Today in Cryptography (5830)

Review of modes of operation & active attacks
Message authentication
CBC-MAC
Attacks against bad CBC-MAC implementations
Variable-length secure CBC-MAC
Authenticated encryption

Malleability example: Encrypted cookies

http://startup.com

GET /index.html HTTP/1.1

HTTP/1.1 200 OK
Content-type: text/html
Set-cookie: sessionToken=ab35h013490...

Has secret key K

GET /info.html HTTP/1.1
Cookie: sessionToken=ab35h013490…

abc35h013490… = CTR-Mode(K, “admin=0”)

Malicious client can simply flip a few bits to change admin=1

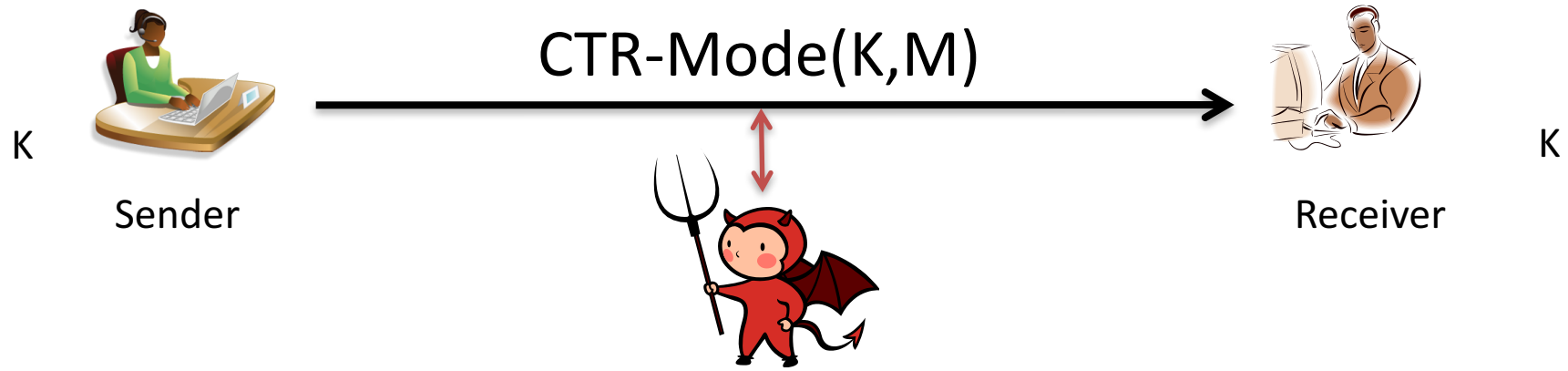
Review

- **Goal**: secure (length-extending) encryption
- What we have so far:
  - Block cipher modes of operation (CBC, CTR)
  - Insecurity against active attacks
    - Bit flip “mauling” attacks against CTR
    - Padding oracle attacks against CBC

- We need another tool:

  authenticity mechanisms

More generally:

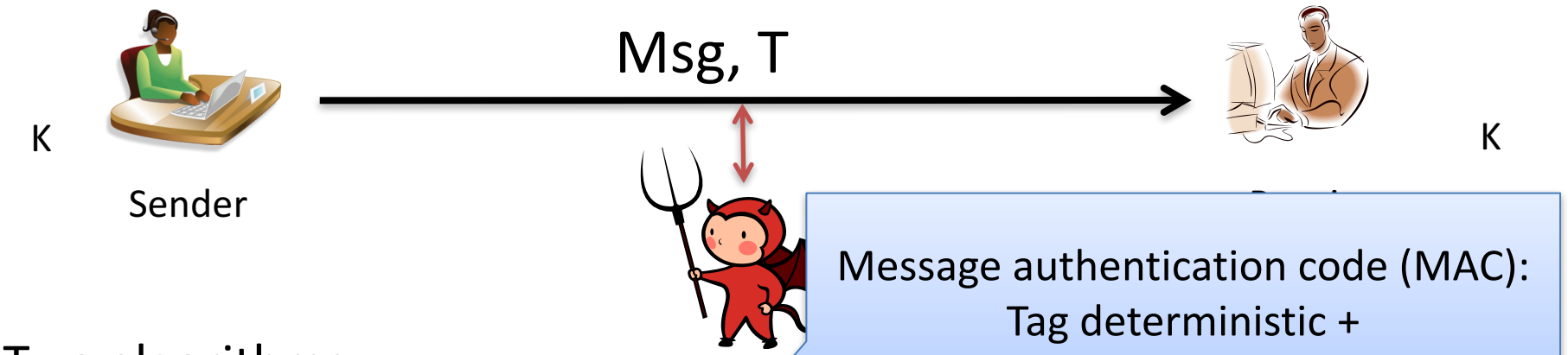


Attacker has read/write access to communications channel

<u>The strategy:</u>

Arrange so that that all bits received can be validated as having come from sender (the person with key K)

The tool: Message authentication schemes



Two algorithms:

(1) Tag(K,Msg) outputs a tag T

(2) Verify(K,Msg,T) outputs 0/1 (invalid / valid)

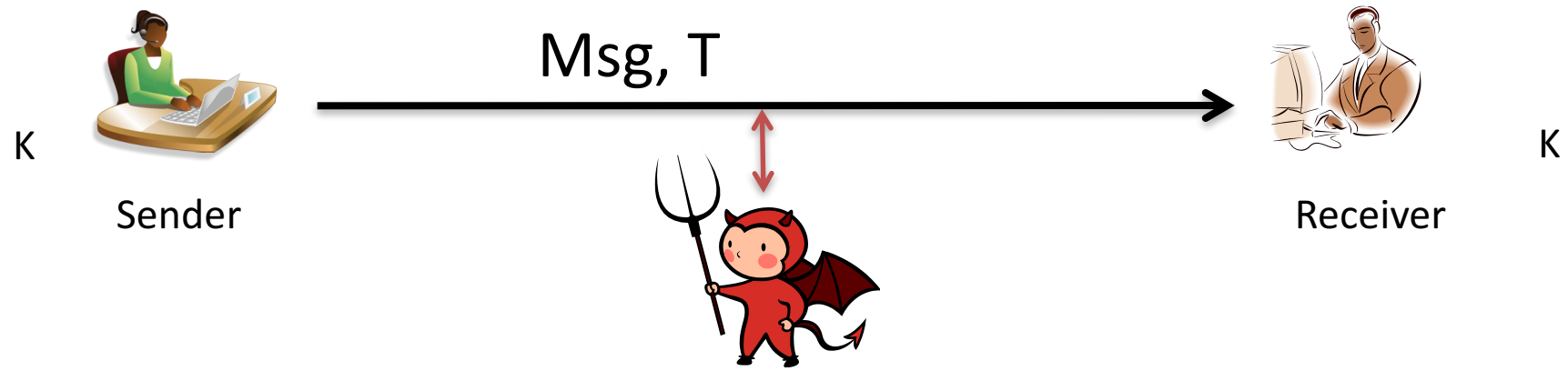
Correctness: Verify(K,Msg,Tag(K,Msg)) = 1 always

Security: No computationally efficient attacker can forge tags for a new message even when attacker gets

$(Msg_1 , T_1) , (Msg_2 , T_2), \ldots , (Msg_q , T_q)$

for messages of his choosing and reasonably large q.

Message authentication using pseudorandom functions (PRFs)



Let $F : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^n$ be a secure PRF

Tag(K,Msg)
Return F(K,Msg)

Verify(K,Msg,T):
If F(K,Msg) = T then Return 1
Return 0

Why is this secure?

What was example of a good PRF?

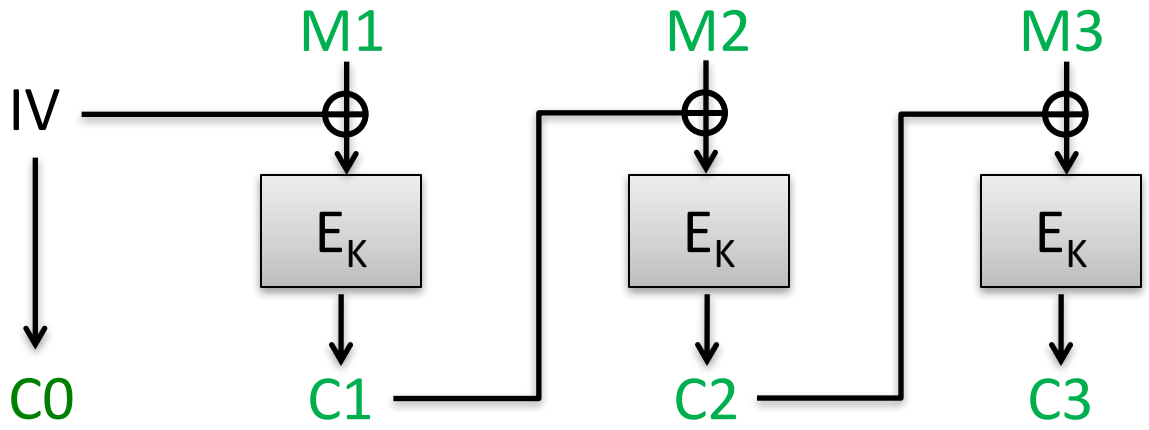
Recall CBC mode

Ciphertext block chaining (CBC)

Pad message M to M1,M2,M3,... where each block Mi is n bits

Choose random n-bit string IV

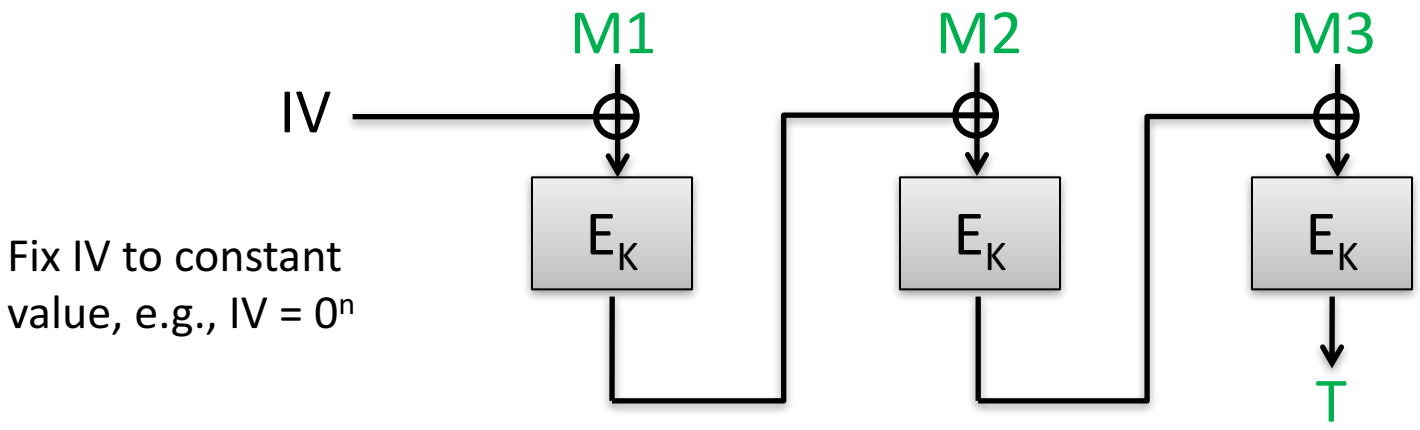
Then:



Can we convert this into variable-message-length PRF?

CBC-MAC

Message authentication code (MAC)

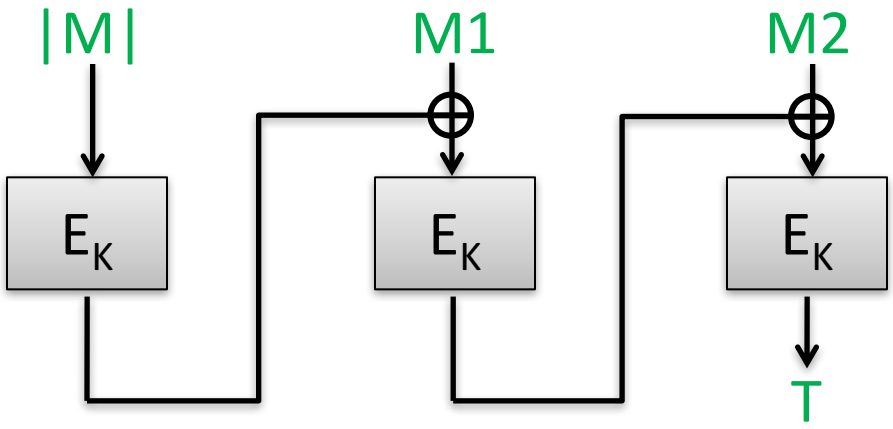


Turns out this is (provably) a good PRF

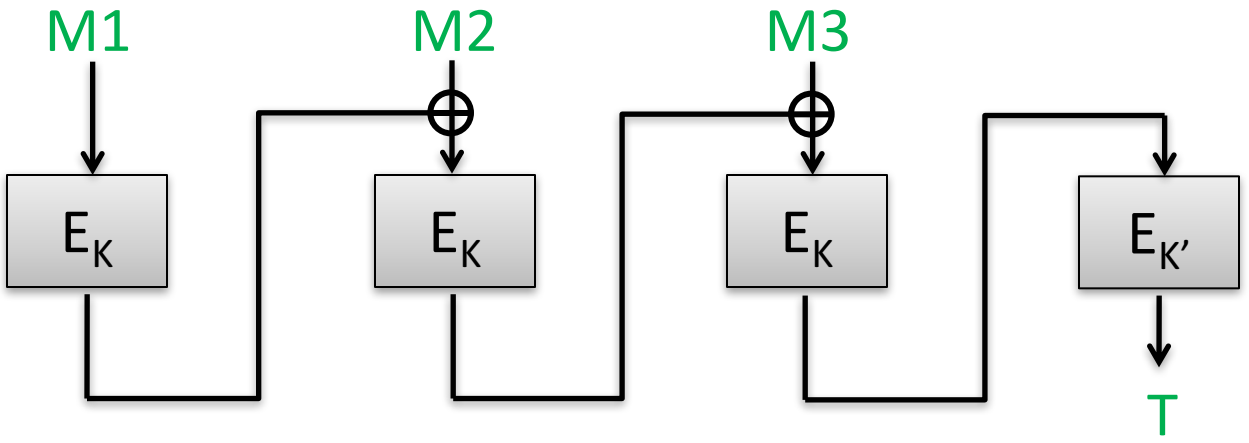
if K used only on same-length messages

Variable-message-length CBC-MAC

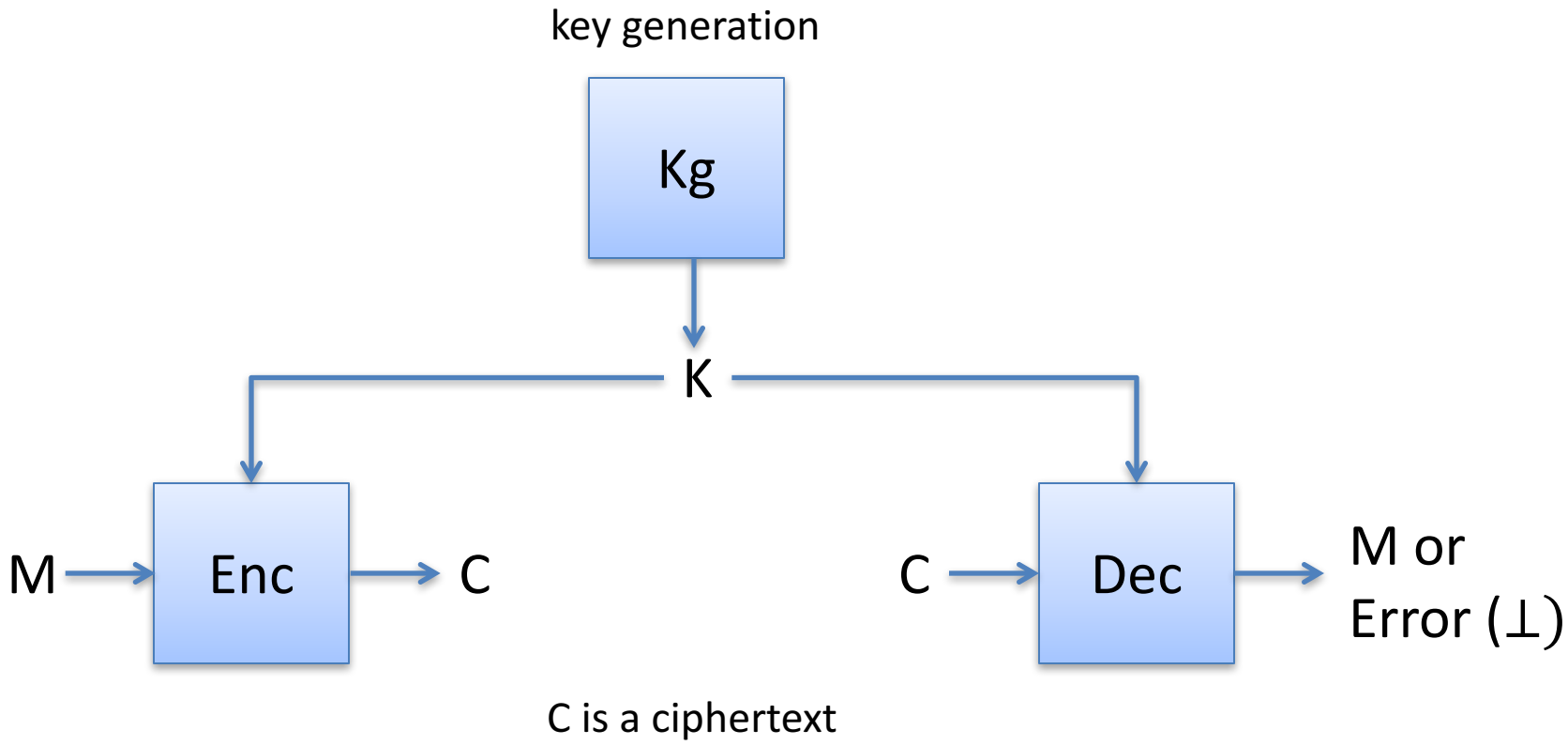
- Prepend message length



- Encrypted CBC-MAC

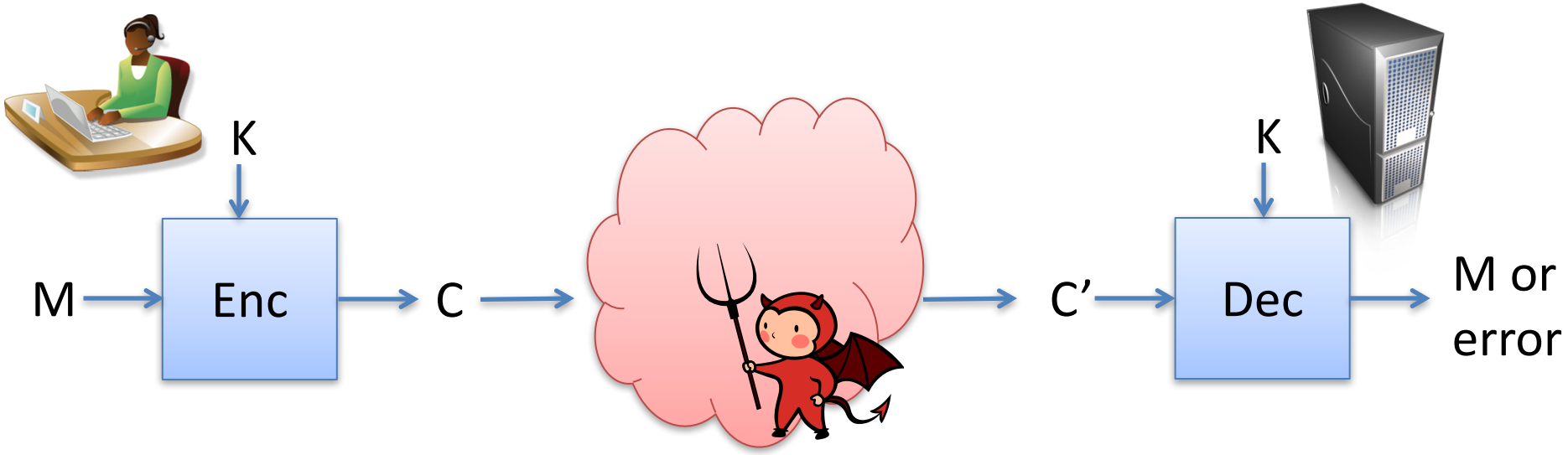


Authenticated encryption (AE)



Correctness: for all K, D(K , E(K,M)) = M with probability 1 over randomness used

Authenticated encryption (AE)



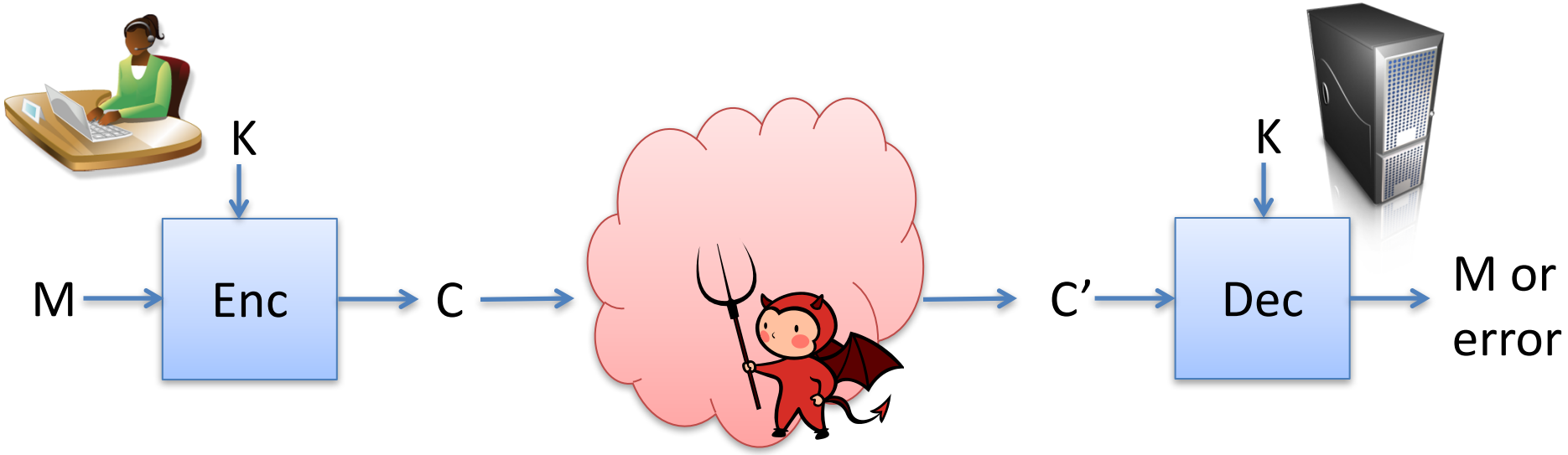
What security properties do we need from symmetric encryption?

1) Confidentiality: should not learn any information about M

2) Authenticity: should not be able to forge ciphertexts

Often referred to as Authenticated Encryption security

Authenticated encryption (AE)



Ciphertext unforgeability: Let K be honestly generated secret key. No computationally efficient attacker can construct ciphertext C* that decrypts correctly under K, even when given

$$(M_1 , C_1) , (M_2 , C_2), \dots , (M_q , C_q)$$

for messages of his choosing and ciphertexts generated under K. It must be that $C^* \neq C_i$ for $1 \leq i \leq q$

How do we do it?

Build a new scheme from Enc mode (CBC, CTR) and MAC

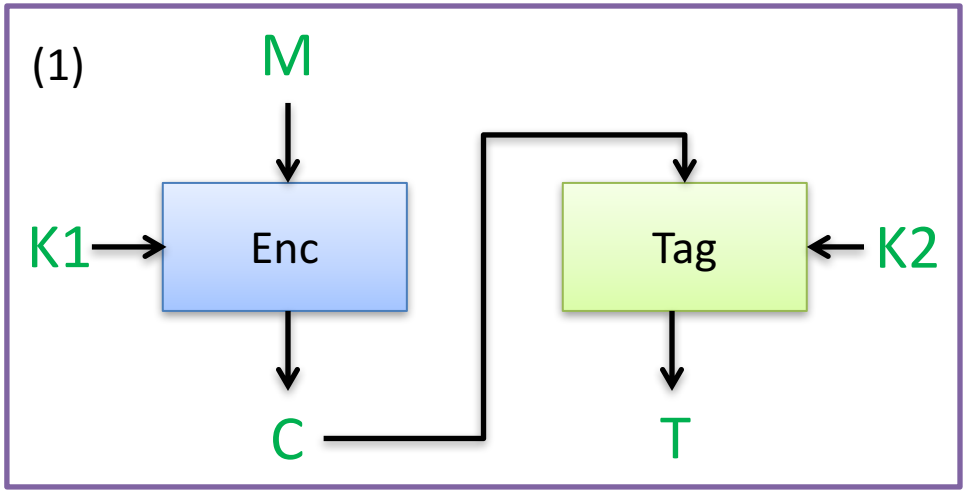
Kg outputs Enc key K1 and MAC key K2

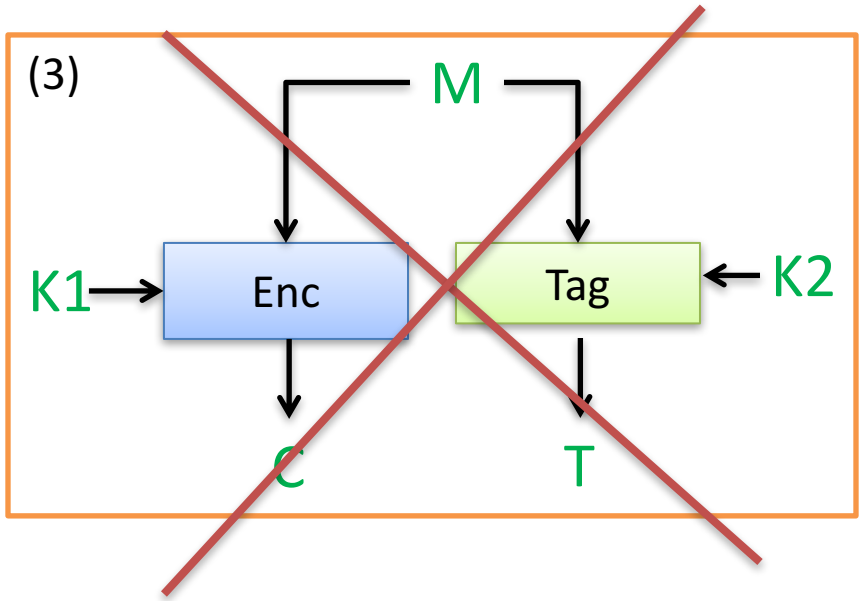
Several ways to combine:

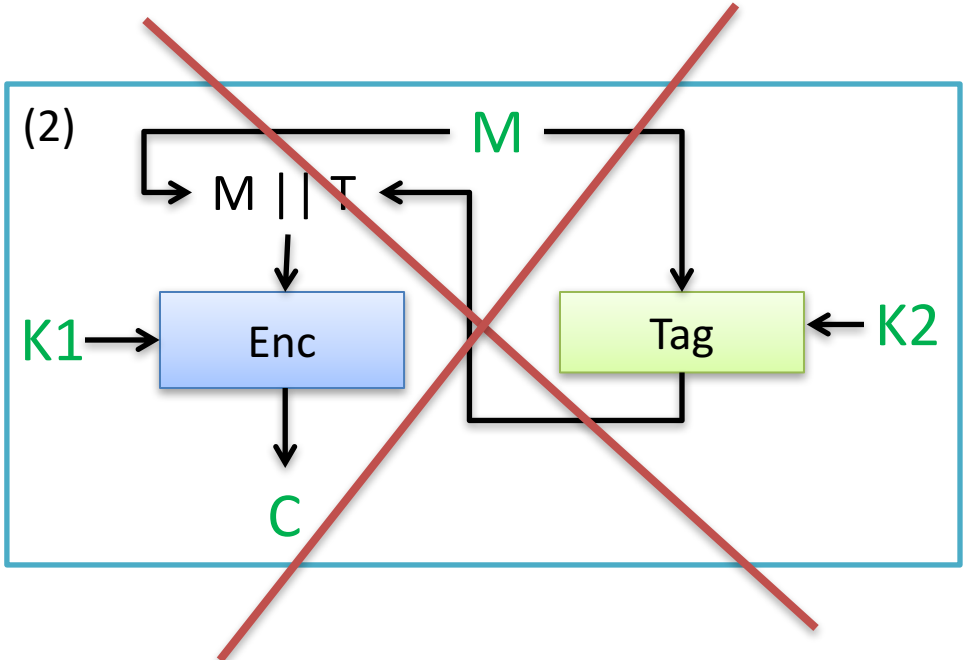
(1) encrypt-then-mac

(2) mac-then-encrypt

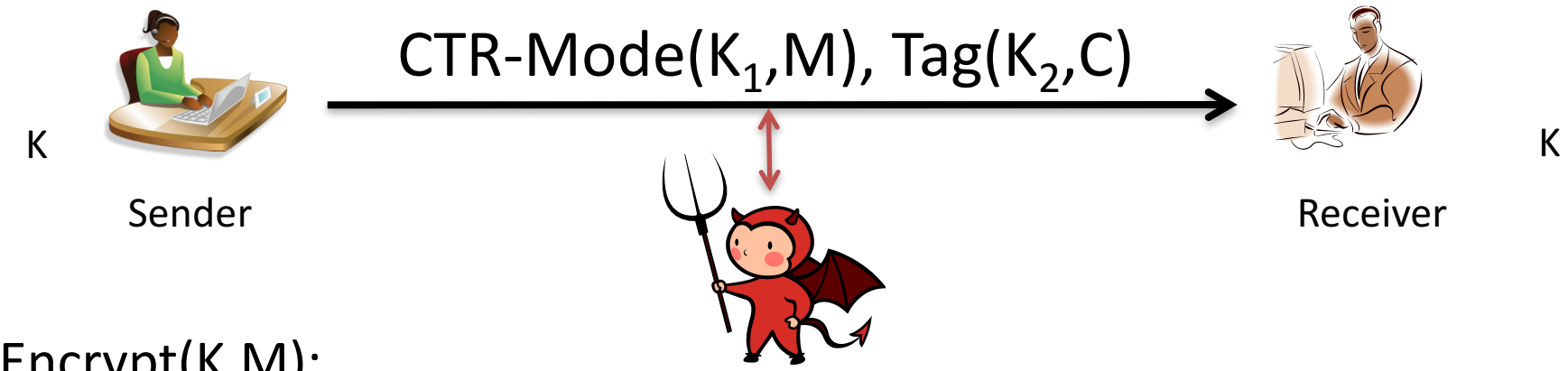
(3) encrypt-and-mac







Composing encryption and authentication



Encrypt(K,M):

Use secret keys K_1 and K_2 . These can be derived from K if needed

$K_1 = AES(K,0^n)$ $K_2 = AES(K,1^n)$

$C = CTR\text{-}Mode(K_1, M)$

$T = Tag(K_2,C)$

Output C||T

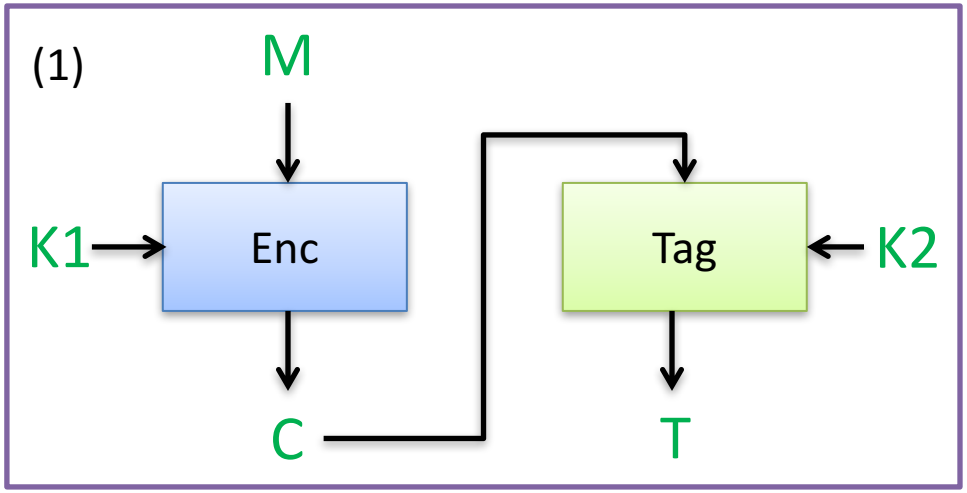
Decrypt(K,C||T)

If $Verify(K_2,C,T) \neq 1$ then Return error

Return $CTR\text{-}Mode(K_1,C)$

Build a new scheme from Enc mode (CBC, CTR) and MAC
Kg outputs Enc key K1 and MAC key K2

Several ways to combine:
(1) encrypt-then-mac
(2) mac-then-encrypt
(3) encrypt-and-mac

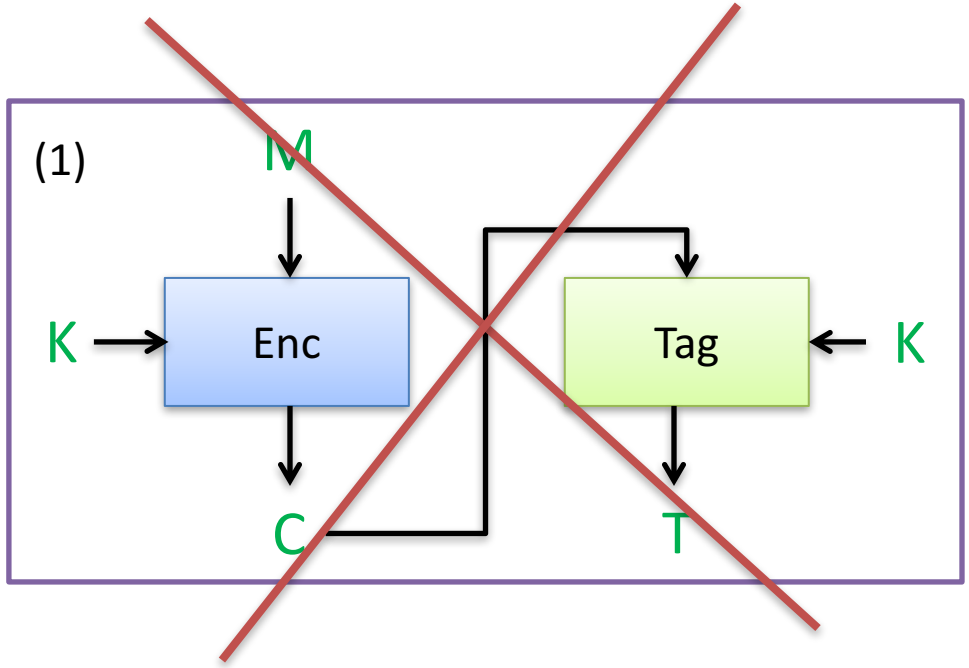


Thm. If encryption scheme provides confidentiality against passive attackers and MAC provides unforgeability, then Encrypt-then-MAC provides secure authenticated encryption

Key separation is essential

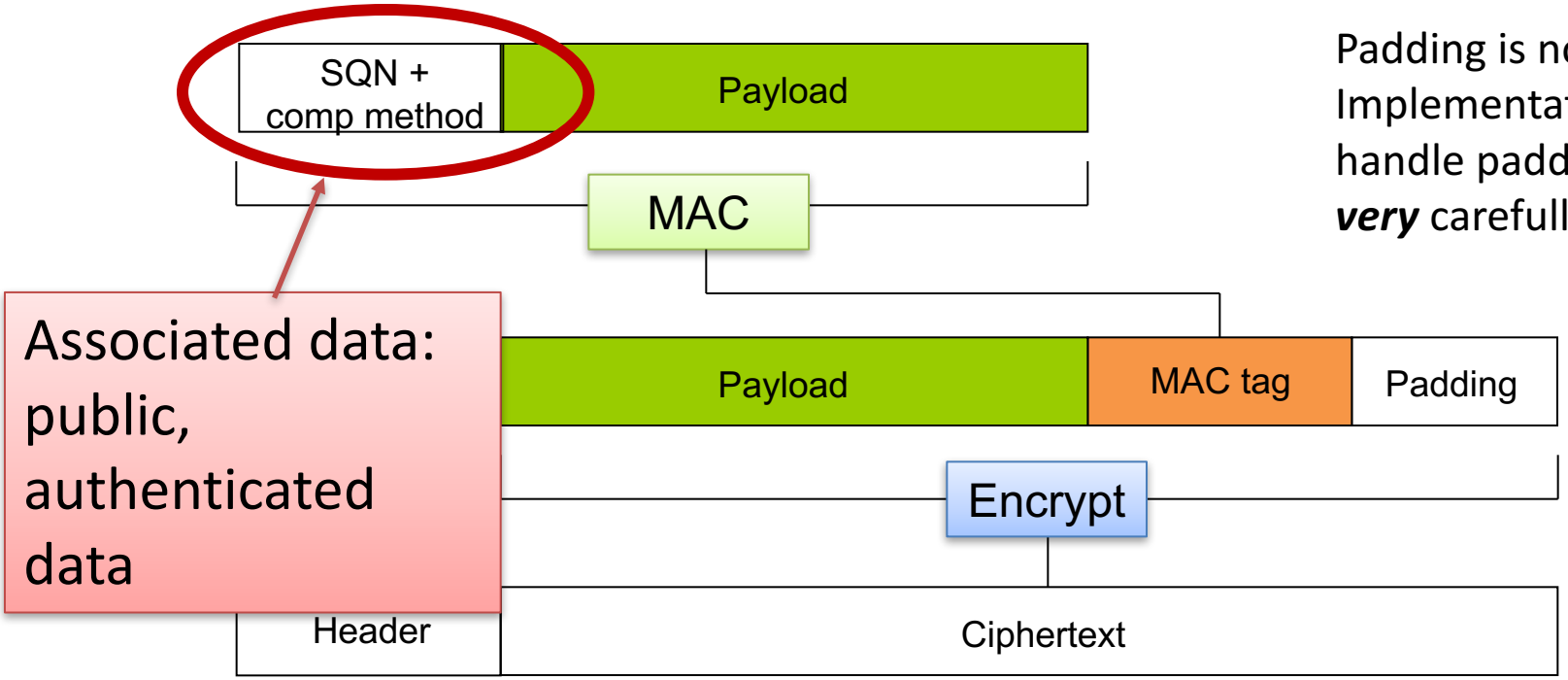
If one uses same key for both encryption and MAC, attacks can arise

Consider CBC-MAC plus CBC-mode encryption



General rule: different crypto primitives or different applications of same primitive, need independent keys

TLS 1.2 record protocol: MAC-Encode-Encrypt (MEE)



Padding is not MAC'd. Implementations must handle padding checks ***very*** carefully.

MAC	HMAC-MD5, HMAC-SHA1, HMAC-SHA256
Encrypt	CBC-AES128, CBC-AES256, CBC-3DES, RC4-128

TLS 1.3 using proper authenticated-encryption schemes

Dedicated authenticated encryption schemes

Not a generic composition of Enc, MAC.
Directly construct from blockcipher

Attack	Inventors	Notes
OCB (Offset Codebook)	Rogaway	One-pass (one blockcipher call per block of message)
GCM (Galois Counter Mode)	McGrew, Viega	CTR mode plus specialized MAC
CWC	Kohno, Viega, Whiting	CTR mode plus Carter-Wegman MAC
CCM	Housley, Ferguson, Whiting	CTR mode plus CBC-MAC
EAX	Wagner, Bellare, Rogaway	CTR mode plus OMAC (variant of CBC-MAC)

Symmetric Encryption Advice

Never use CTR mode or CBC mode by themselves

Passive security is almost never good enough!!

Encrypt-then-MAC better than MAC-then-Encrypt, Encrypt and MAC

Dedicated modes that have been analyzed thoroughly are also good