Authenticated Encryption

Viet Tung Hoang

Agenda

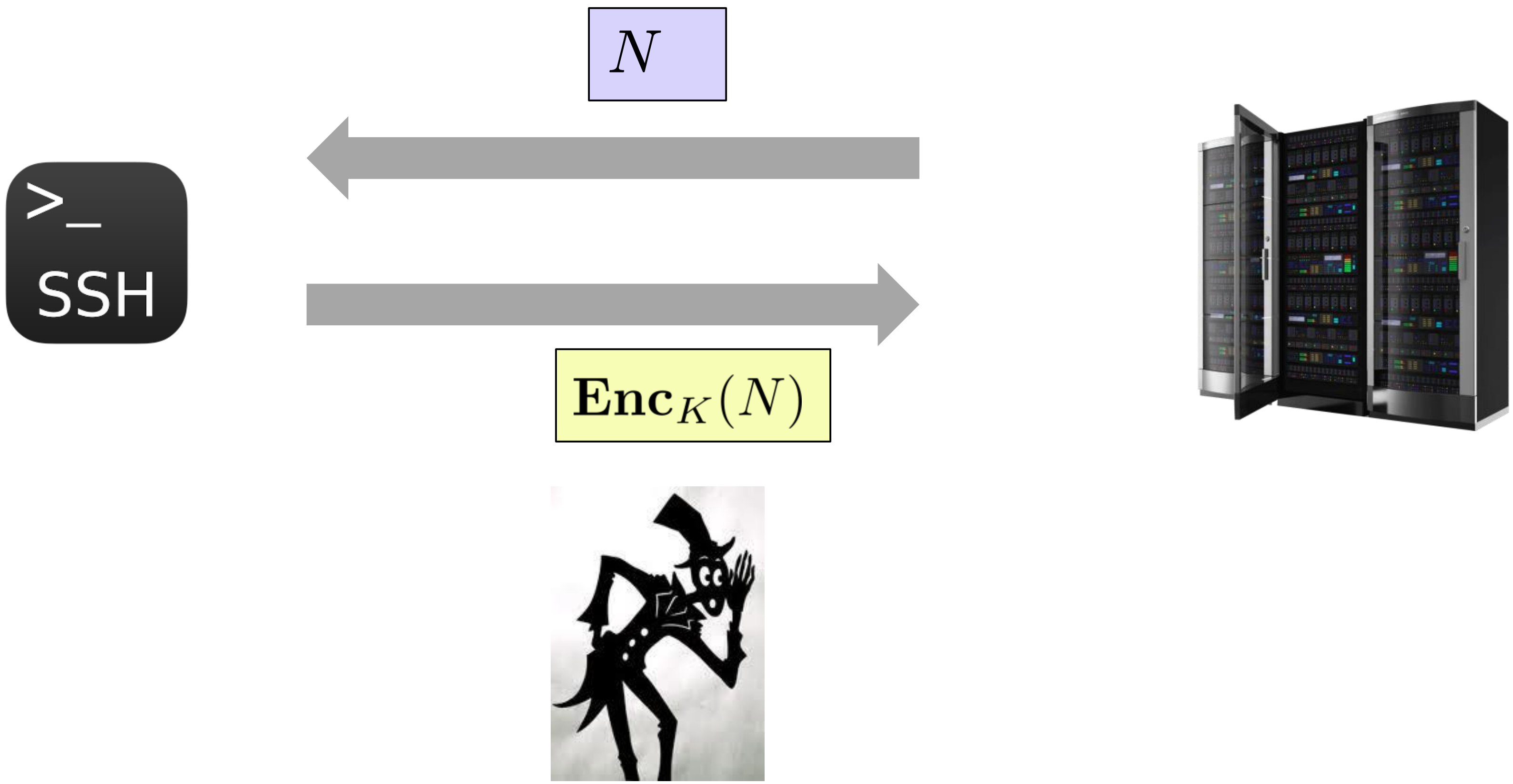
1. AE and Its Security Definitions

2. Failed Ways to Build AE

3. Generic Compositions

4. Padding-Oracle Attack on SSL/TLS

Motivation: Challenge-Response Revisited



Question: Break this identification mechanism if encryption is CTR.

Solution: Authenticated Encryption

Transfer $5 to account 12345

K

K

K

Privacy — **Authenticity**

Encryption scheme

Authenticated Encryption
Achieve **both** of these aims

MAC

Authenticated Encryption (AE)

Emerged ~ 2000

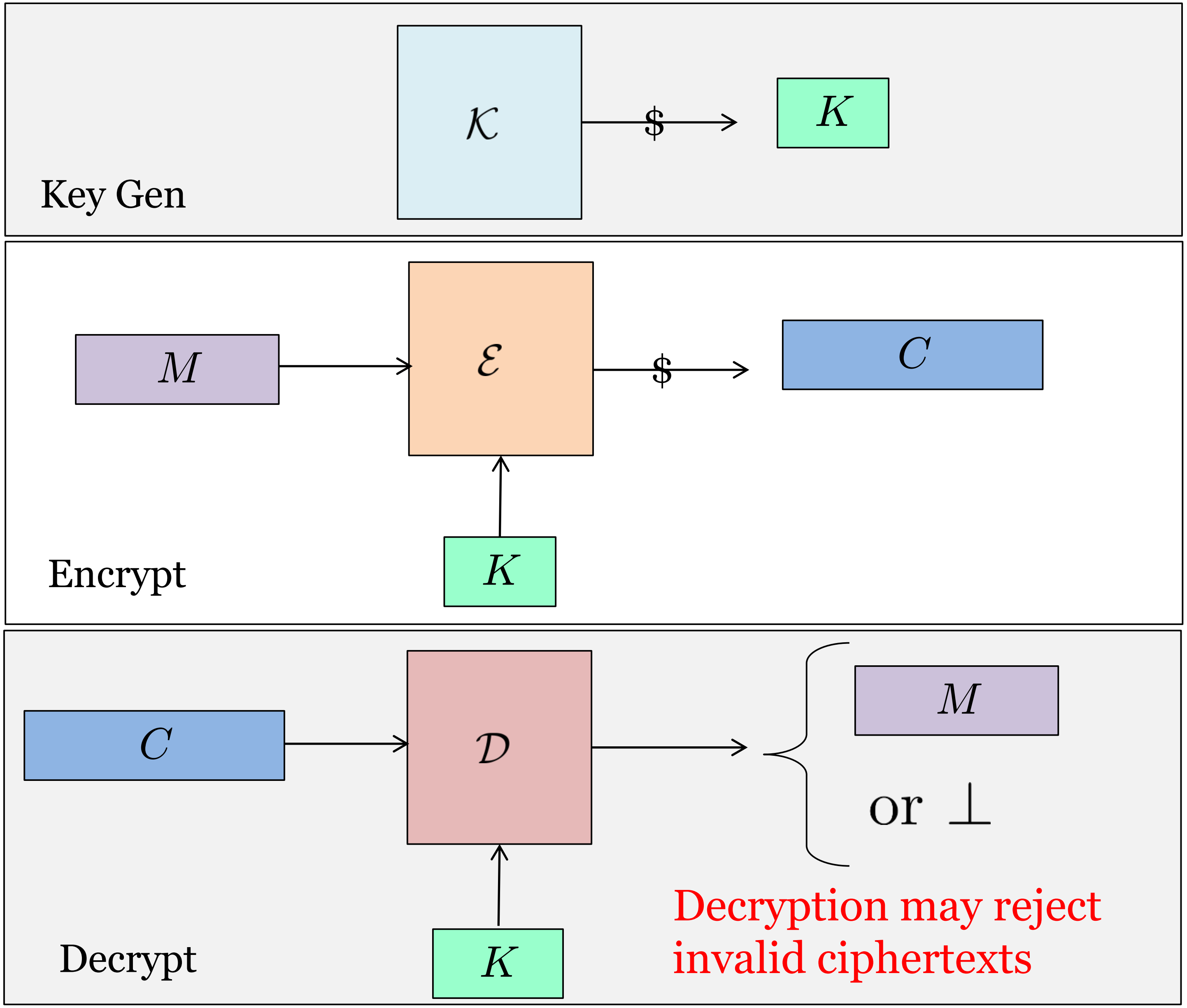
Begin with two **realizations**

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

↓

Provide an easier-to-correctly-use abstraction boundary

AE Syntax



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

Defining Security for AE

Authenticity

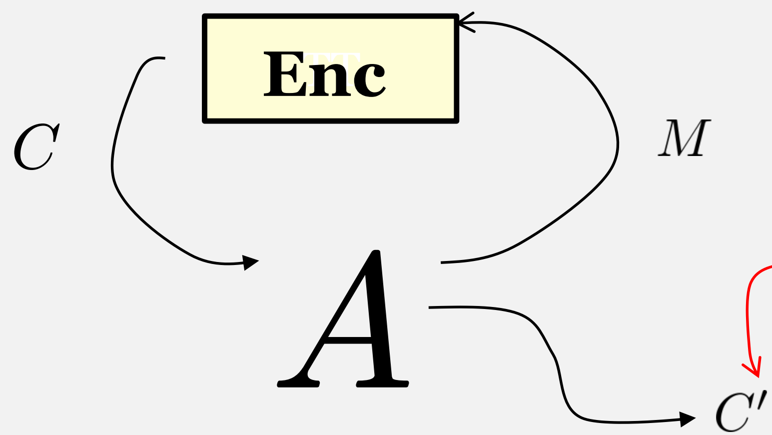
-Use Left-or-Right security for privacy

$\text{Auth}_{\mathcal{E}}$

procedure Initialize()
$K \leftarrow\!\!\$\ \mathcal{K}$

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

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



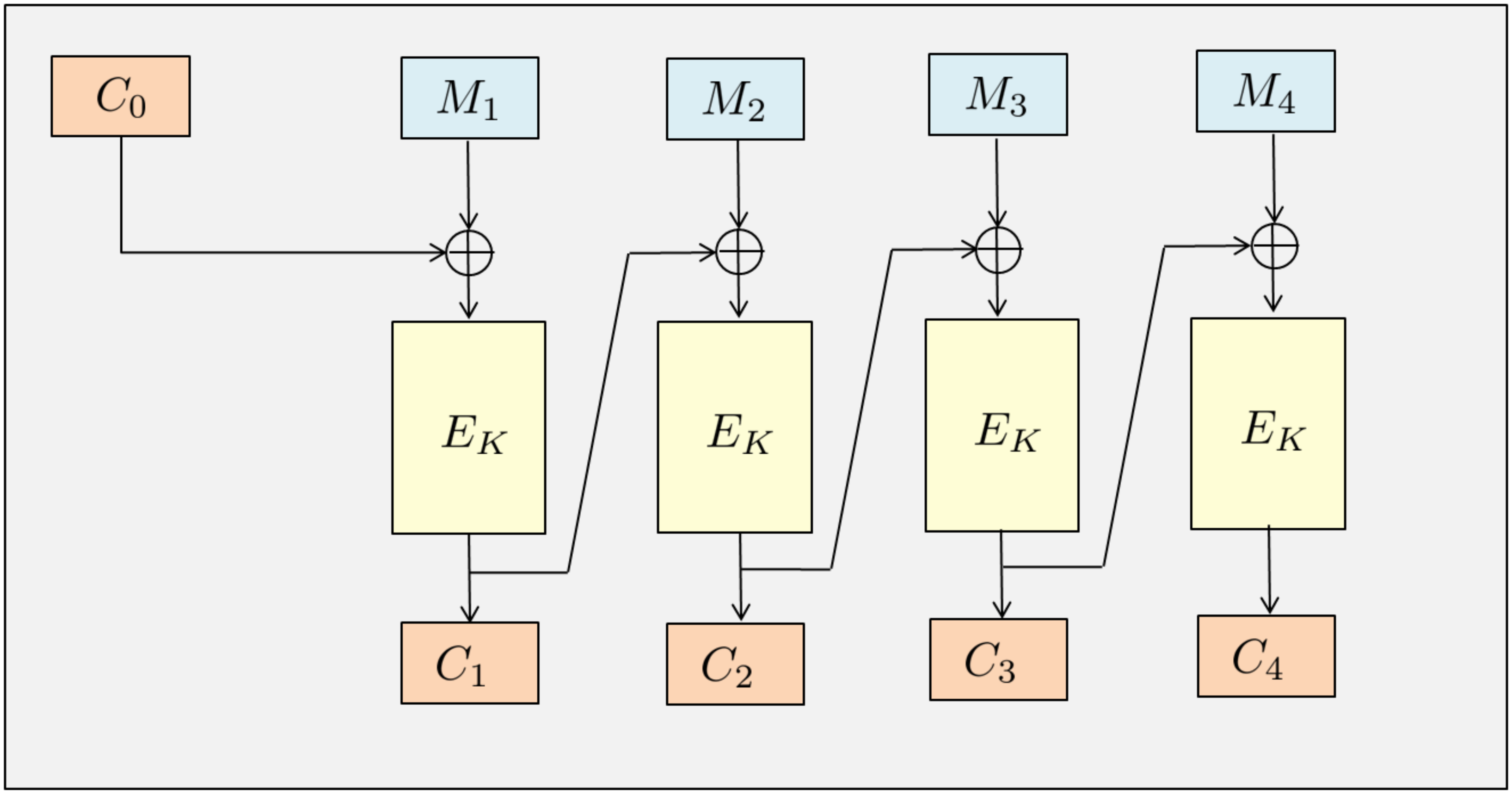
Must never receive from Enc

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

Agenda

1. AE and Its Security Definitions
2. **Failed Ways to Build AE**
3. Generic Compositions
4. Padding-Oracle Attack on SSL/TLS

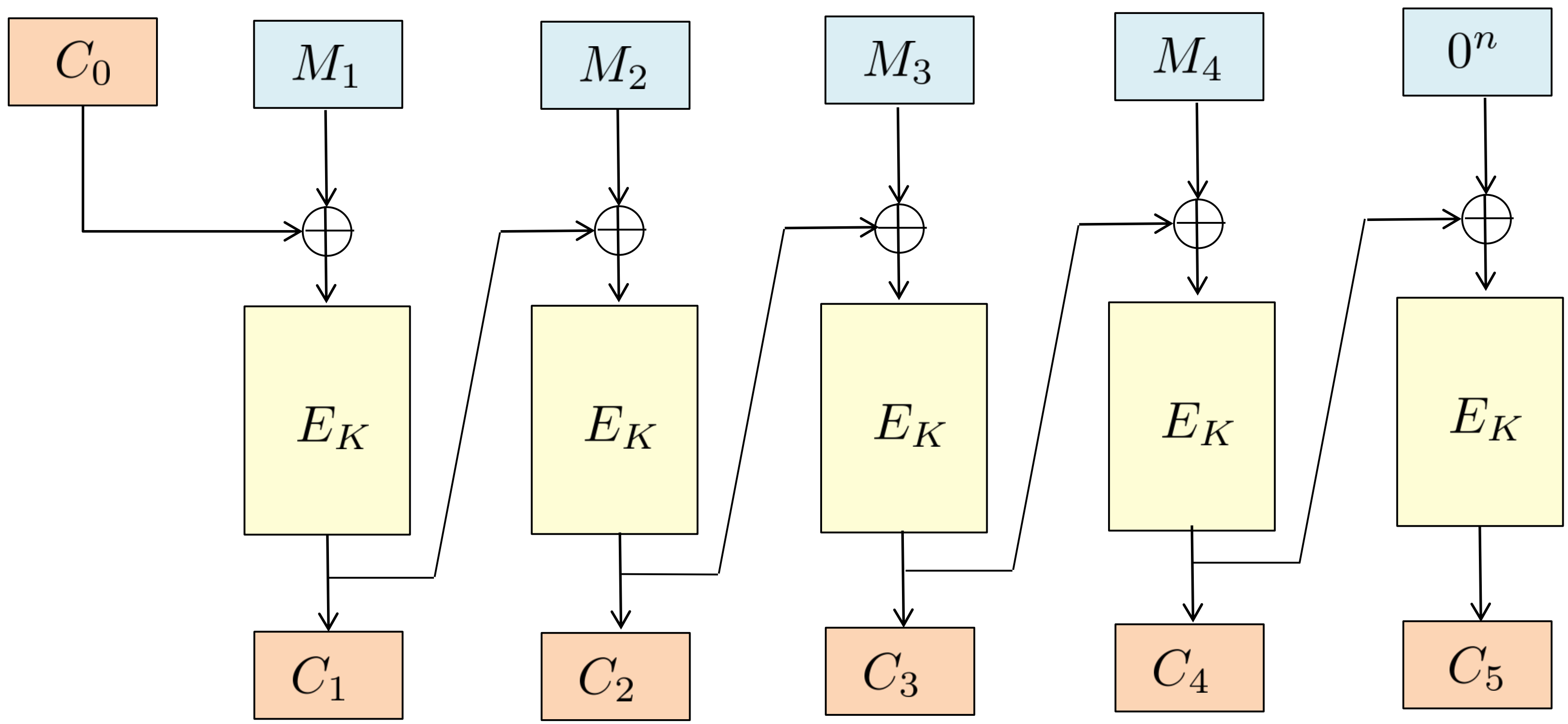
Plain Encryption Doesn't Provide Authenticity



Question: Does CBC provide authenticity?

Answer: No, because any ciphertext has valid decryption

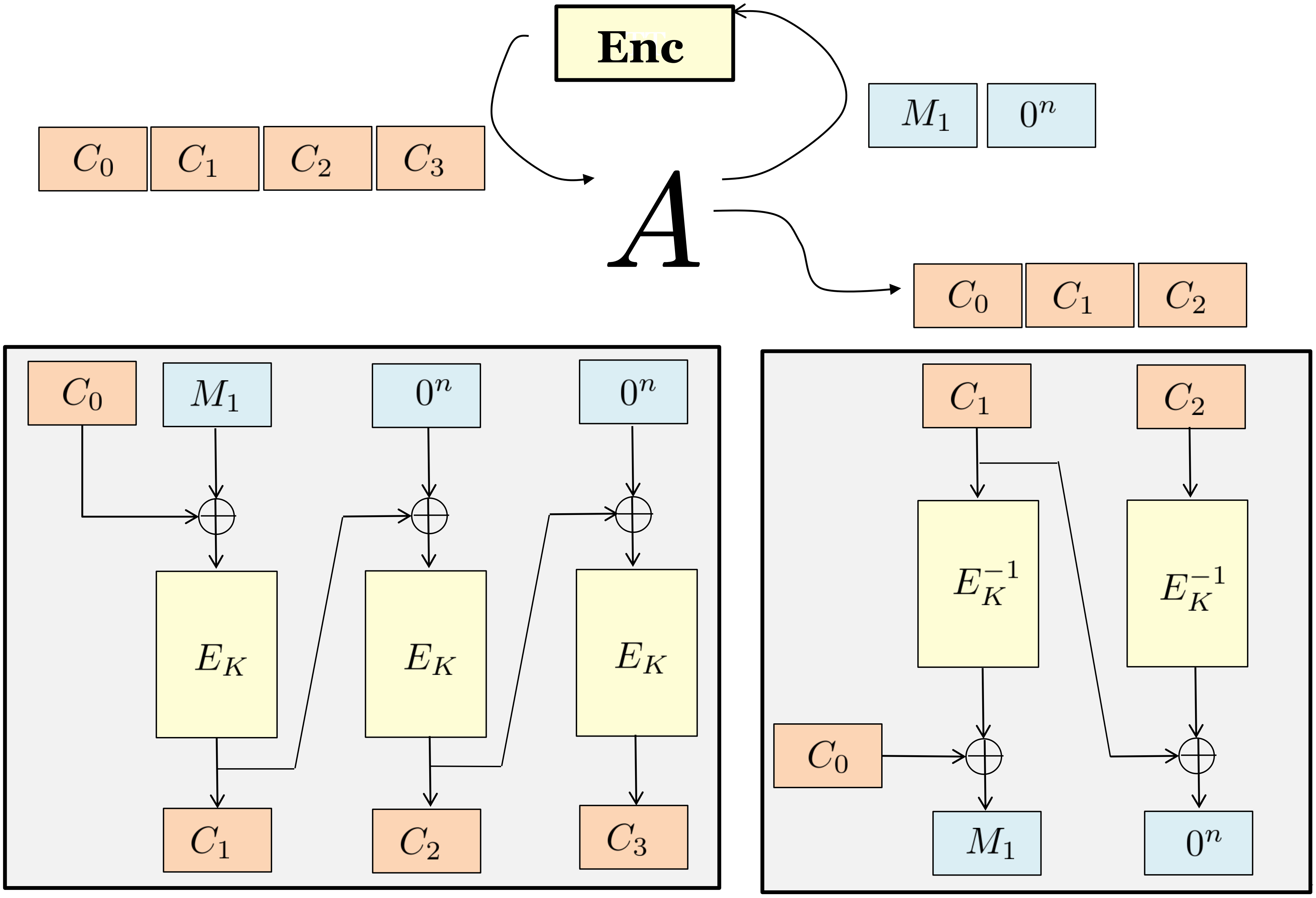
A Bad Fix: CBC with Redundancy



On decryption, verify the decrypted last block is zero.

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

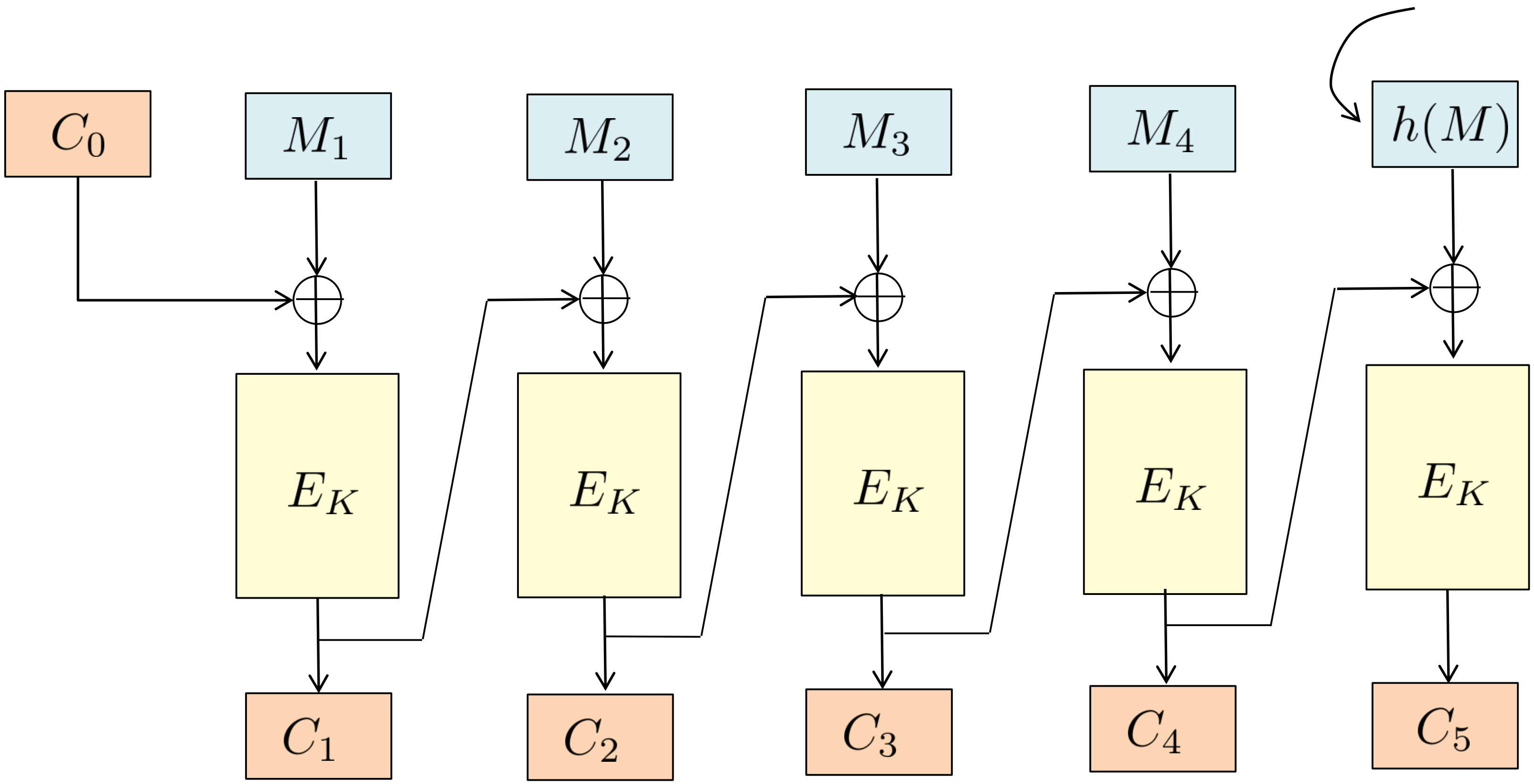
An Attack



Enc
C_0 C_1 C_2 C_3
$\mathcal{A}$
M_1 0^n
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

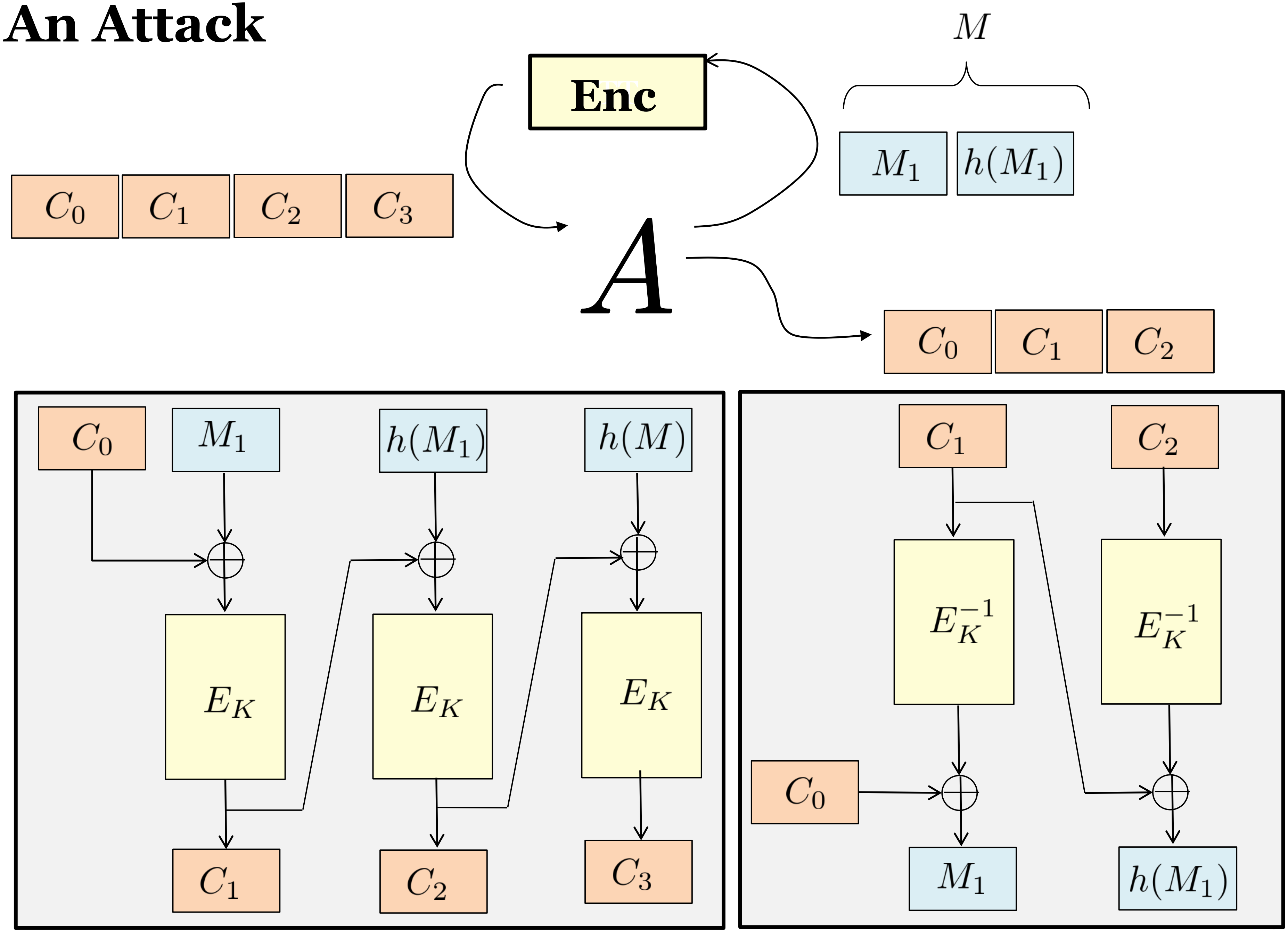
Some (unkeyed) "redundancy" function, such as checksum



The redundancy is verified upon decryption

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

An Attack



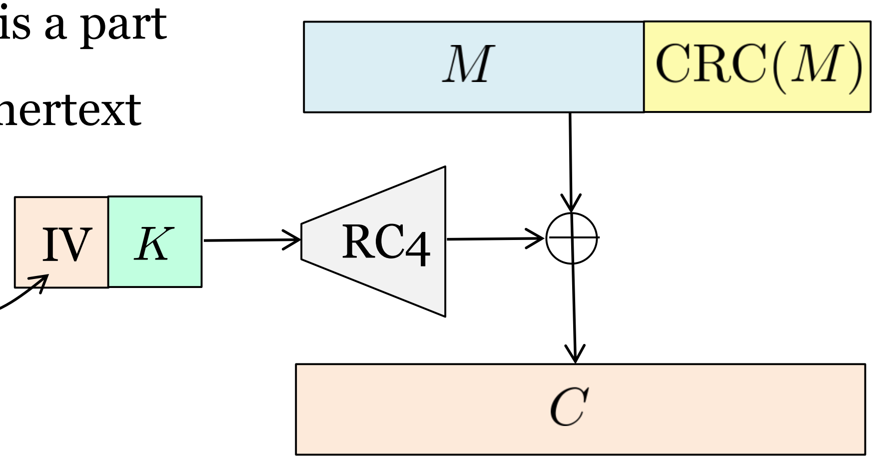
Enc
M
M₁
h(M₁)
C₀ C₁ C₂ C₃
A
C₀ C₁ C₂
C₀
M₁
h(M₁)
h(M)
E_K
E_K
E_K
C₁ C₂ C₃
C₁ C₂
E_K^{-1}
E_K^{-1}
C₀
M₁
h(M₁)

A Case Study: WEP

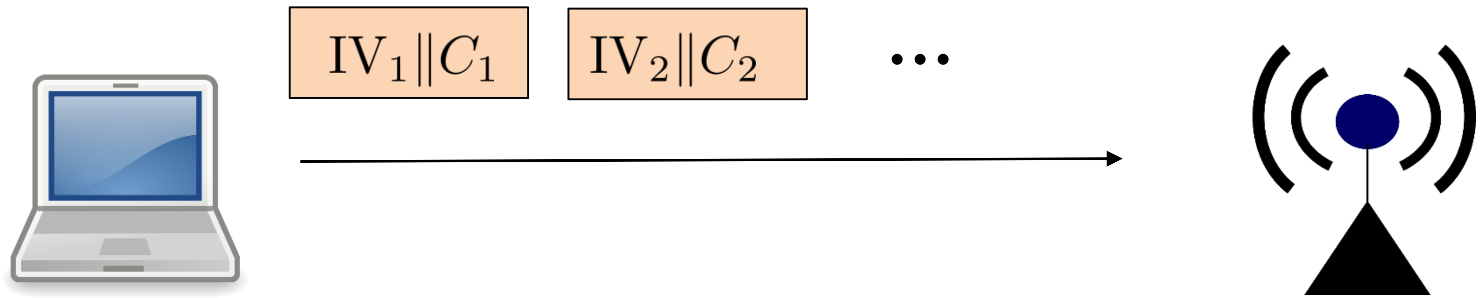
Used in IEEE WiFi standard

24-bit IV is a part

of the ciphertext



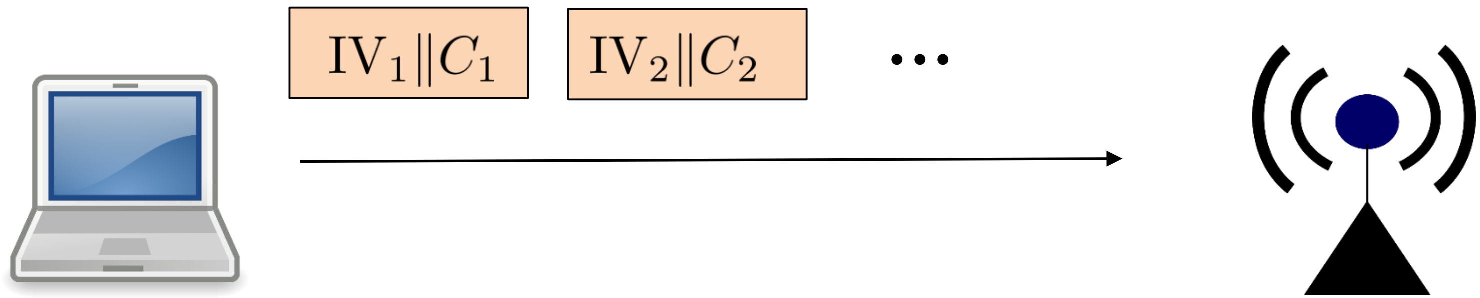
Attack 1: Exploiting Short IV



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

Goal: recover at least one message

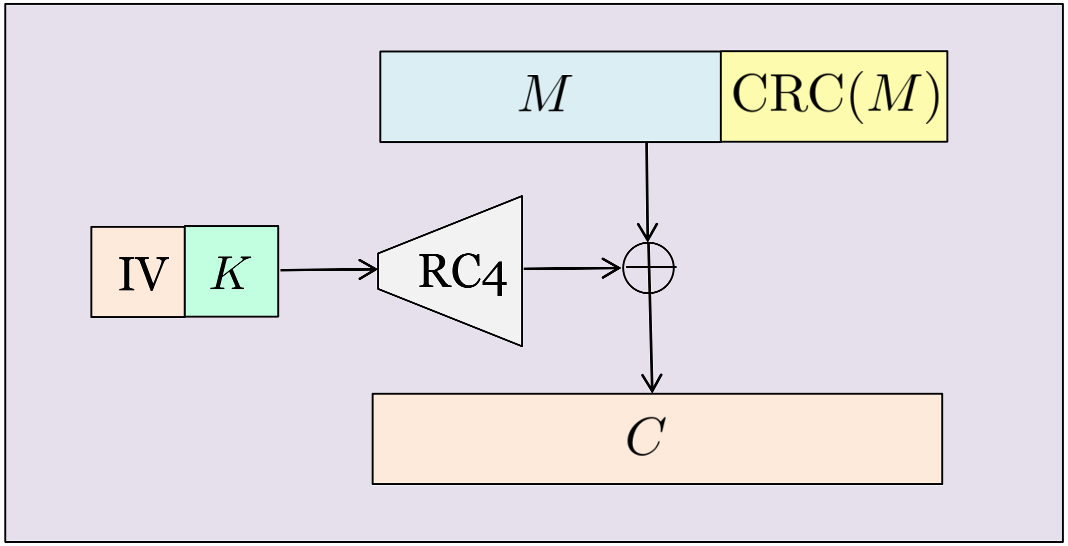
Attack 1: Exploiting Short IV

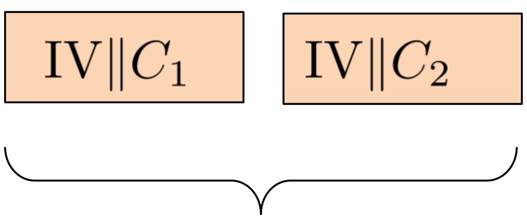


Aim for an IV collision

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

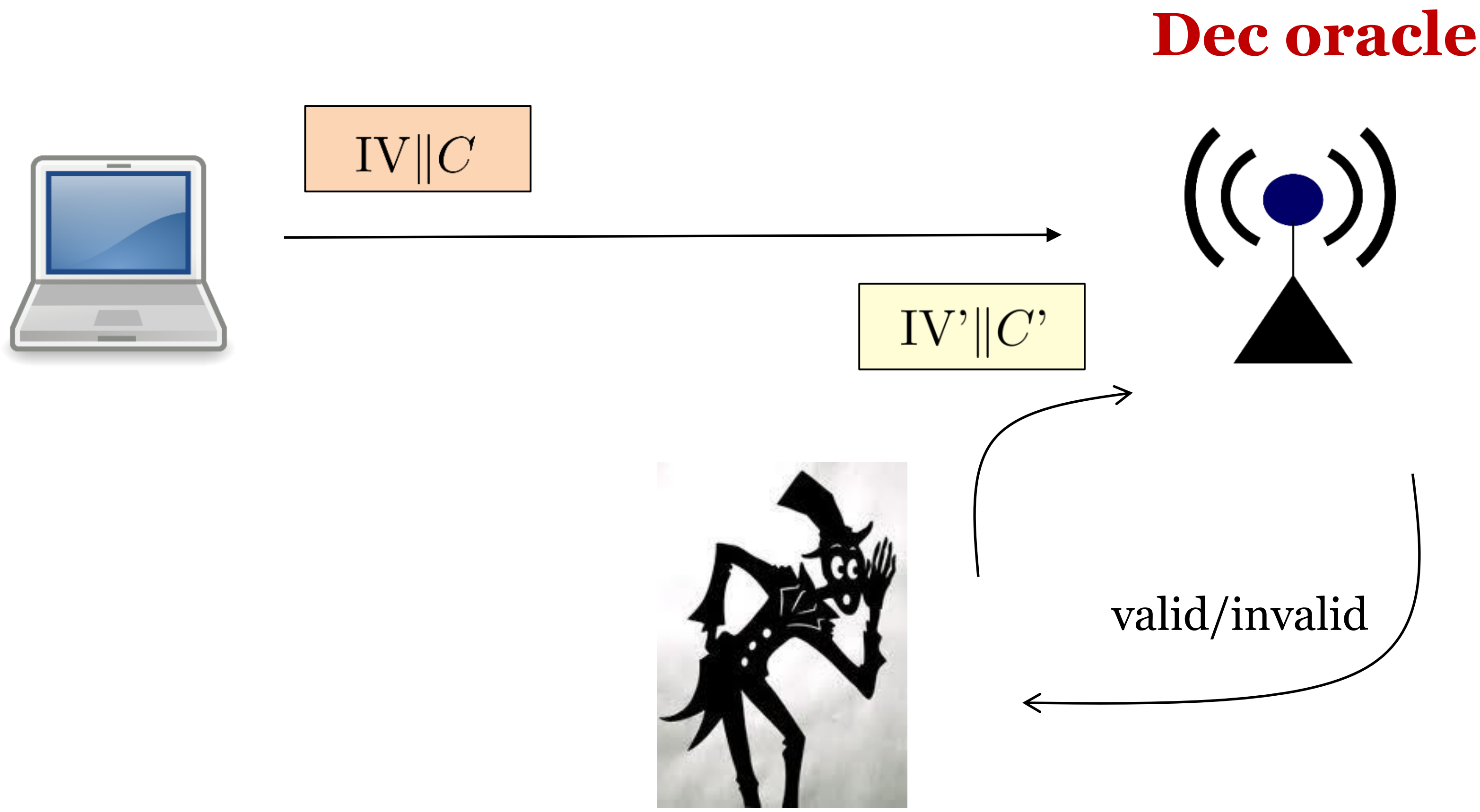
Attack 1: Exploiting Short IV





Same IV, can recover $M_1 \oplus M_2$

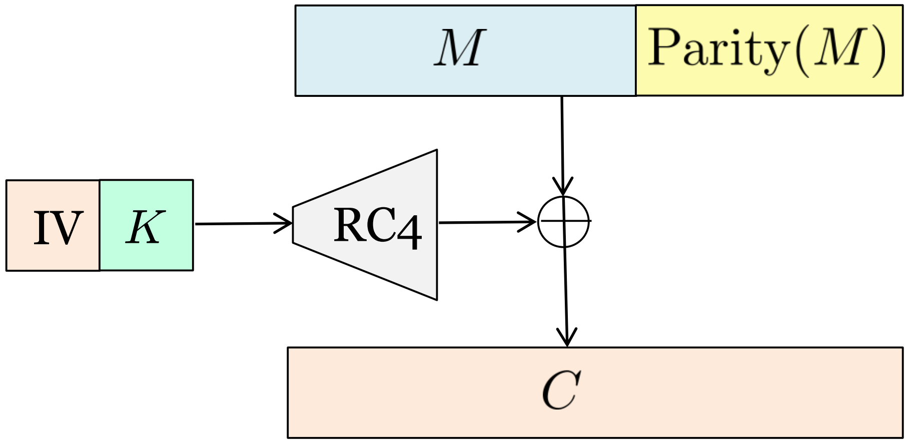
Attack 2: Chop-Chop Attack



Goal: recover the underlying message by exploiting Dec queries

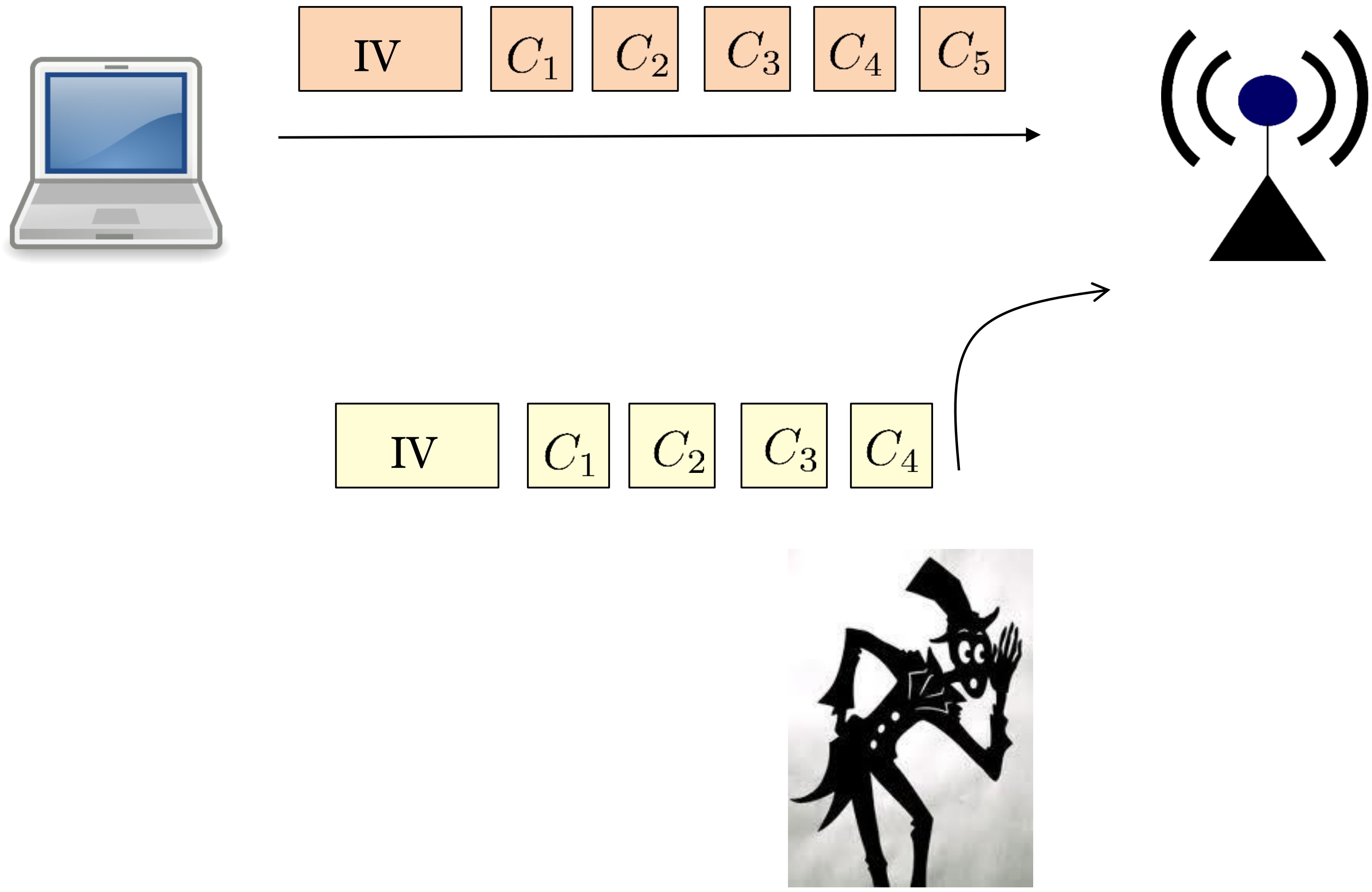
Attack 2: Chop-Chop Attack

Illustrated Via A Simpler Variant of WEP

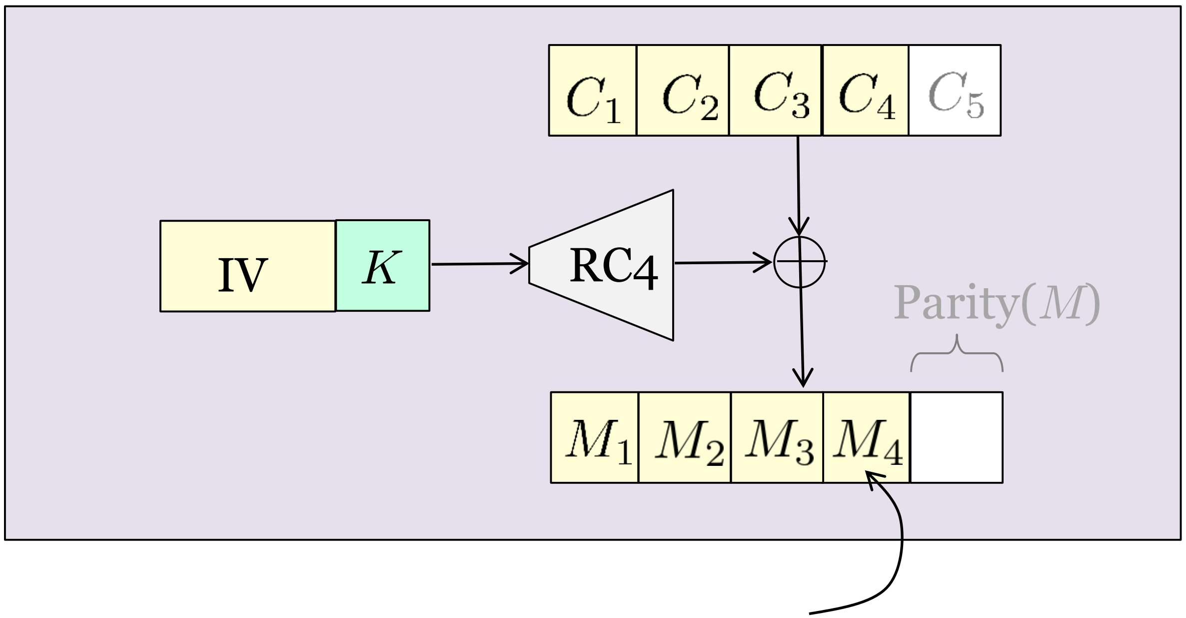


Example: $\text{Parity}(10011) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$

Attack 2: Chop-Chop Attack
Illustrated For 4-bit Message

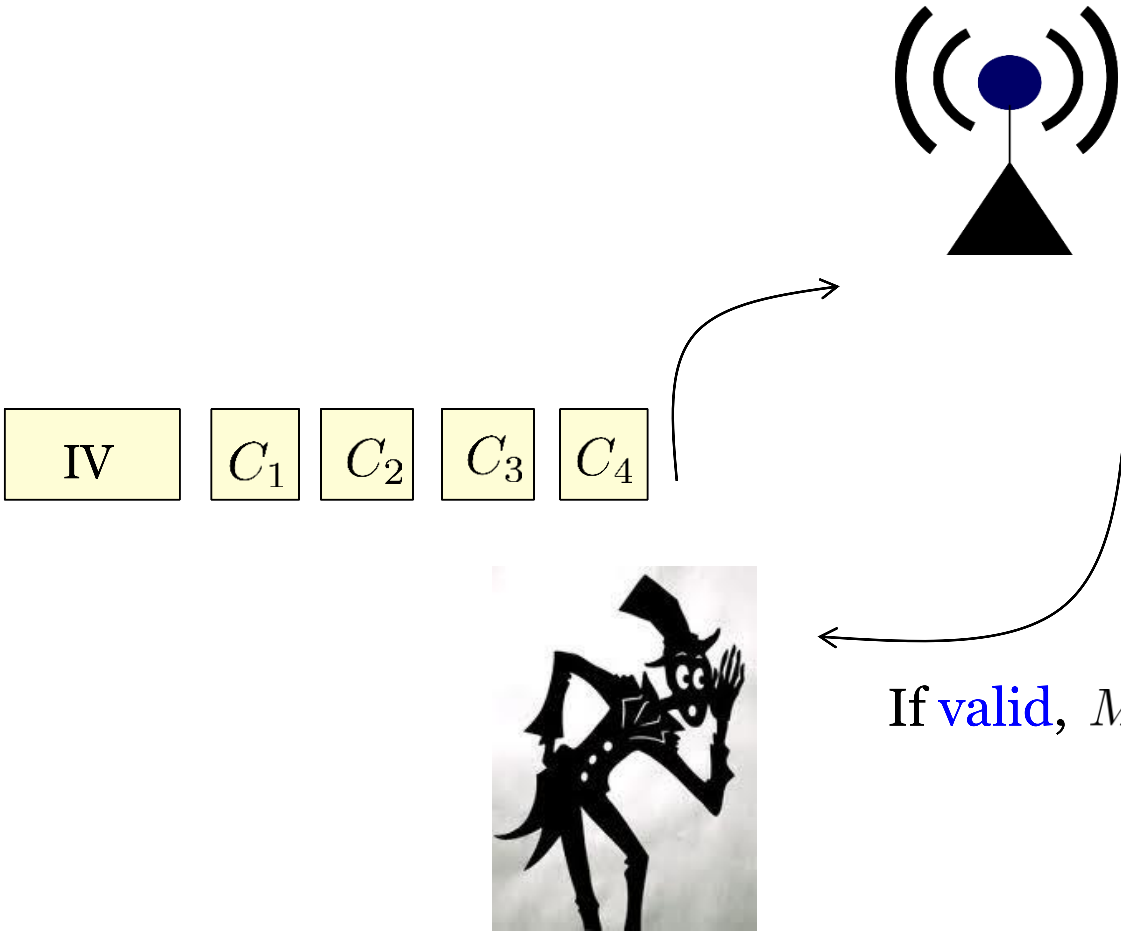


Decryption CloseUp



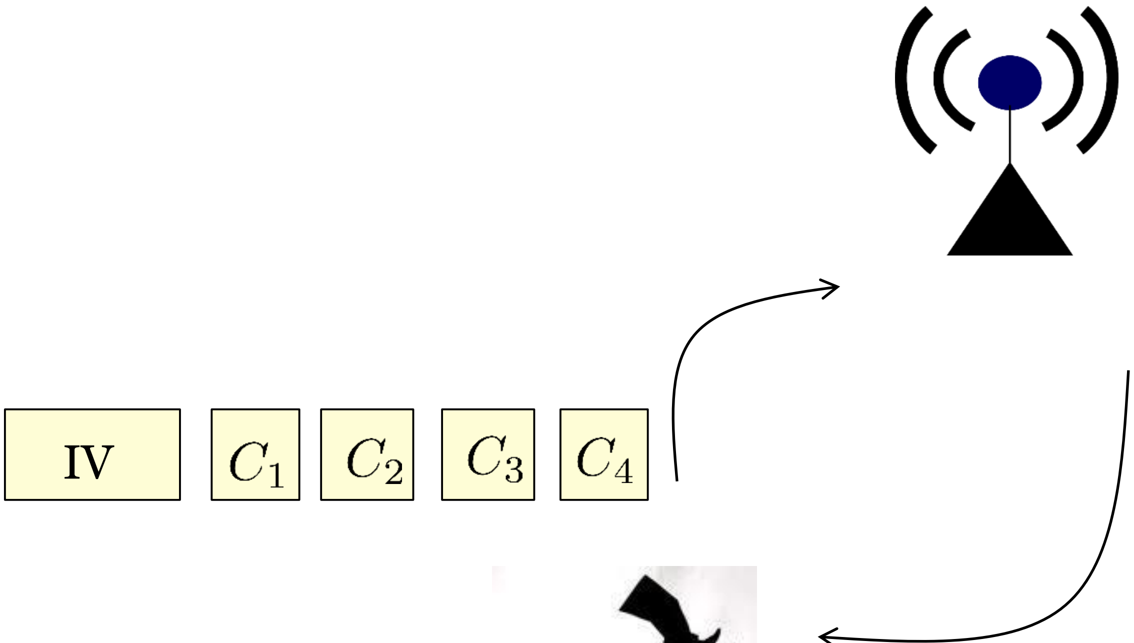
Compare with $\text{Parity}(M_1 M_2 M_3)$

Exploit Decryption Response



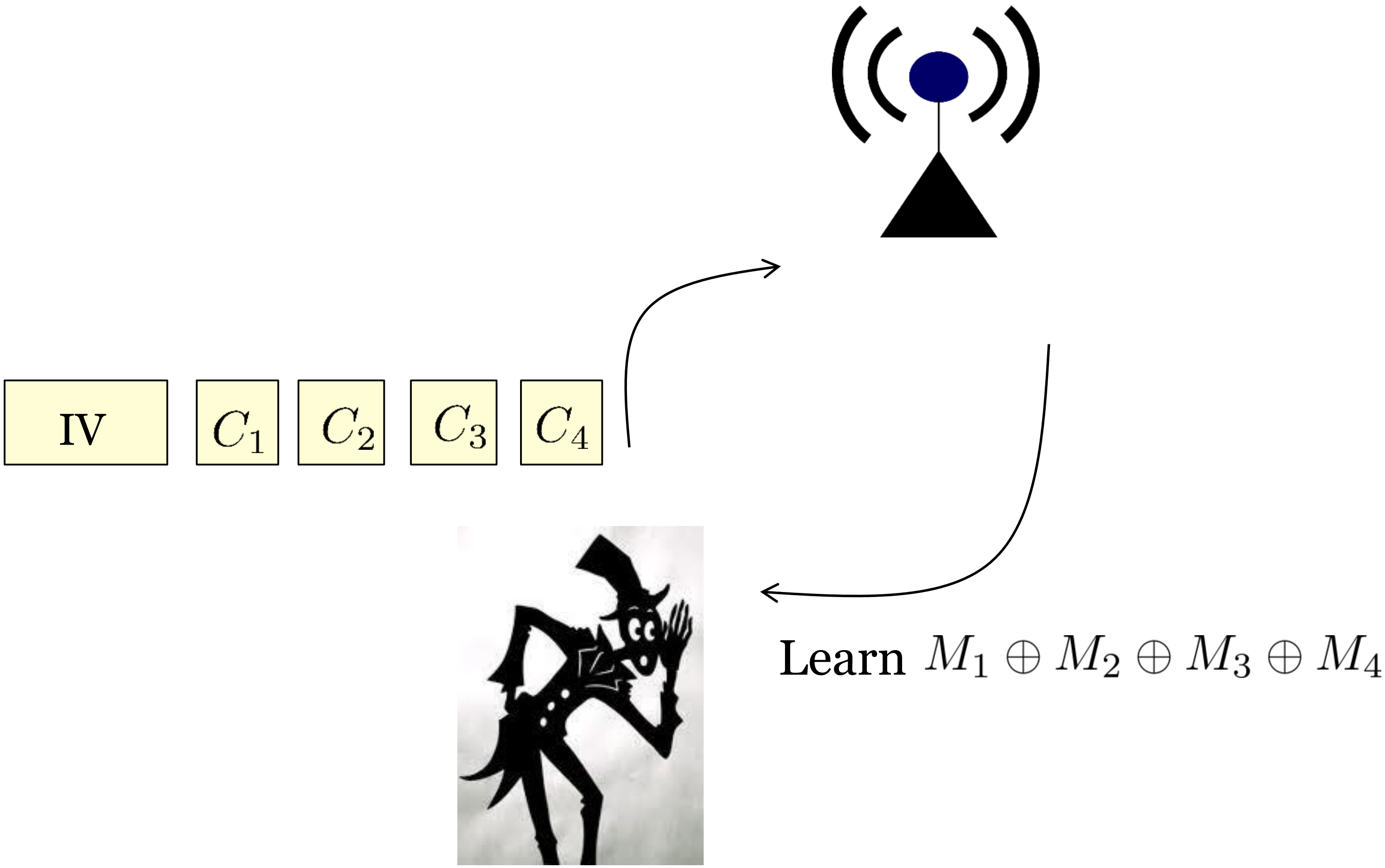
If valid, $M_1 \oplus M_2 \oplus M_3 \oplus M_4 = 0$

Exploit Decryption Response

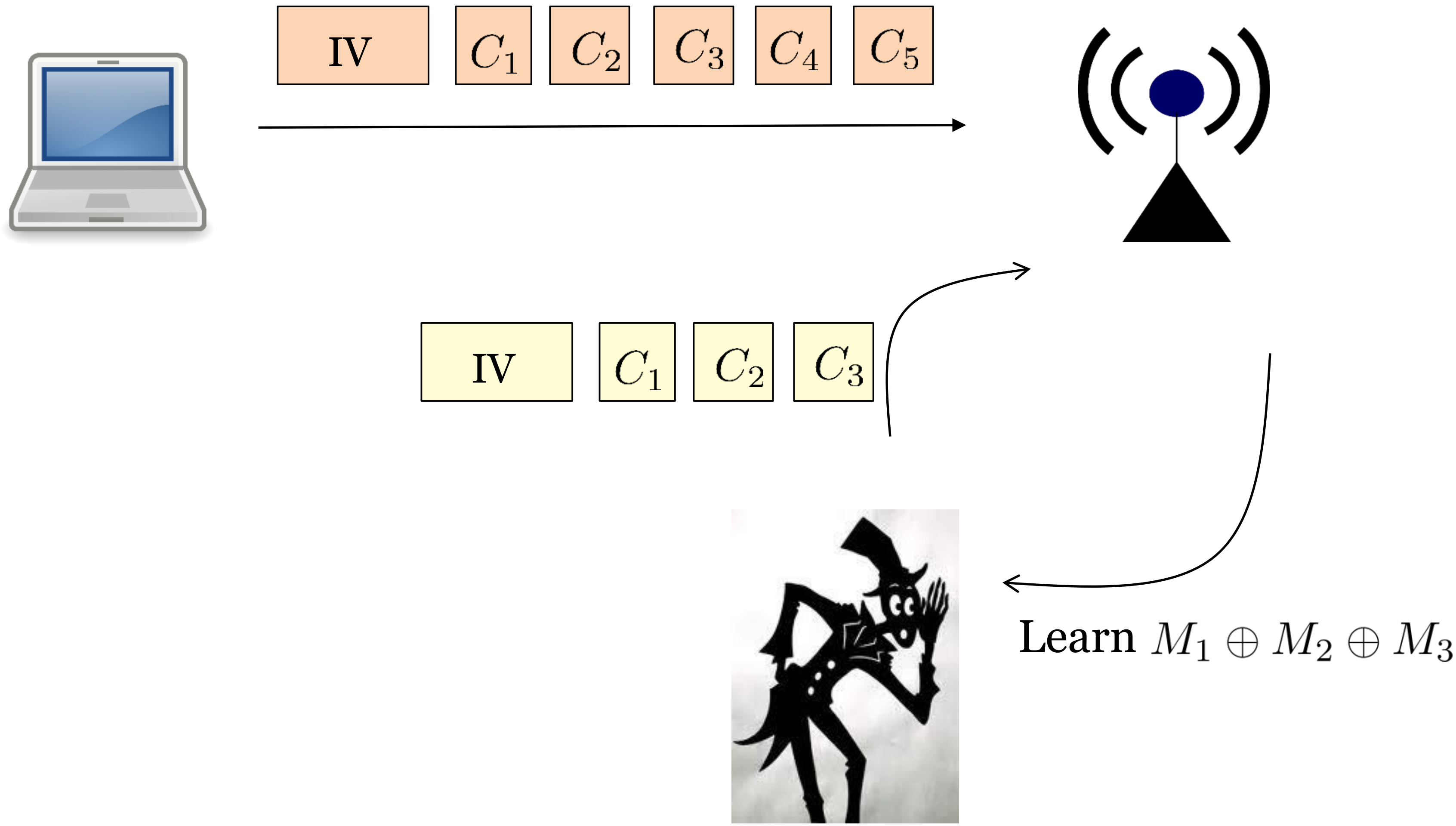


If invalid, $M_1 \oplus M_2 \oplus M_3 \oplus M_4 = 1$

Exploit Decryption Response



Exploit Decryption Even Further



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

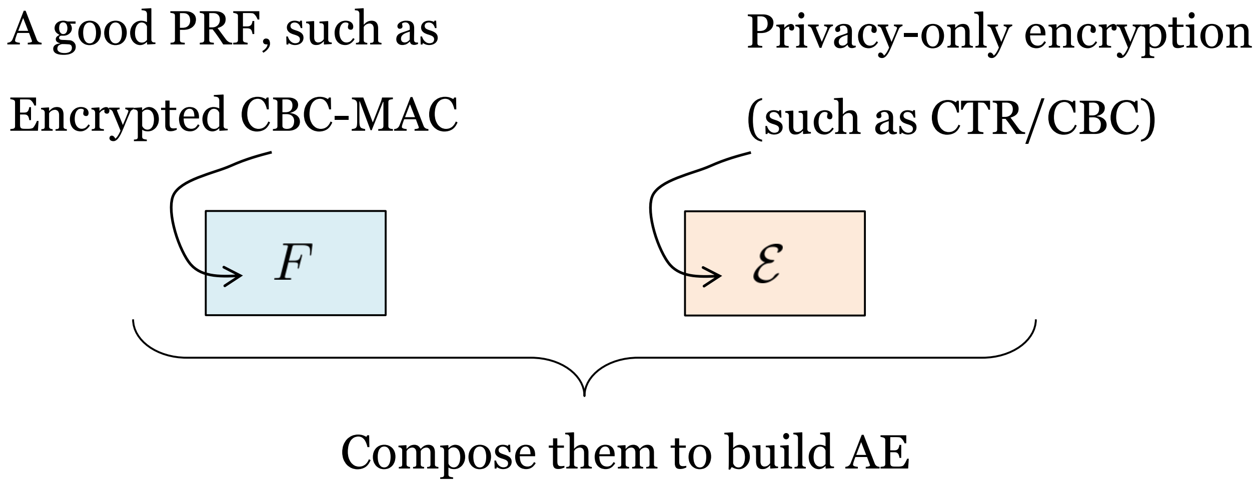
1. AE and Its Security Definitions

2. Failed Ways to Build AE

3. **Generic Compositions**

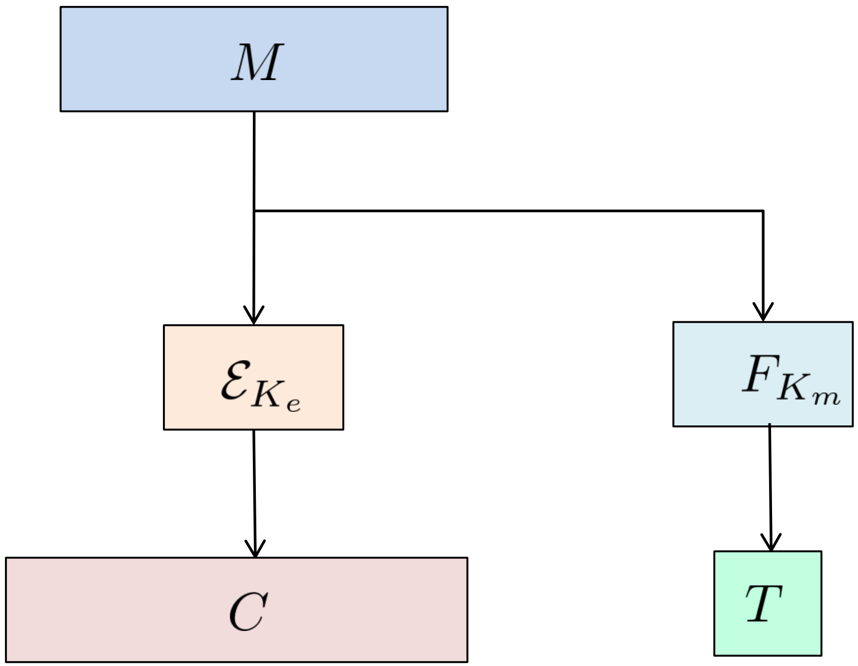
4. Padding-Oracle Attack on SSL/TLS

Constructing AE: Generic Composition



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

Encrypt-and-MAC: Simple Composition



Privacy	Authenticity
No	No

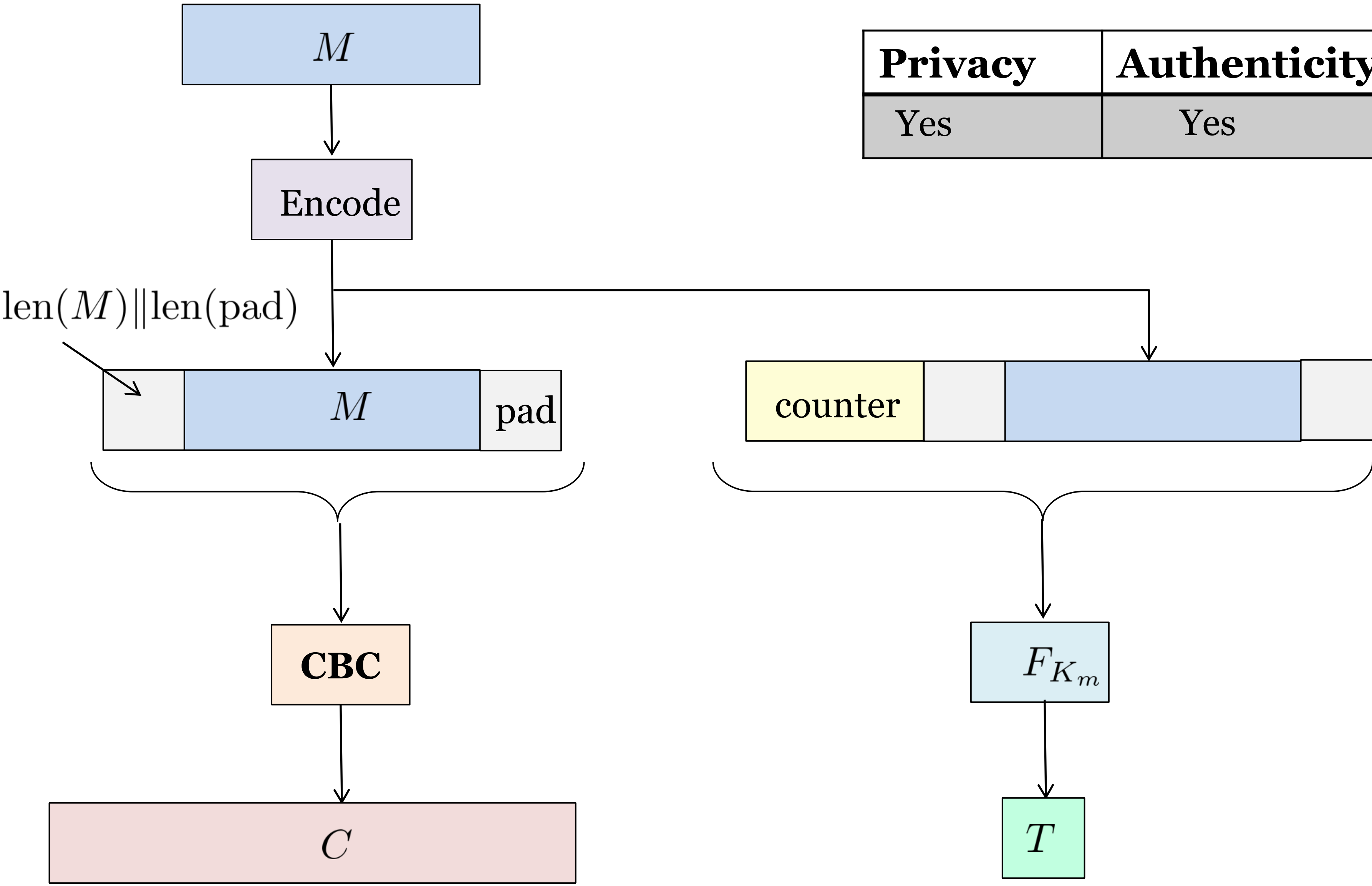
for some bad encryption scheme

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

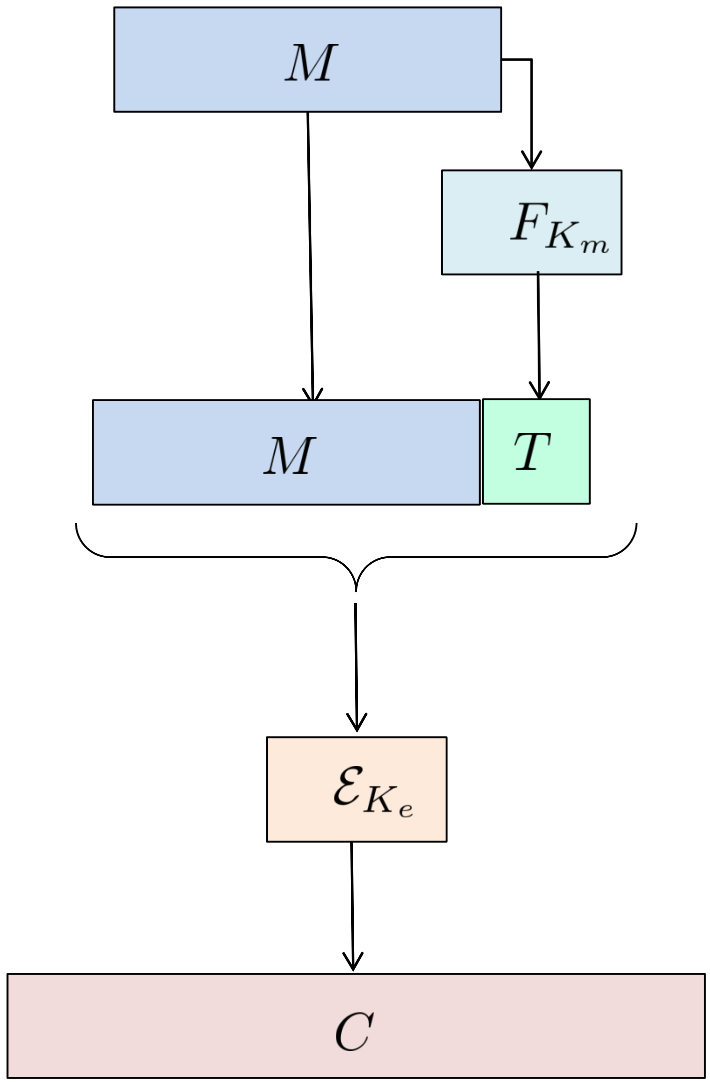
No authenticity if one can modify C such that decryption is unchanged.

Encrypt-and-MAC in SSH

Privacy	Authenticity
Yes	Yes



MAC-then-Encrypt

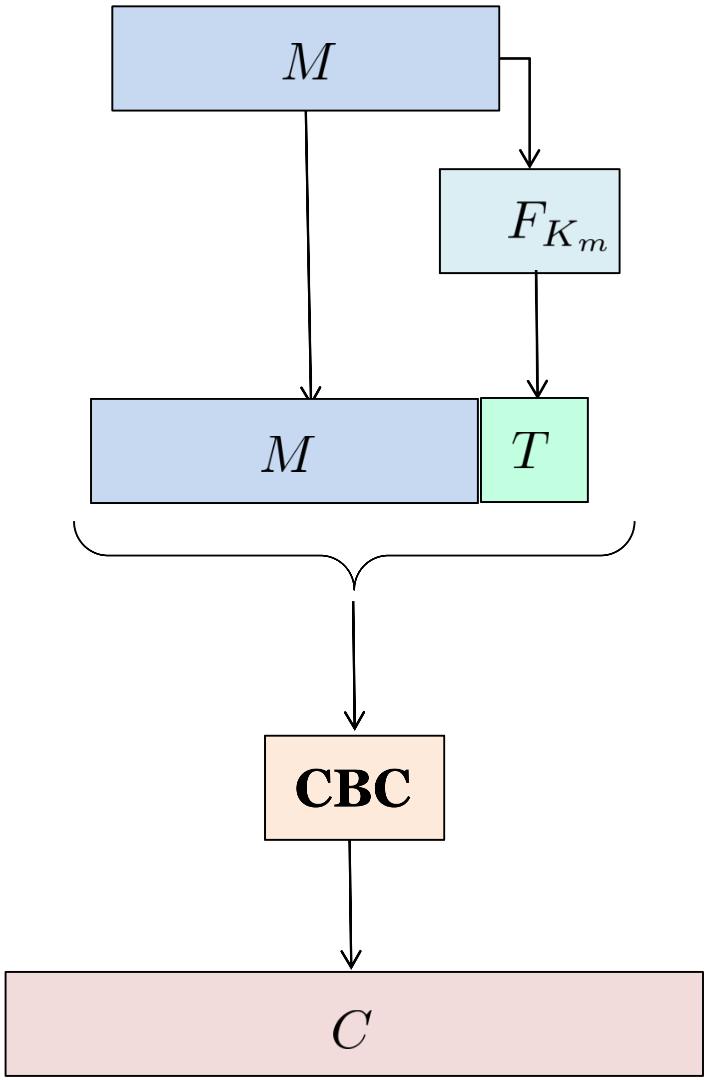


Privacy	**Authenticity**
Yes	No

for some bad encryption scheme

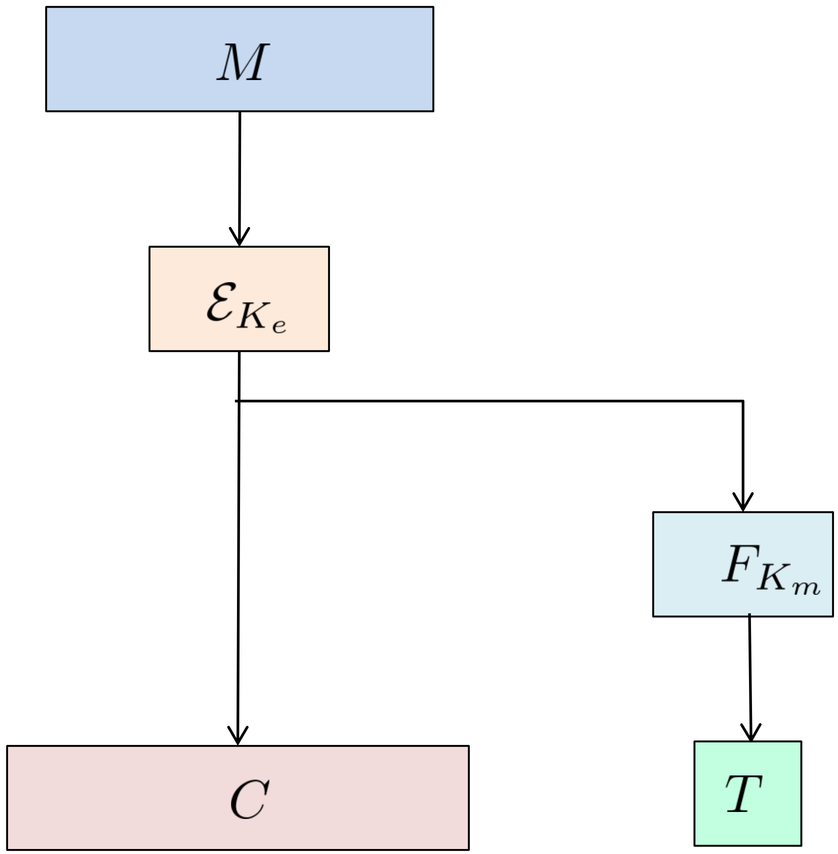
No authenticity if one can modify C such that decryption is unchanged.

MAC-then-Encrypt in TLS



Privacy	Authenticity
Yes	Yes

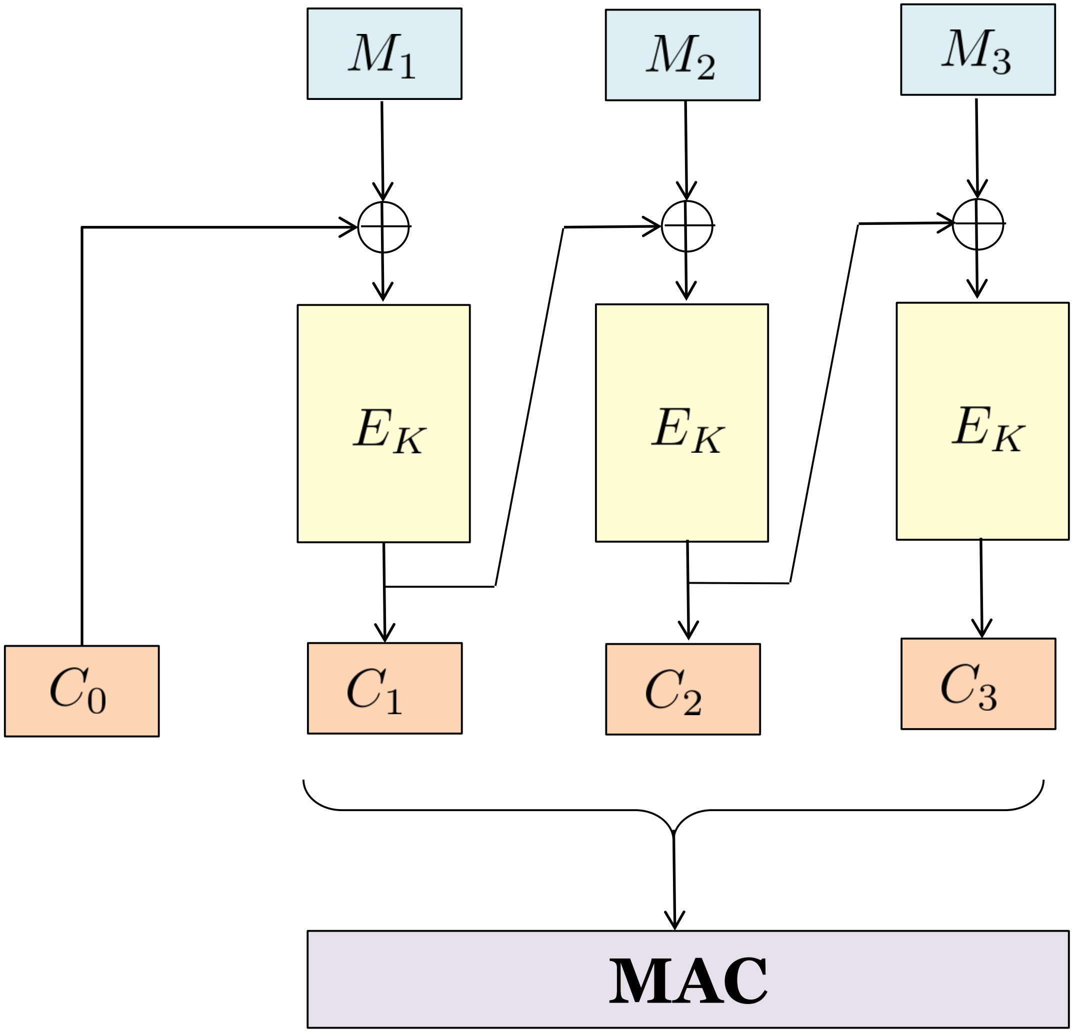
Encrypt-then-MAC



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

Agenda

1. AE and Its Security Definitions

2. Failed Ways to Build AE

3. Generic Compositions

4. **Padding-Oracle Attack on SSL/TLS**

The Padding-Oracle Attack

"Lucky Thirteen" attack snarfs cookies protected by SSL encryption

Exploit is the latest to subvert crypto used to secure Web transactions.

by D

Meaner POODLE bug that bypasses TLS crypto bites 10 percent of websites

Some of the world's leading sites are vulnerable to an easier, more simplified attack.

by D

Researchers poke hole in custom crypto built for Amazon Web Services

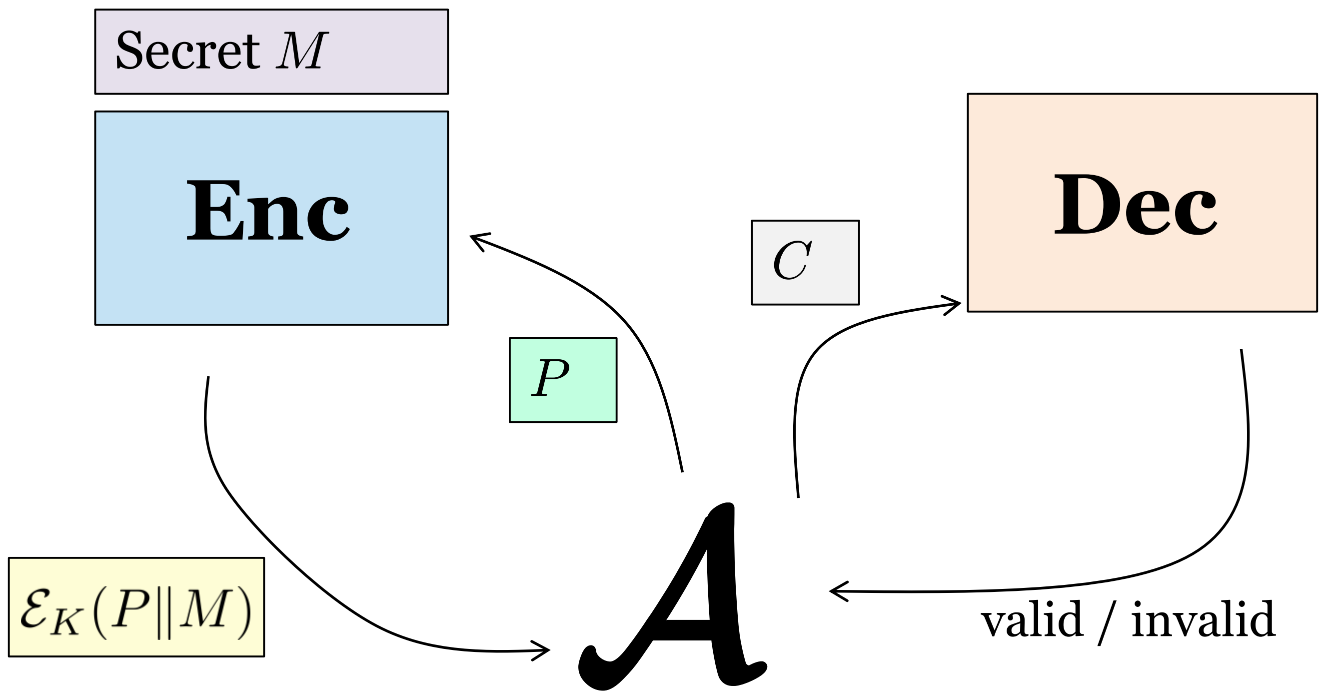
Even when engineers do everything by the book, secure crypto is still hard.

by D

New TLS encryption-busting attack also impacts the newer TLS 1.3

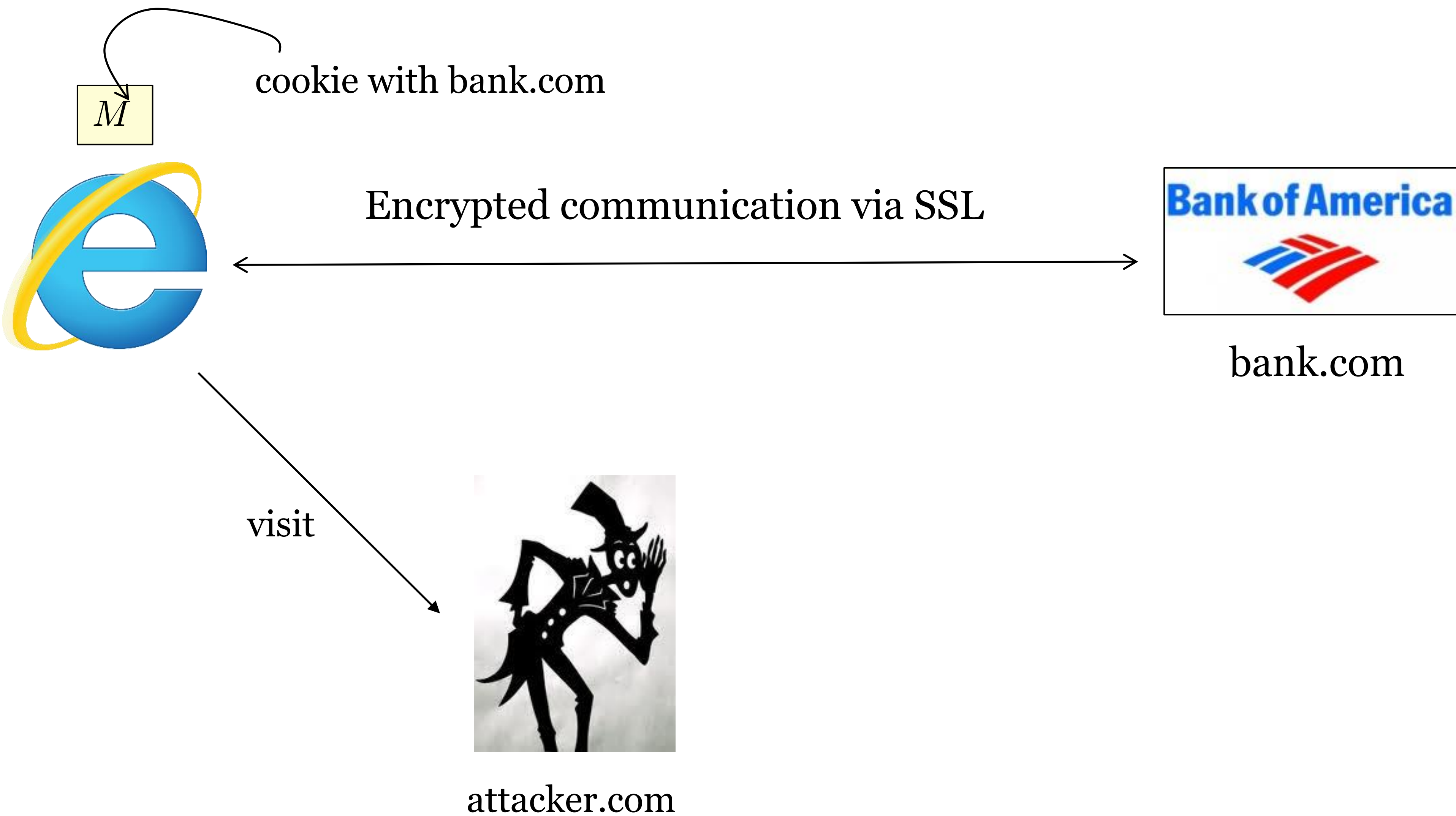
Researchers discover yet another Bleichenbacher attack variation (yawn!).

Attack Model: Chosen Prefix Secret Suffix

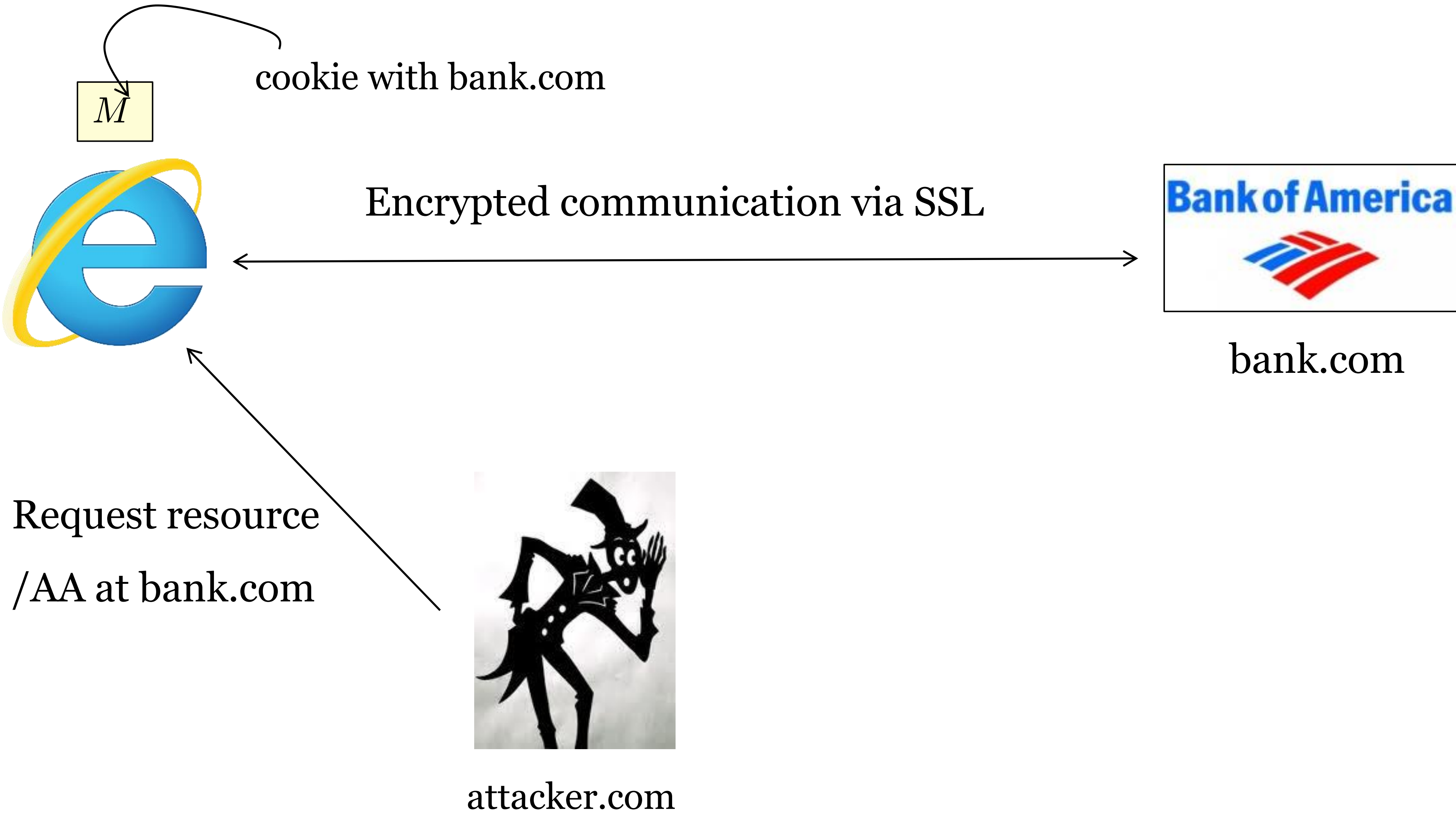


Goal: Recover M

This Model Is Realistic: Attacking SSL



This Model Is Realistic: Attacking SSL



This Model Is Realistic: Attacking SSL

M

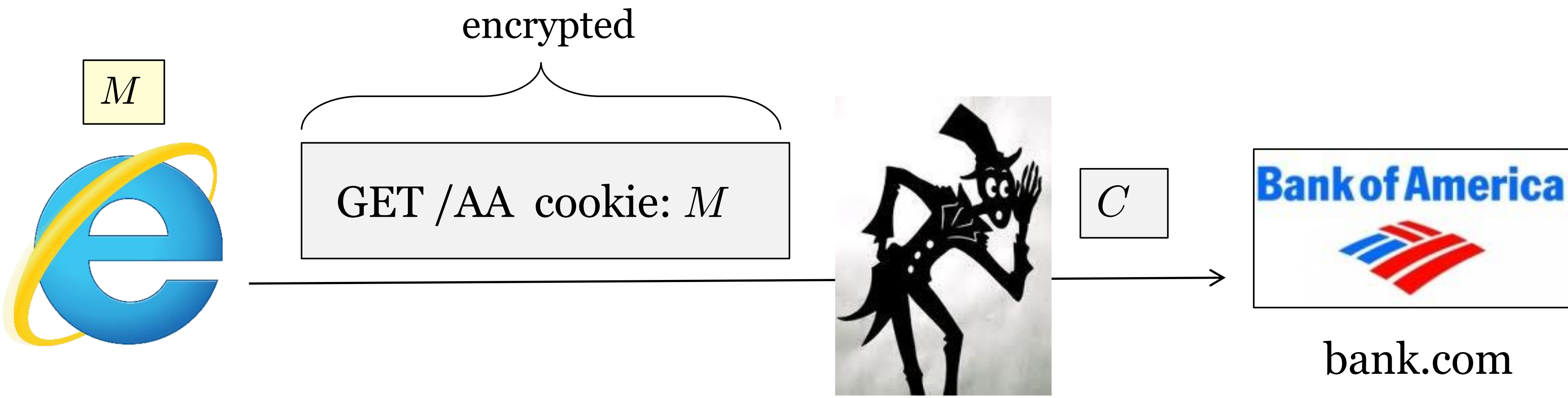
encrypted

GET /AA cookie: M

bank.com

attacker.com

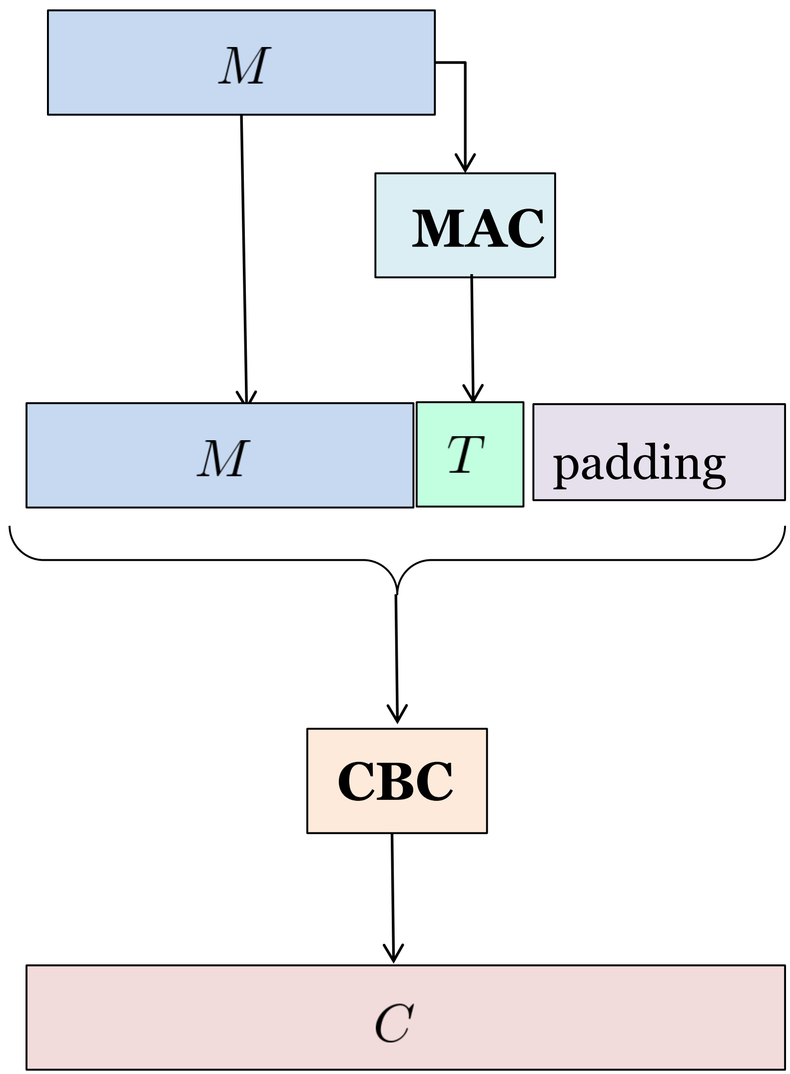
This Model Is Realistic: Attacking SSL



Enc oracle

Dec oracle

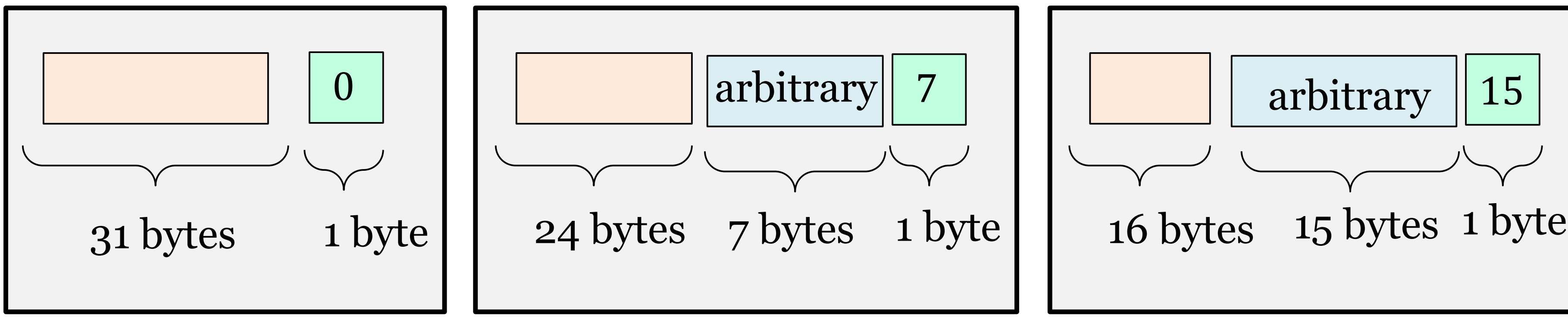
Encryption In SSL: MAC-then-Encrypt



Padding In SSL Encryption

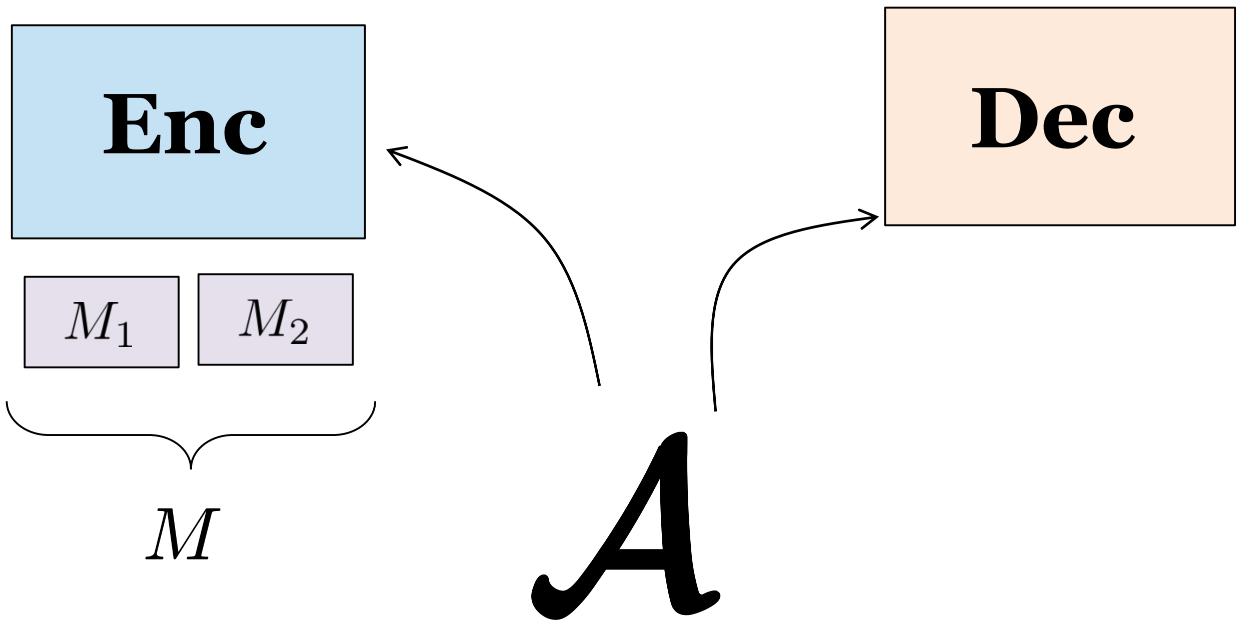
block length is 16 bytes

Consider byte strings only



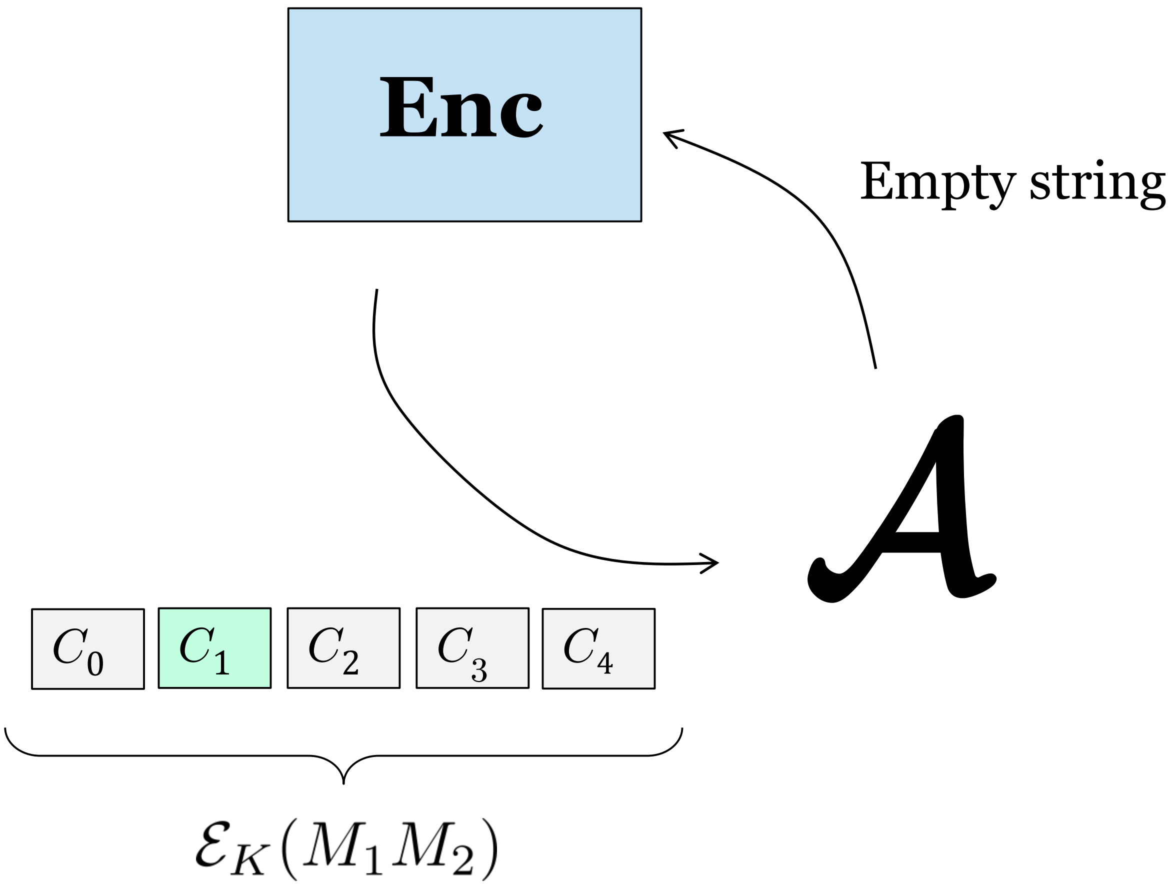
The Attack in Action

Illustration For Two-block Message

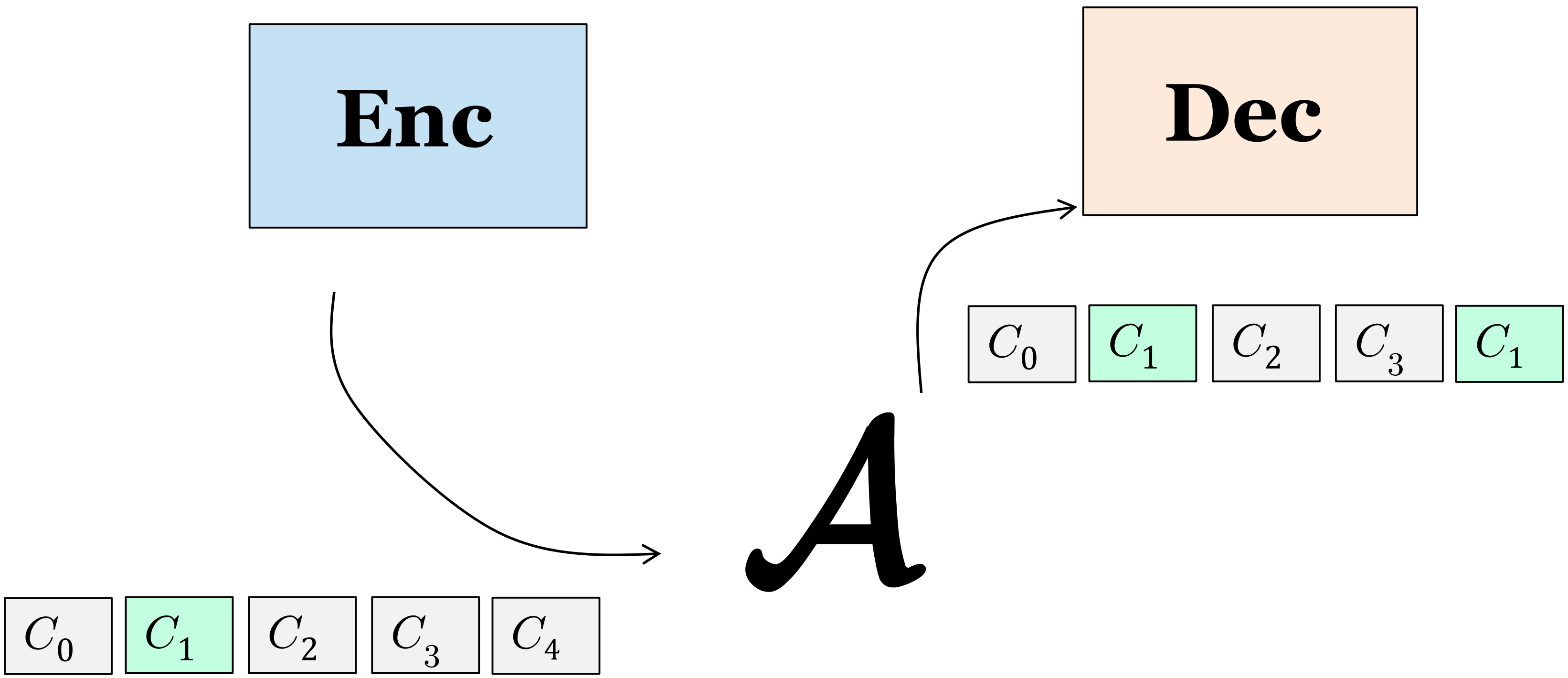


Aim: Recover the message byte by byte

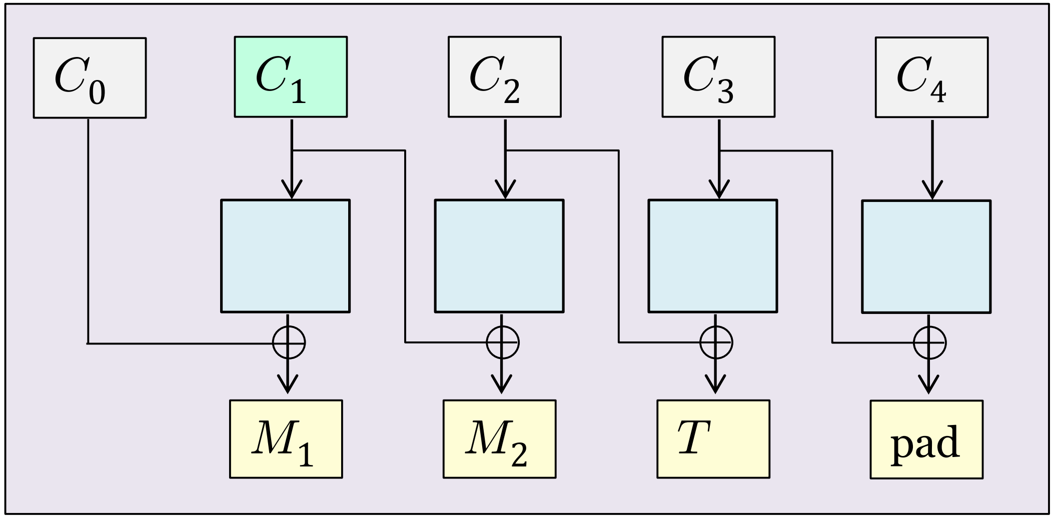
Recover Last Byte of First Block



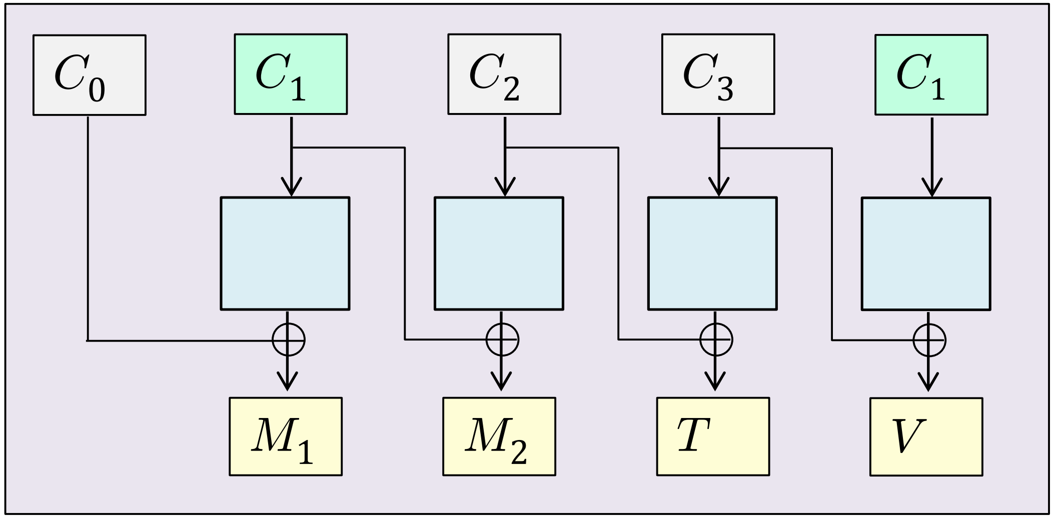
Recover Last Byte of First Block



CBC Decryption



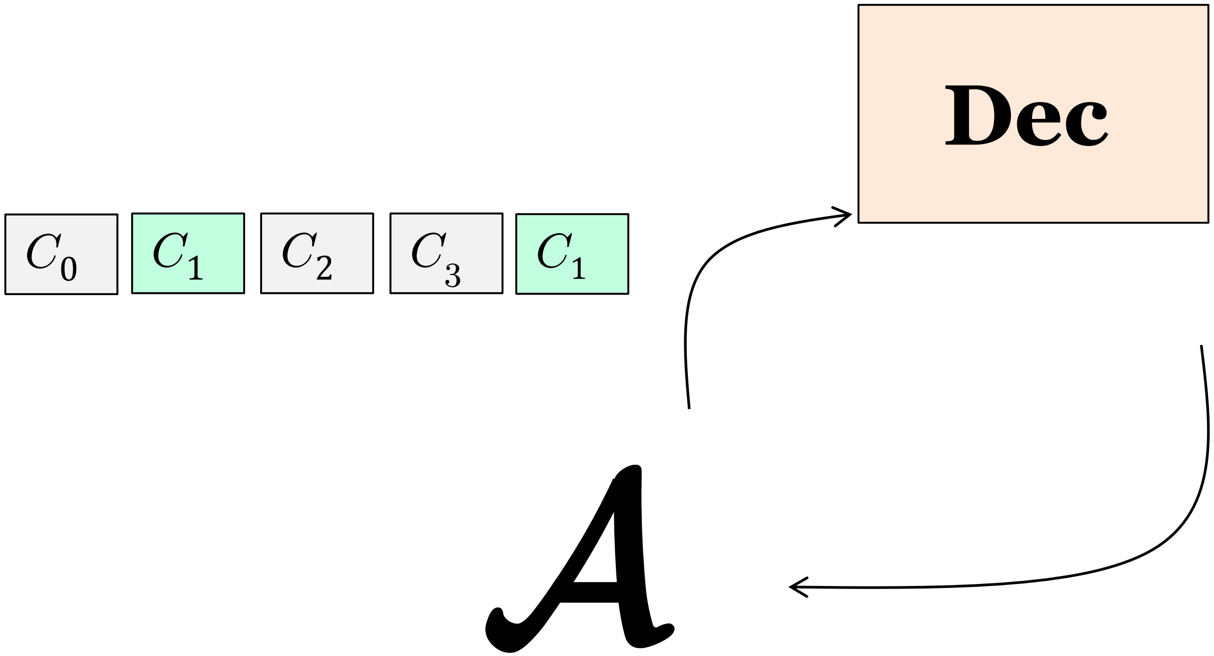
$$V = M_1 \oplus C_0 \oplus C_3$$



To pass MAC check, want the last byte of V to be 15

Pass with prob ~ 1/256

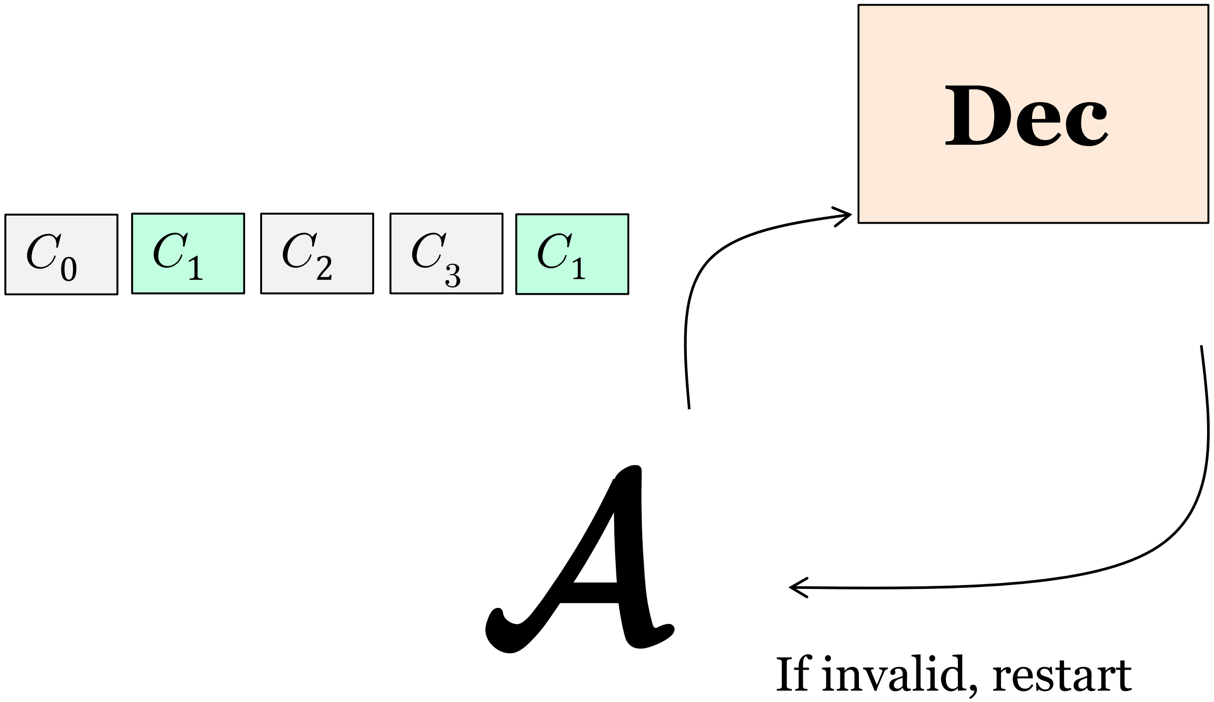
Exploit Decryption Output



If valid, last byte of $V = M_1 \oplus C_0 \oplus C_3$ is 15

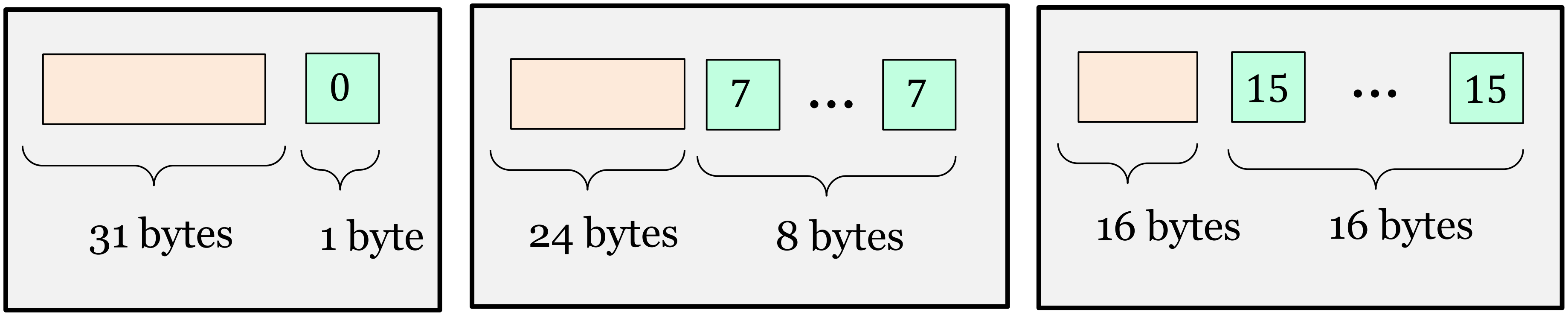
Learn last byte of M_1

Exploit Decryption Output



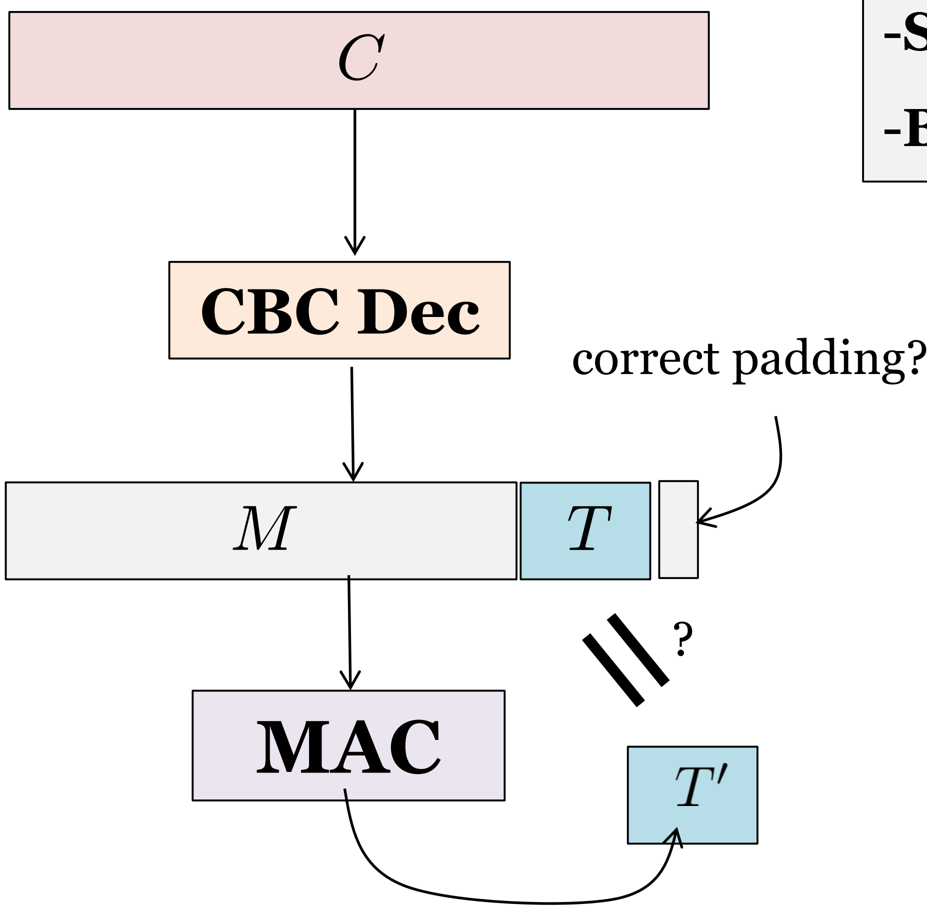
After t attempts, succeed with prob $\sim 1-(1-1/256)^t$ times

Patching Via Different Padding



Secure if implement properly

Careless Implementation Leads To Attack



-**Secure** if return a single error signal

-**Broken** if tell what kind of error it is.

Scanning For Vulnerable Implementations

"Given final block not properly padded"

Spring boot | Eclipse | PKCS12 | Javax | Java | Intellij | Android Studio | Images | AES

About 16,000 results (0.32 seconds)

Stack Overflow
https://stackoverflow.com › questions › given-final-bl...

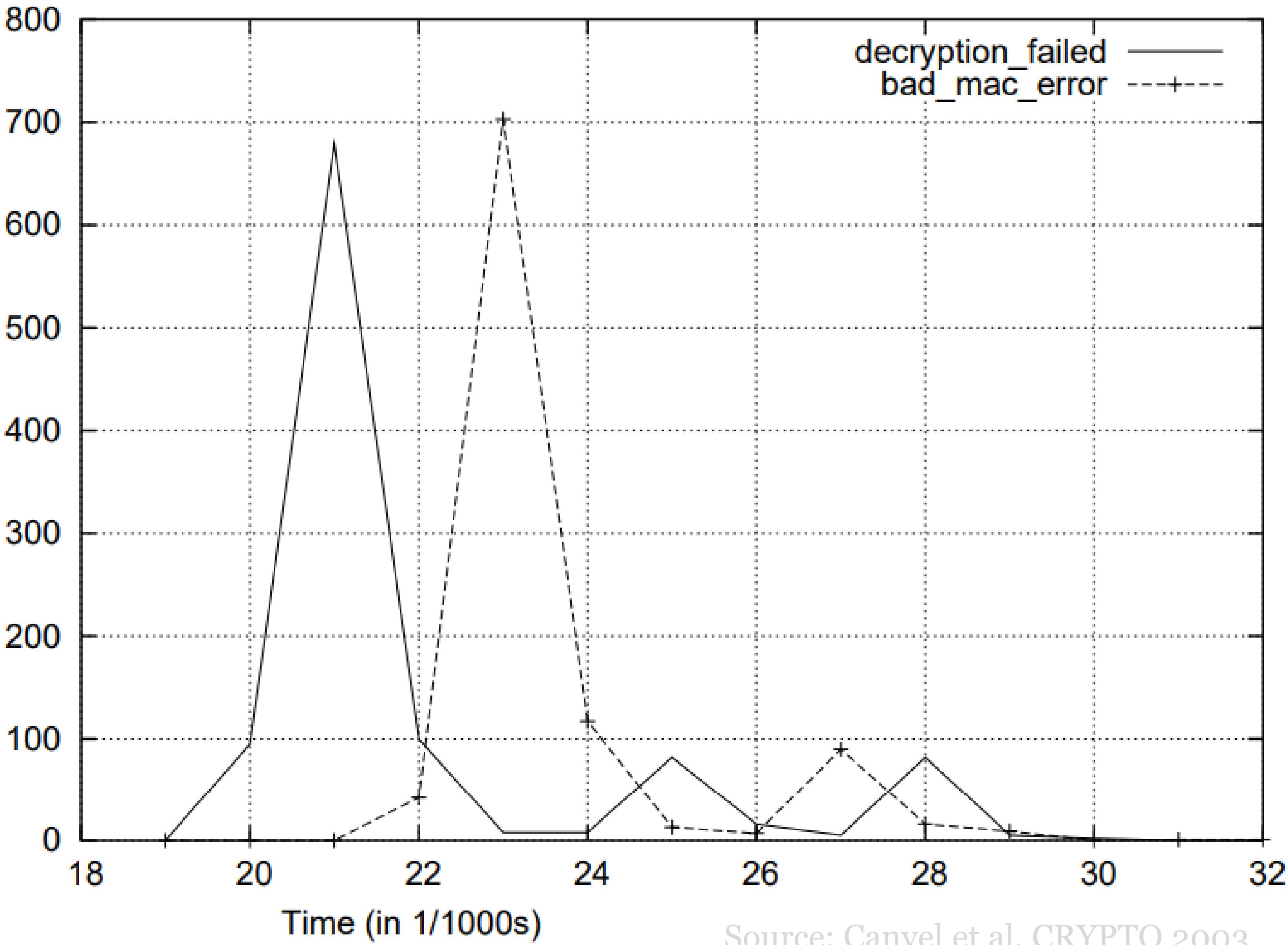
Given final block not properly padded - java

Nov 8, 2011 — BadPaddingException: **Given final block not properly padded**. Such issues can arise if a bad key is used during decryption.

7 answers · Top answer: If you try to decrypt PKCS5-padded data with the wrong key, and then ...

Given final block not properly padded exception - Stack Overflow — Apr 11, 2018
Given final block not properly padded. AES Decryption - Stack ... — Nov 13, 2022
Given final block not properly padded. Such issues can arise if ... — Jul 10, 2020
"Get Key Failed: **Given final block not properly padded**" when I ... — Sep 22, 2021
More results from stackoverflow.com

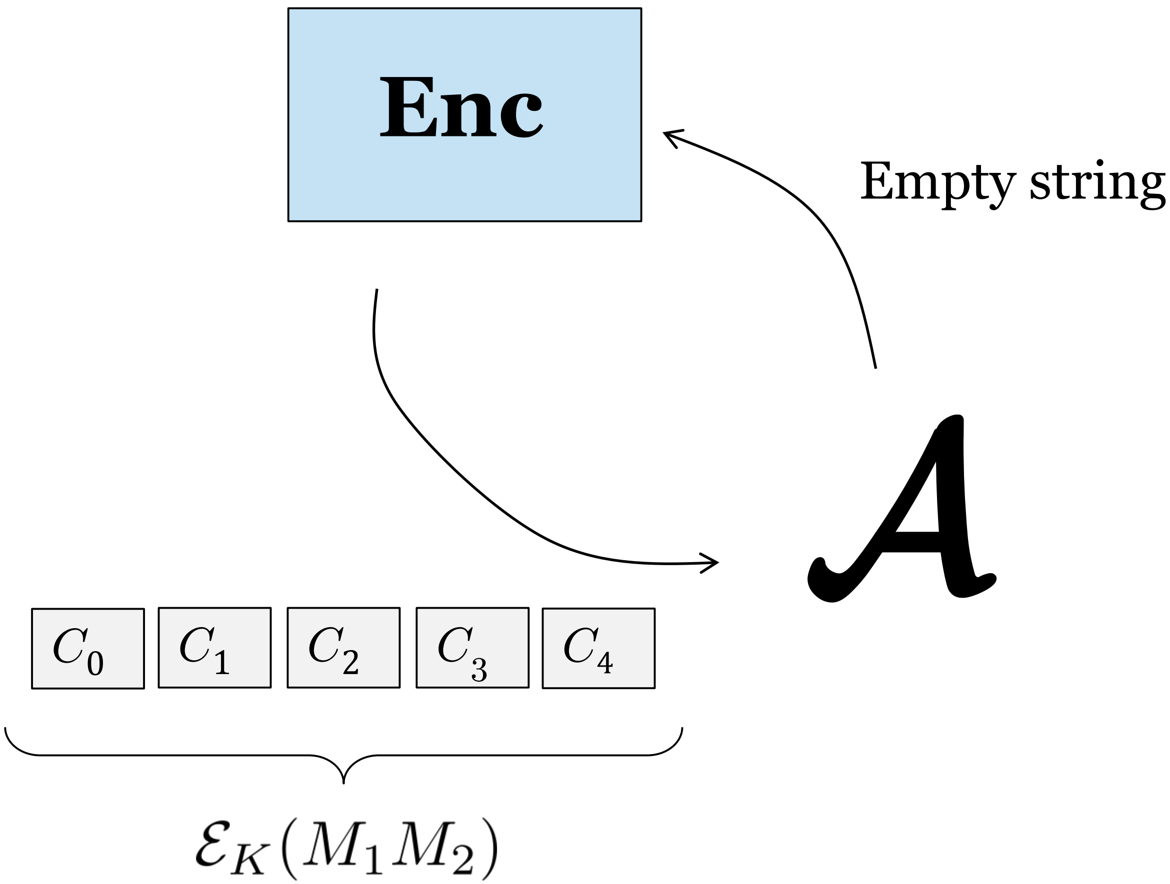
Implementation Is Hard: Timing Leakage



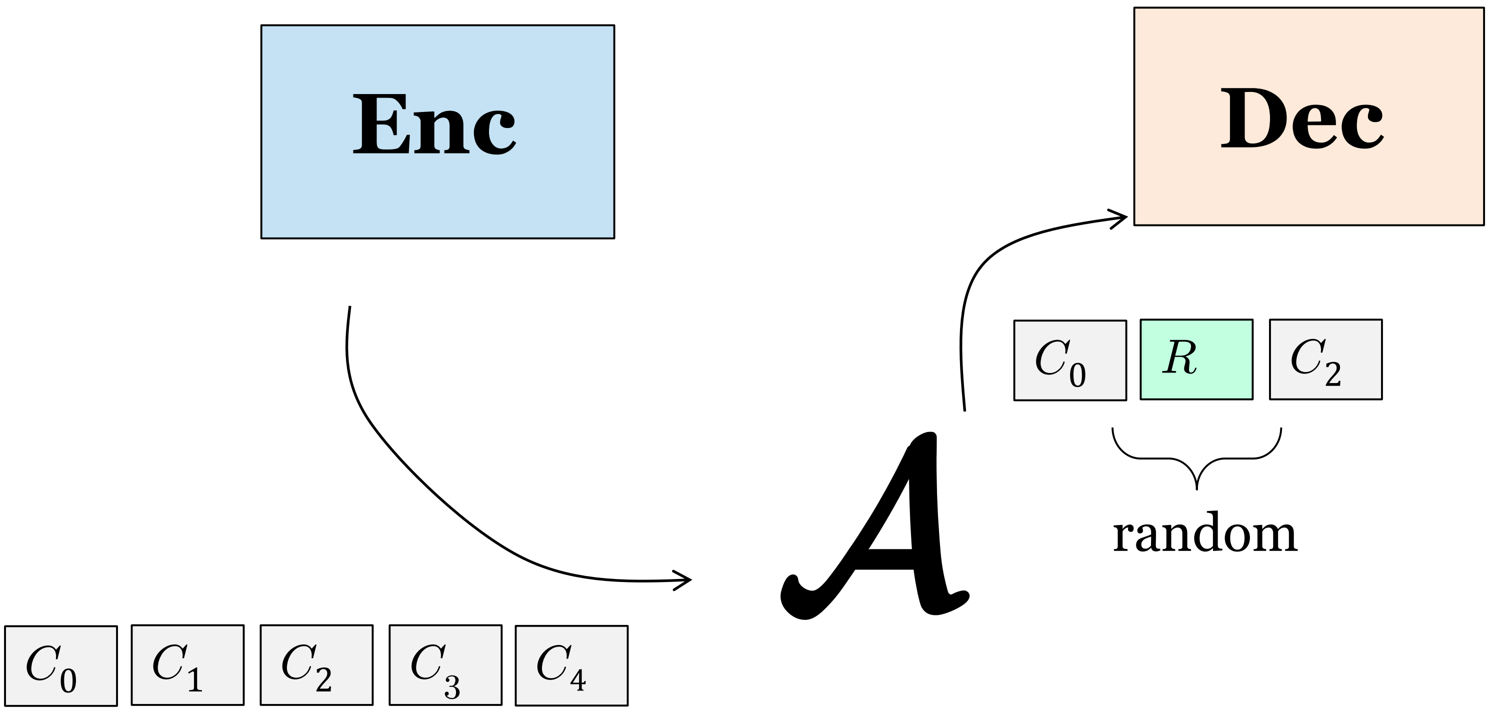
Source: Canvel et al, CRYPTO 2003

How To Attack

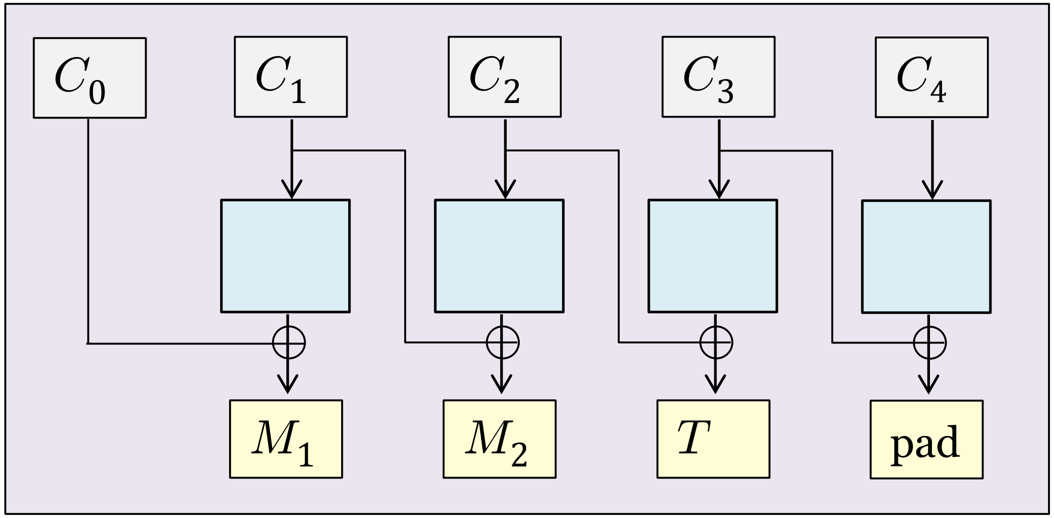
Illustration For Two-block Message



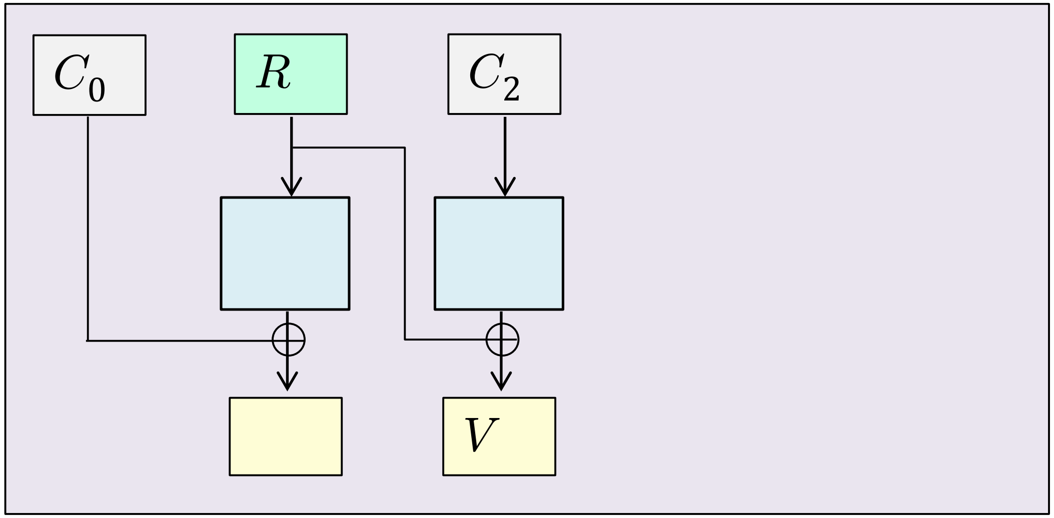
Recover Last Byte of Second Block



CBC Decryption



$$V = M_2 \oplus C_1 \oplus R$$



If V ends with a zero byte

Bad tag signal