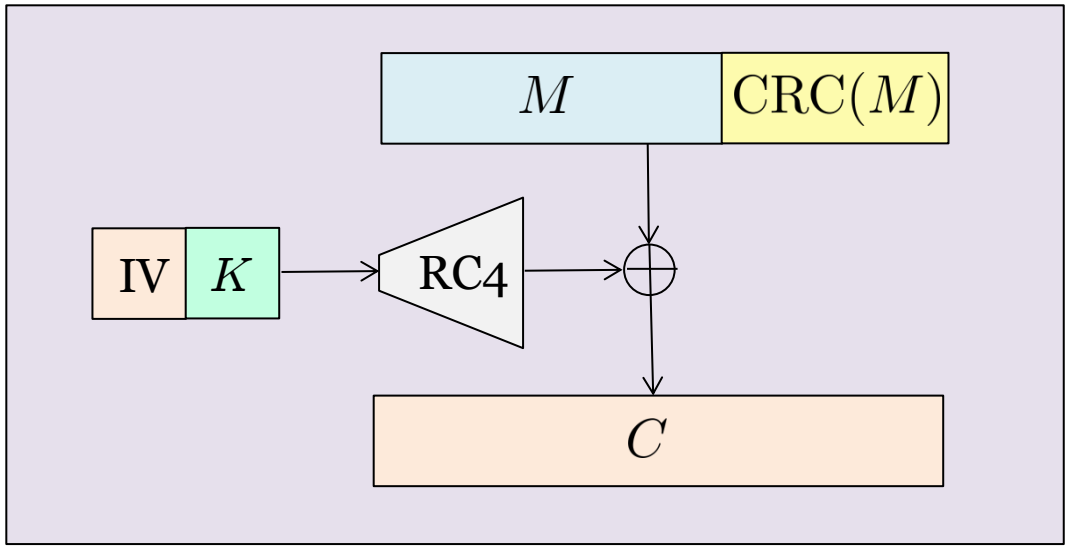
Attack 1: Exploiting Short IV

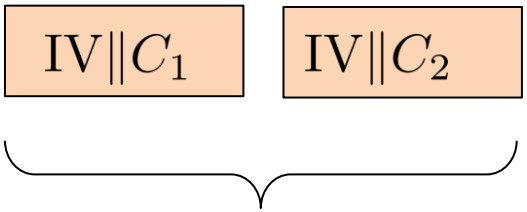


Aim for an IV collision

For 24-bit IV's, how many ctx to wait for collision prob ≈ 0.5?

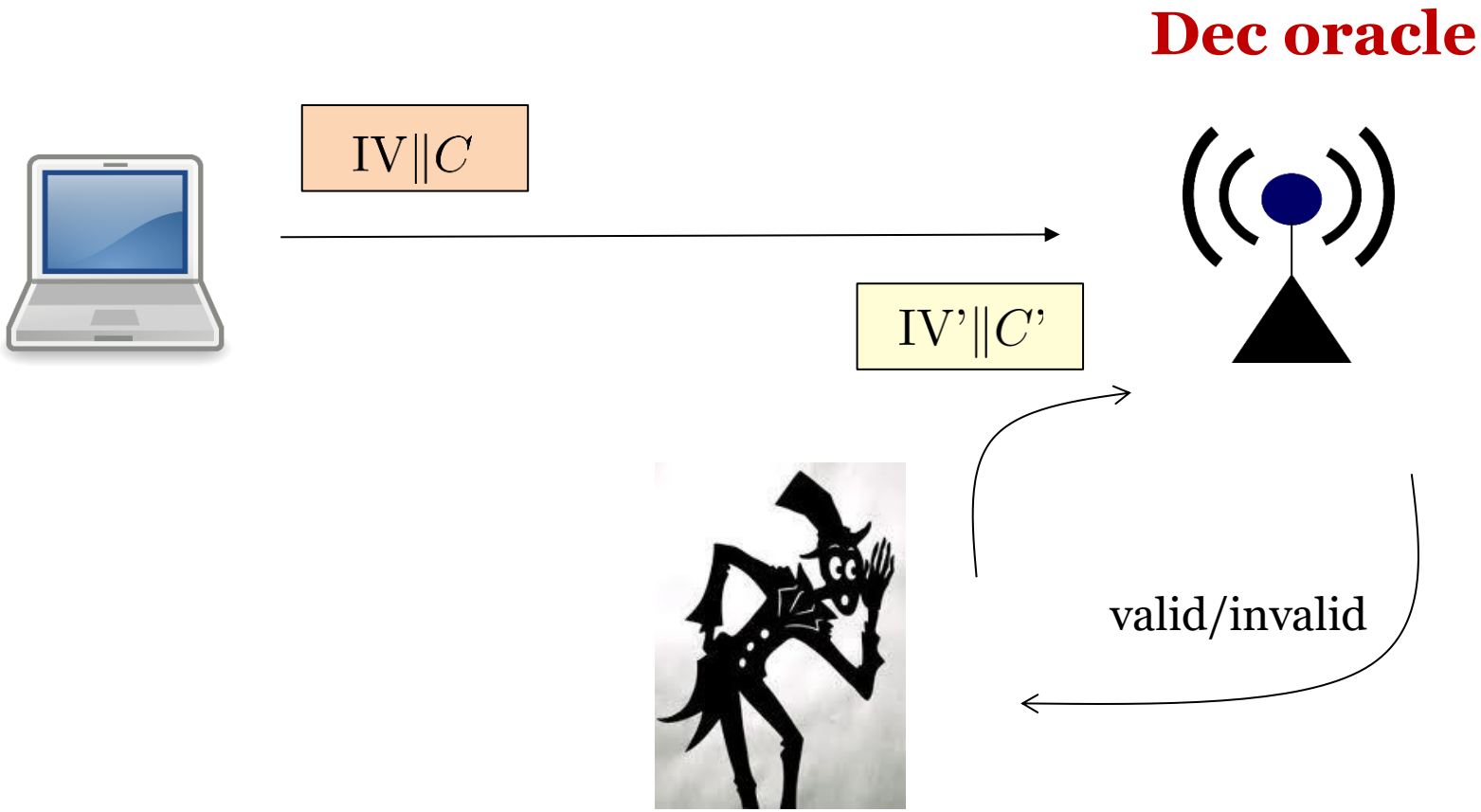
Attack 1: Exploiting Short IV





Same IV, can recover $M_1 \oplus M_2$

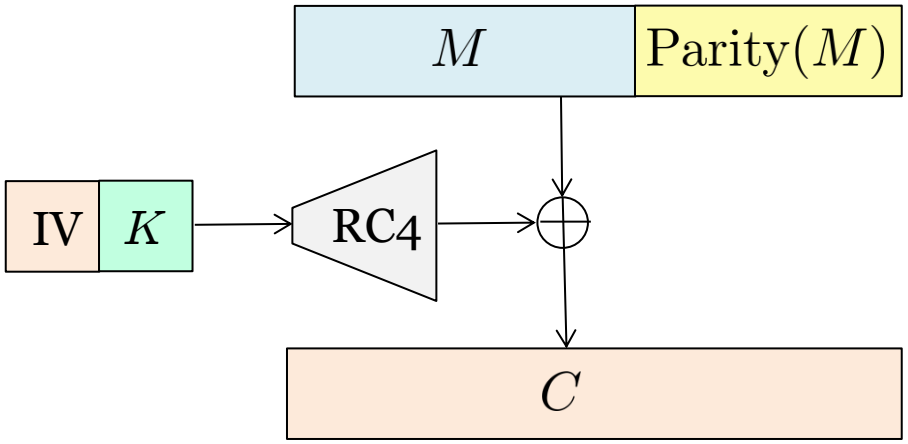
Attack 2: Chop-Chop Attack



Goal: recover the underlying message by exploiting Dec queries

Attack 2: Chop-Chop Attack

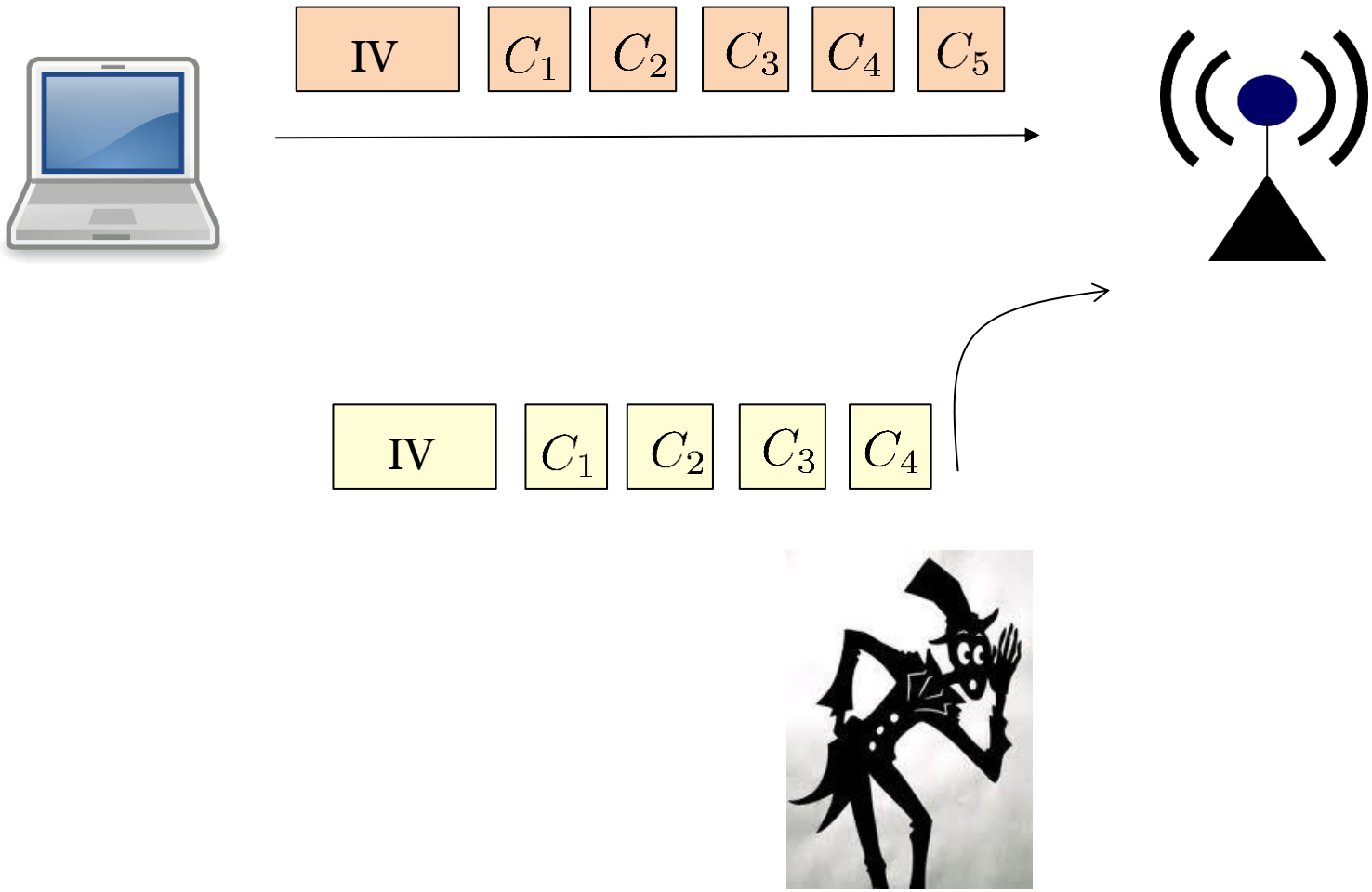
Illustrated Via A Simpler Variant of WEP



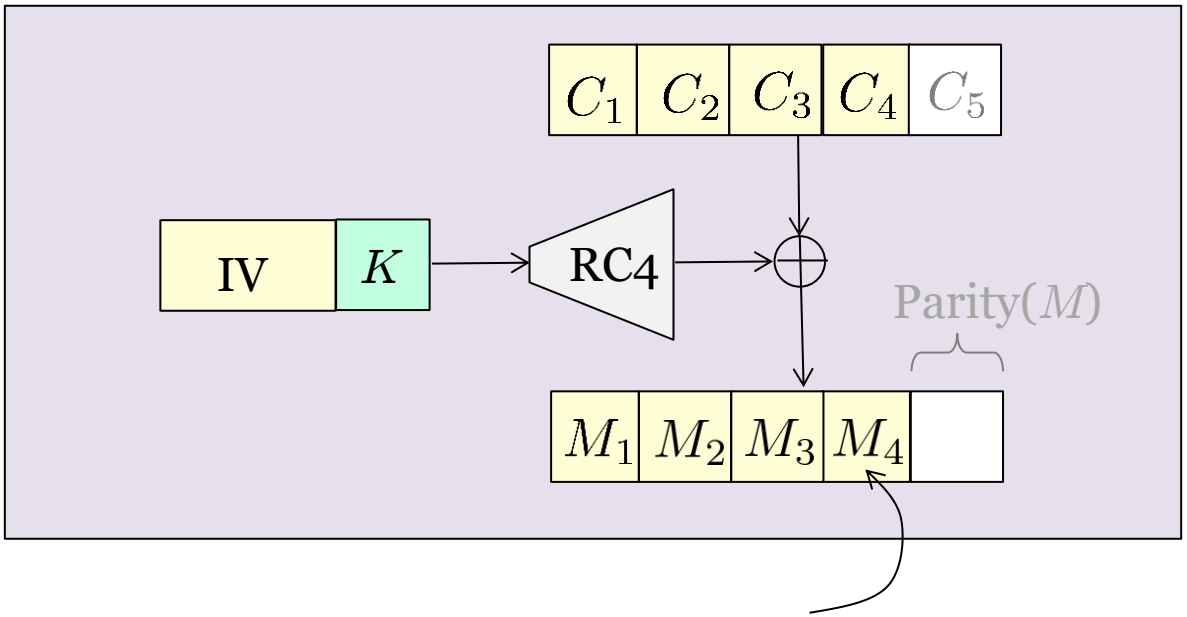
Example: $\text{Parity}(10011) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$

Attack 2: Chop-Chop Attack

Illustrated For 4-bit Message

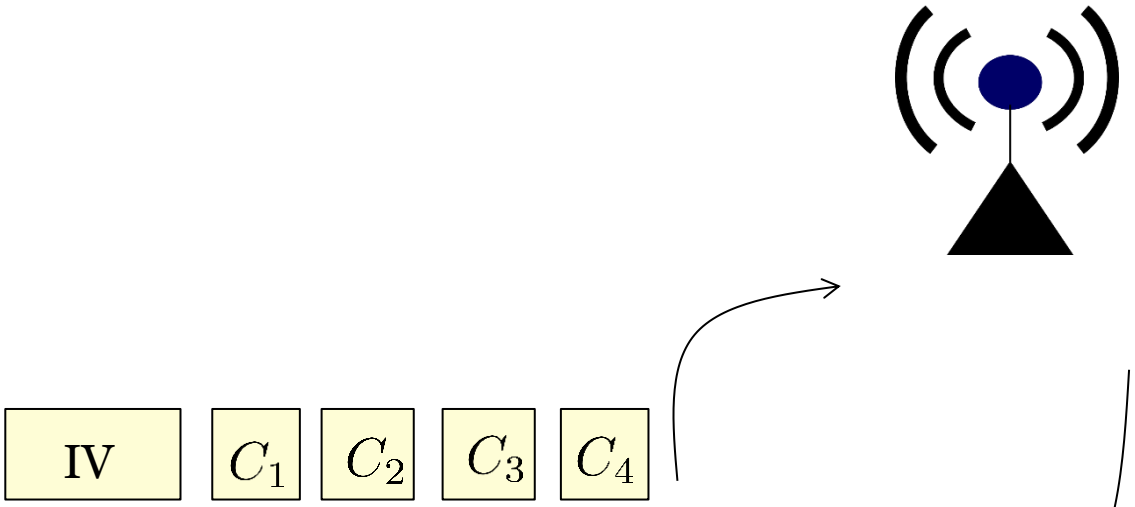


Decryption In CloseUp



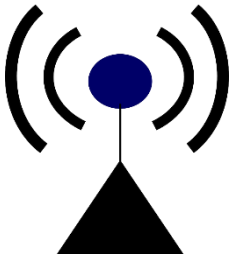
Compare with $\text{Parity}(M_1 M_2 M_3)$

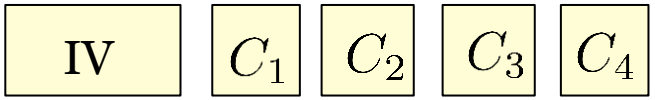
Exploit Decryption Response



If valid, $M_1 \oplus M_2 \oplus M_3 \oplus M_4 = 0$

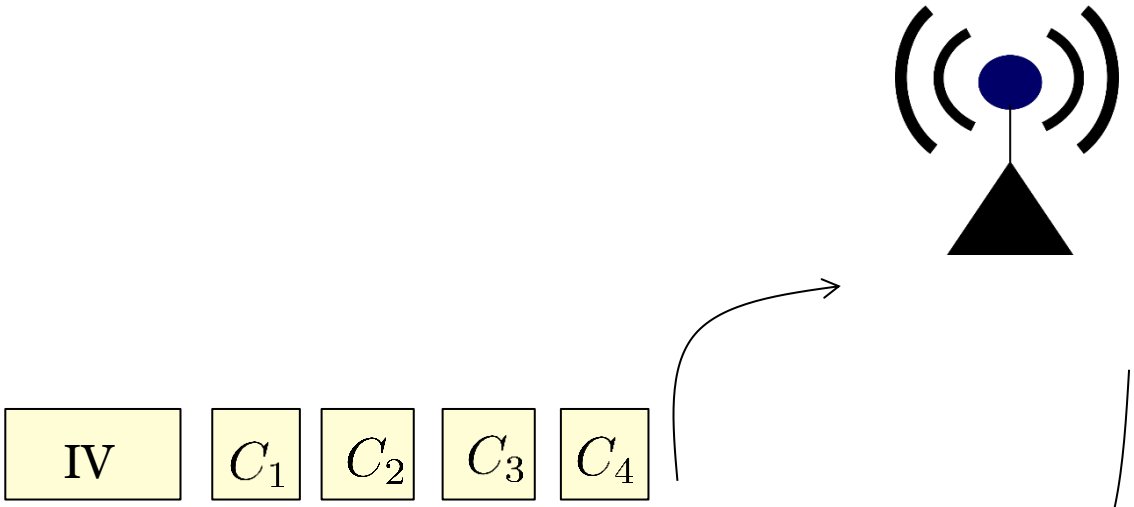
Exploit Decryption Response





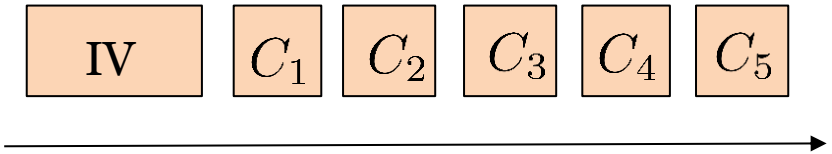
If invalid, $M_1 \oplus M_2 \oplus M_3 \oplus M_4 = 1$

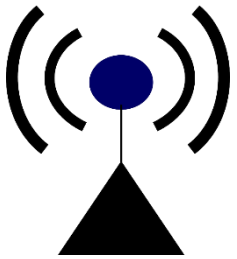
Exploit Decryption Response

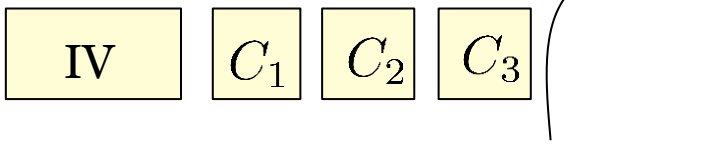


Learn $M_1 \oplus M_2 \oplus M_3 \oplus M_4$

Exploit Decryption Even Further







Learn $M_1 \oplus M_2 \oplus M_3$

Solve A System of Linear Equations

$$\begin{cases} M_1 \oplus M_2 \oplus M_3 \oplus M_4 = \square \\ M_1 \oplus M_2 \oplus M_3 = \square \\ M_1 \oplus M_2 = \square \\ M_1 = \square \end{cases}$$

Agenda

1. AE and Its Security Definitions

2. Failed Ways To Build AE

3. Generic Compositions

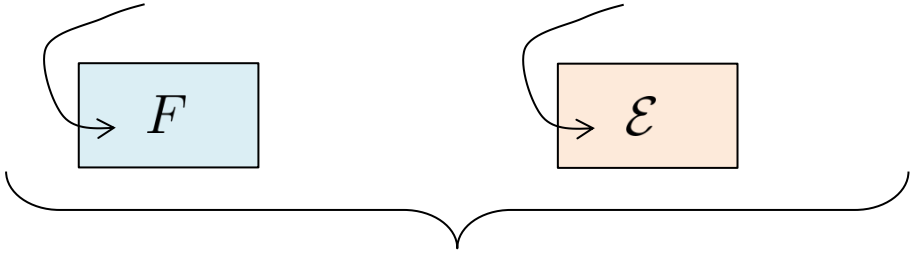
Constructing AE: Generic Composition

A good PRF, such as

Encrypted CBC-MAC

Privacy-only encryption

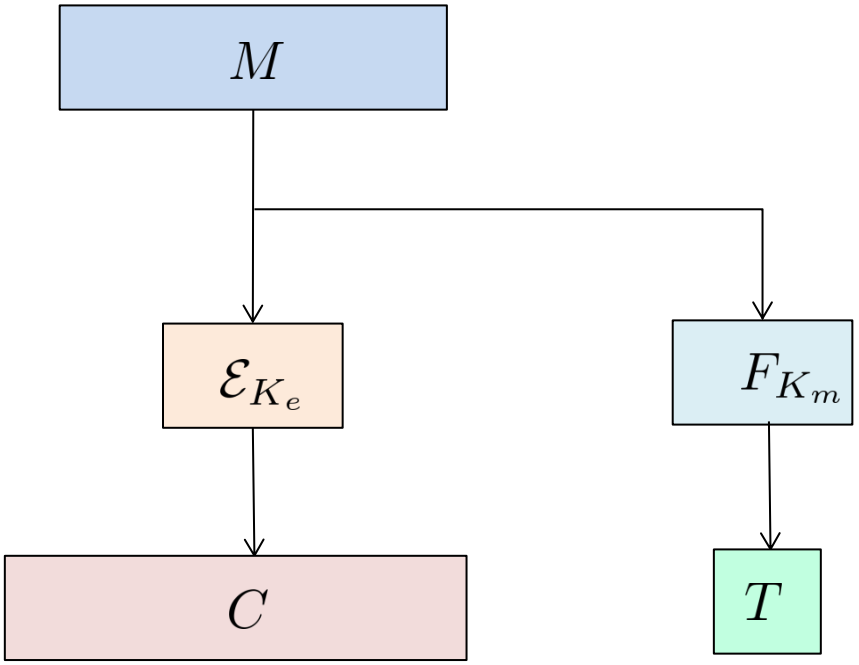
(such as CTR/CBC)



Compose them to build AE

Method	**Usage**
Encrypt-and-MAC	SSH
MAC-then-Encrypt	SSL/TLS
Encrypt-then-MAC	IPSec

Encrypt-and-MAC: Simple Composition



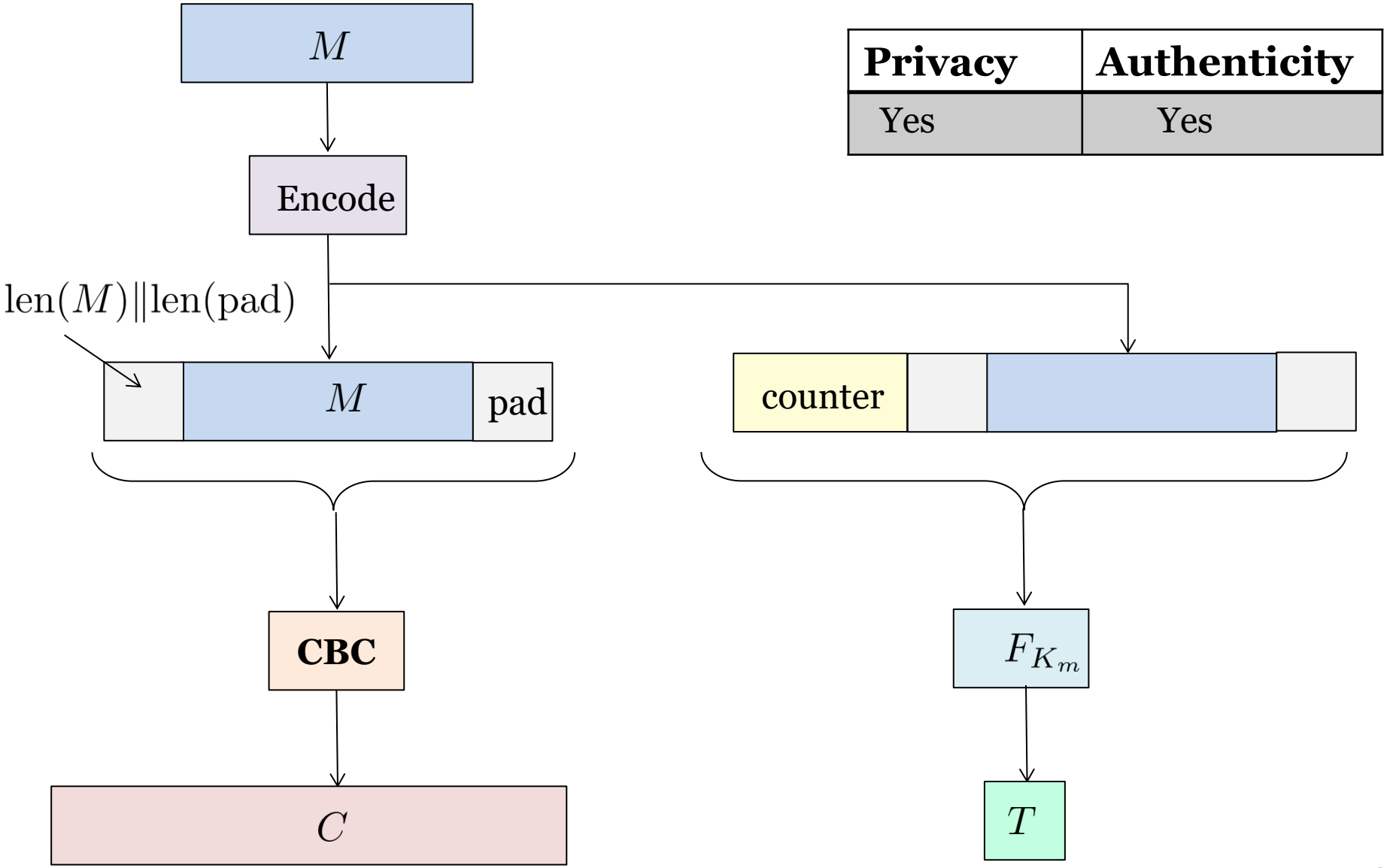
Privacy	Authenticity
No	No

for some bad encryption scheme

No privacy: encrypting the same message results in the same tag

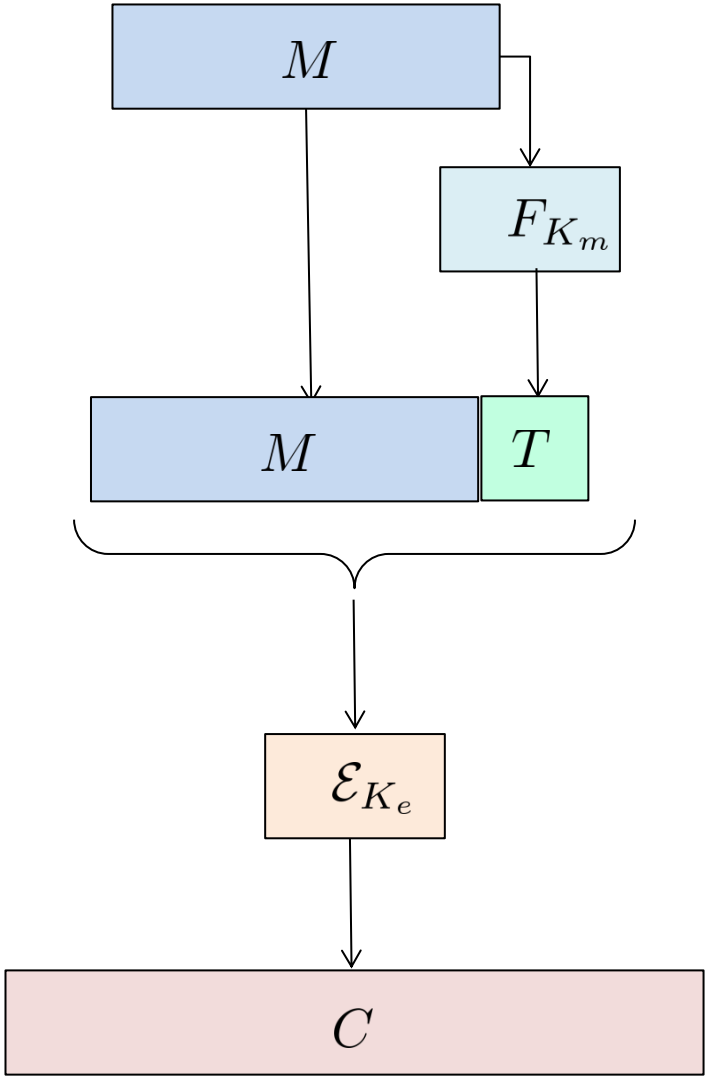
No authenticity if one can modify C such that decryption is unchanged.

Encrypt-and-MAC in SSH



Privacy	Authenticity
Yes	Yes

MAC-then-Encrypt

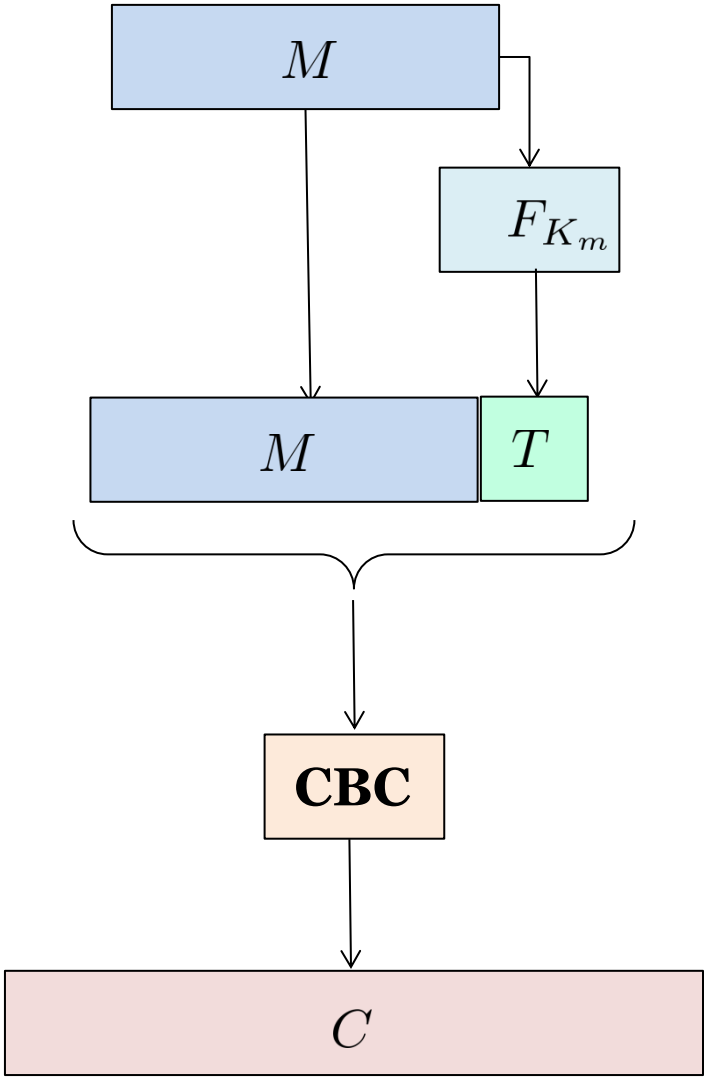


Privacy	Authenticity
Yes	No

for some bad encryption scheme

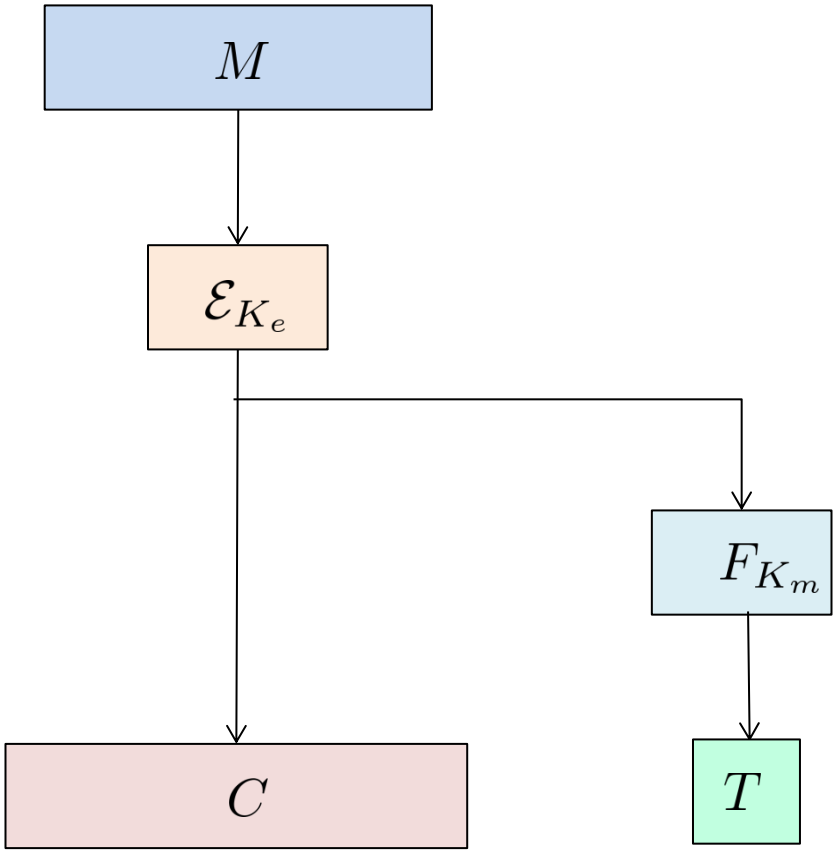
No authenticity if one can modify C such that decryption is unchanged.

MAC-then-Encrypt in TLS



Privacy	Authenticity
Yes	Yes

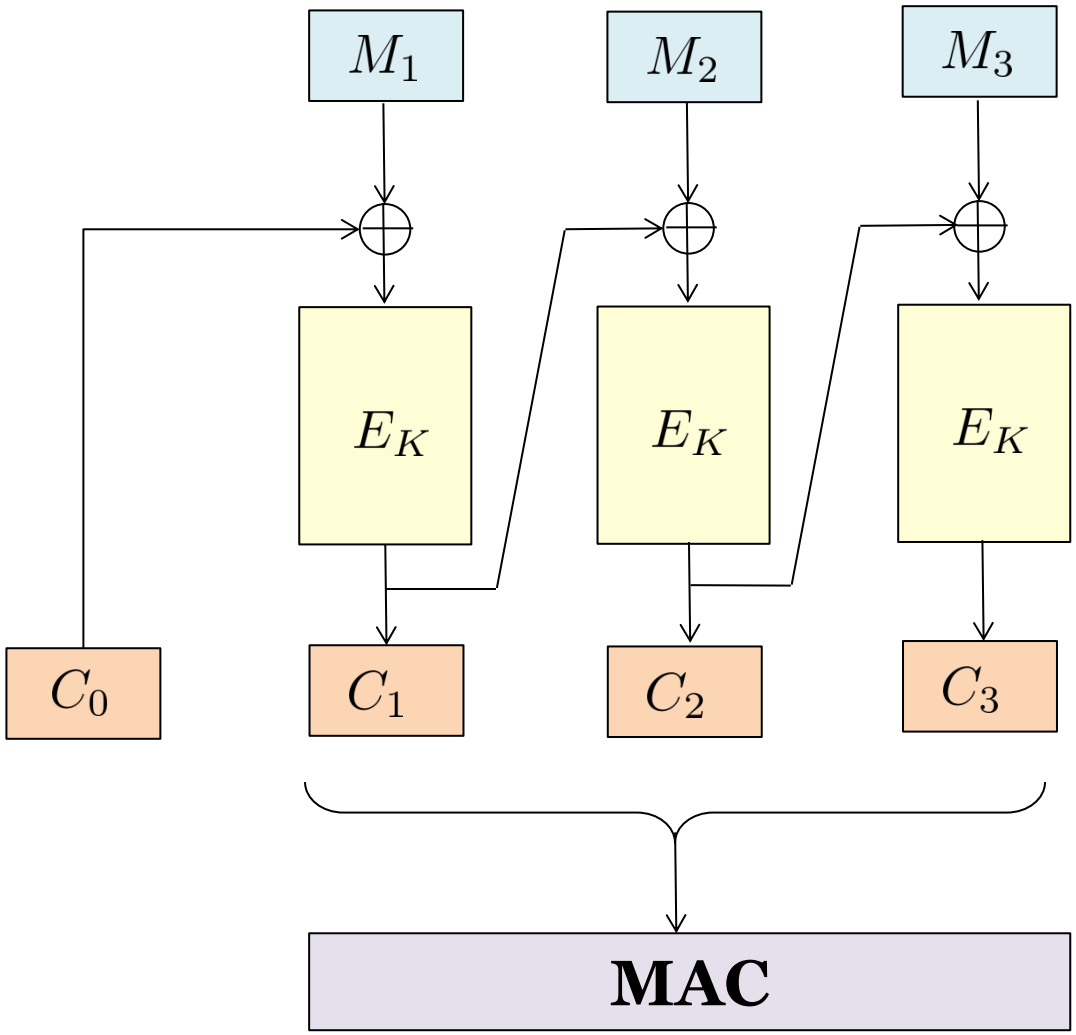
Encrypt-then-MAC



Privacy	Authenticity
Yes	Yes

A Common Pitfall in Implementing EtM

Happened in ISO 1972 standard, and in RNCryptor of iOS



Forget to feed IV into MAC

Break auth with one query