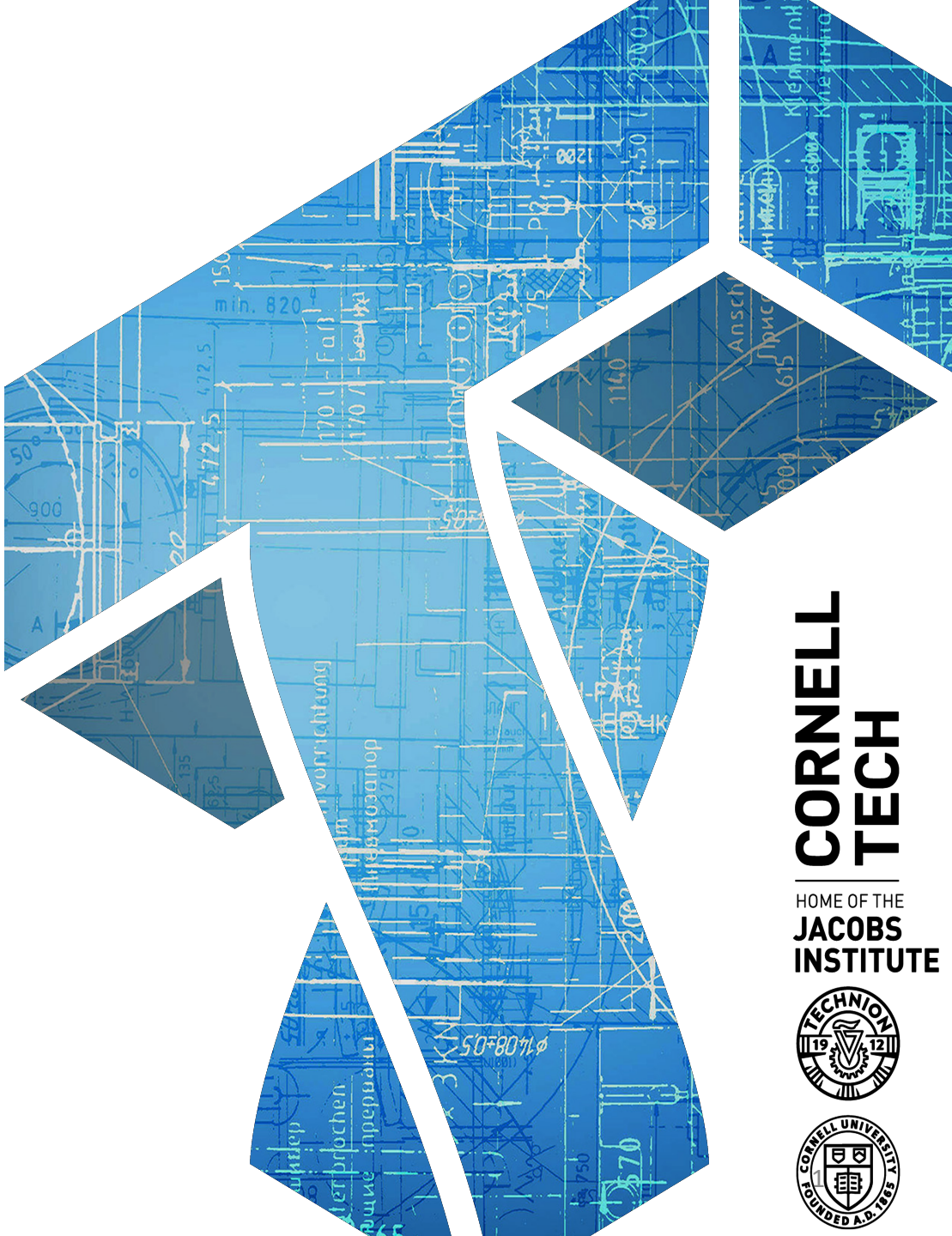
# CS 5830
# Cryptography

Instructor: Tom Ristenpart
TAs: Yan Ji, Sanketh Menda

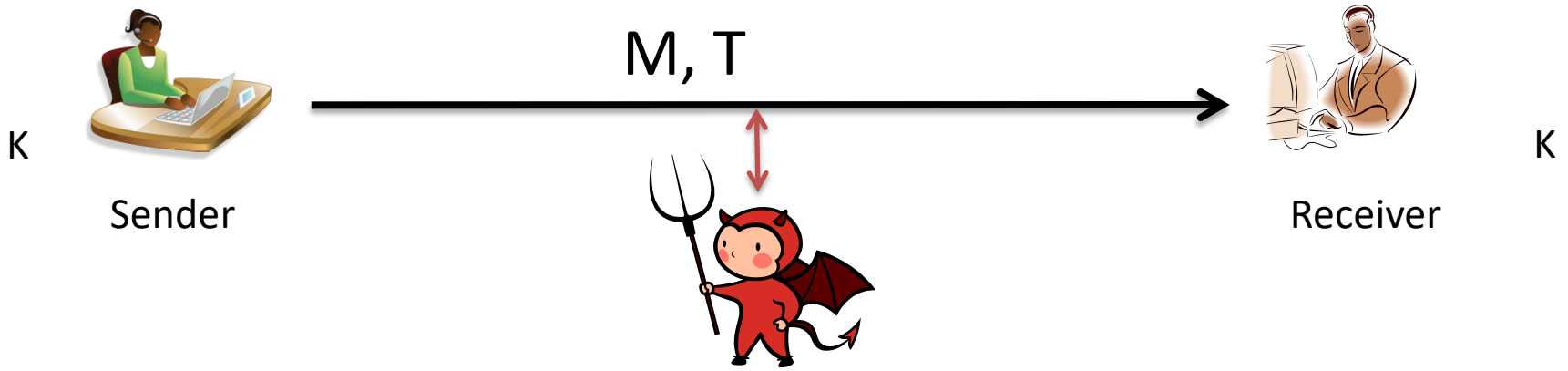


CORNELL TECH
HOME OF THE JACOBS INSTITUTE

# Primitives we've seen so far

- **Stream ciphers**
  - Secret key + IV => pseudorandom bit string ("pad")
  - RC4, CTR mode as a stream cipher
- **Block ciphers**
  - Secret key defines permutation on n bits (n is blocklength)
  - Examples: AES, DES
- **IND-CPA symmetric encryption (Kg, Enc, Dec)**
  - Secret key + per-message randomness (IV)
  - Examples: CTR mode, CBC mode (requires message padding)
- **Message authentication schemes (MACs)**
  - Secret key + message => authentication tag
  - Examples: CBC-MAC

# Message authentication schemes




Scheme is triple of algorithms MA = (Kg,Tag,Verify):

(1) Kg outputs secret key K

(2) Tag(K,M) outputs a tag T

(3) Verify(K,M,T) outputs 0/1 (invalid / valid)

Message authentication code (MAC):
Tag deterministic +
Verify checks by recomputing Tag

*Correctness*: Verify(K,M,Tag(K,M)) = 1 always

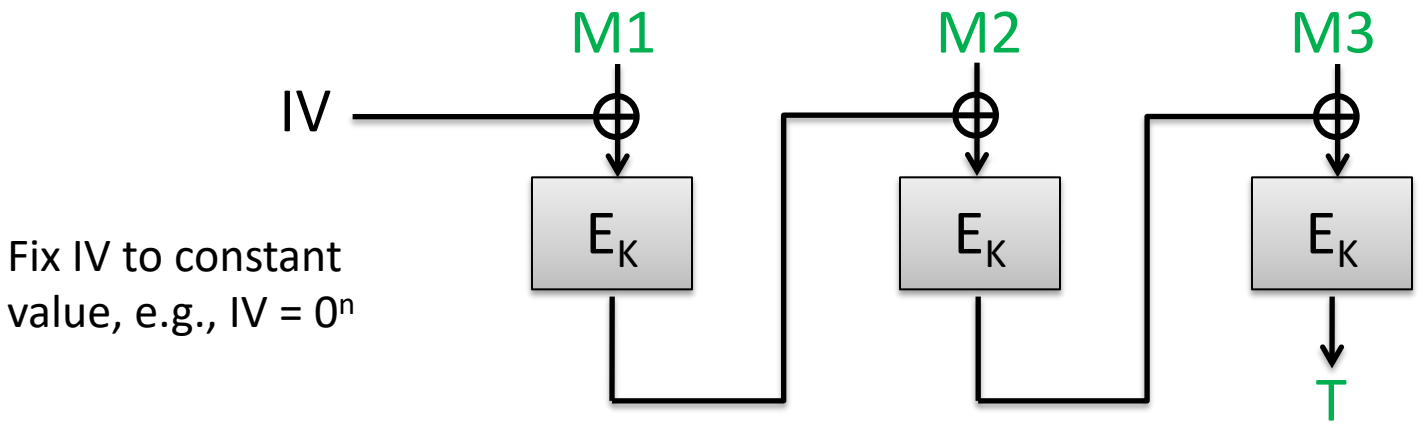
*Security*: No computationally efficient attacker can forge tags for a new message even when attacker gets

$$(M_1, T_1), (M_2, T_2), \ldots, (M_q, T_q)$$

for messages of their choosing and reasonably large q.

# CBC-MAC

Message authentication code (MAC)




Fix IV to constant value, e.g., IV = $0^n$

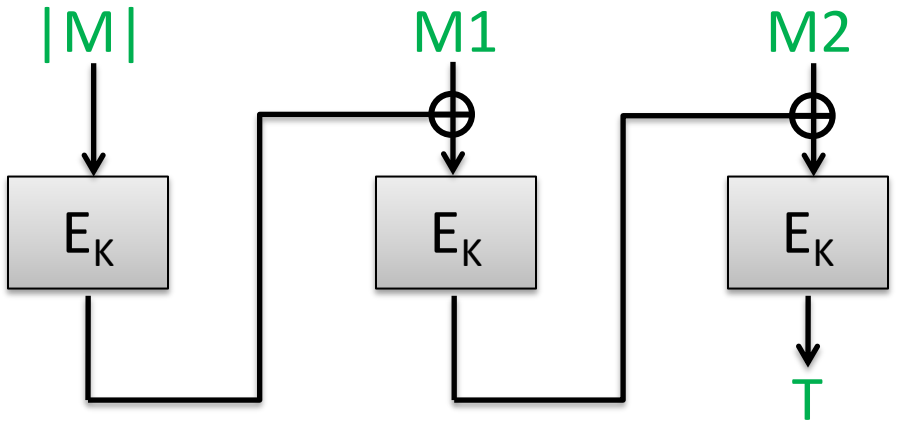
Turns out this is (provably) a good PRF

***if K used only on same-length messages***

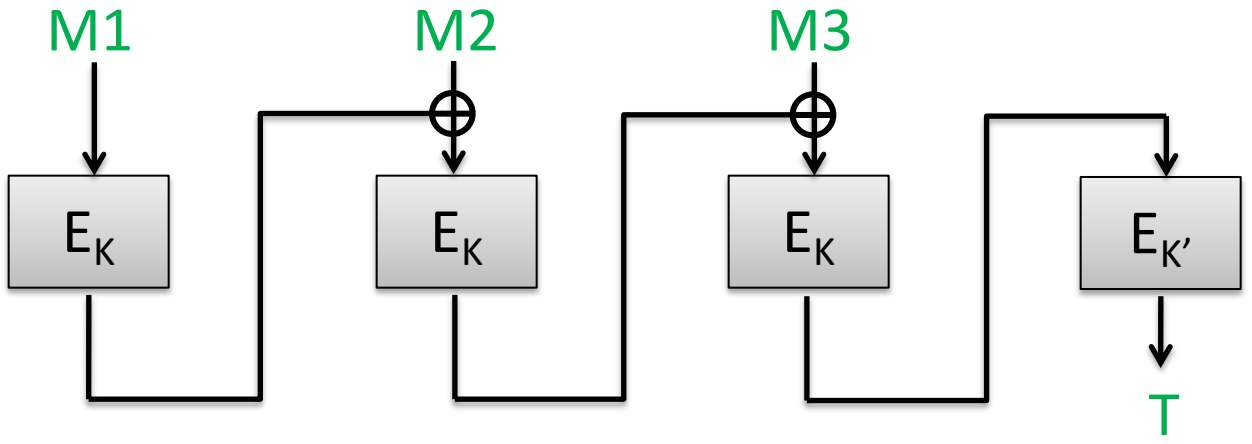
http://blog.cryptographyengineering.com/2013/02/why-i-hate-cbc-mac.html

# Secure variable-message-length CBC-MAC

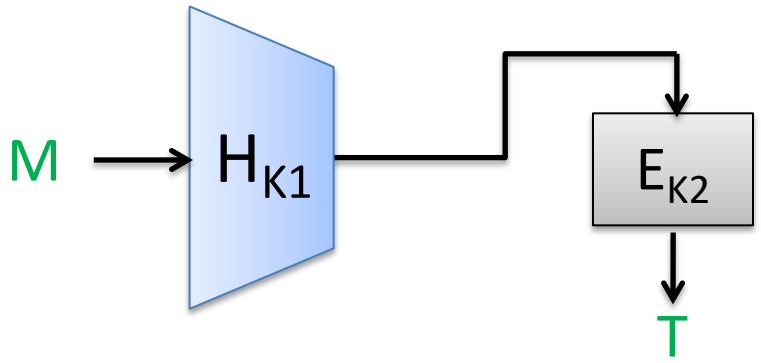
- Prepend message length




- Encrypted CBC-MAC



# Universal hash then PRF approach




A keyed function $H : \{0,1\}^k \times \mathcal{M} \rightarrow \{0,1\}^n$ is an $\epsilon$-almost universal (AU) hash function if for all $M \neq M'$

$$\Pr\left[\, H_K(M) = H_K(M') \,\right] \leq \epsilon$$

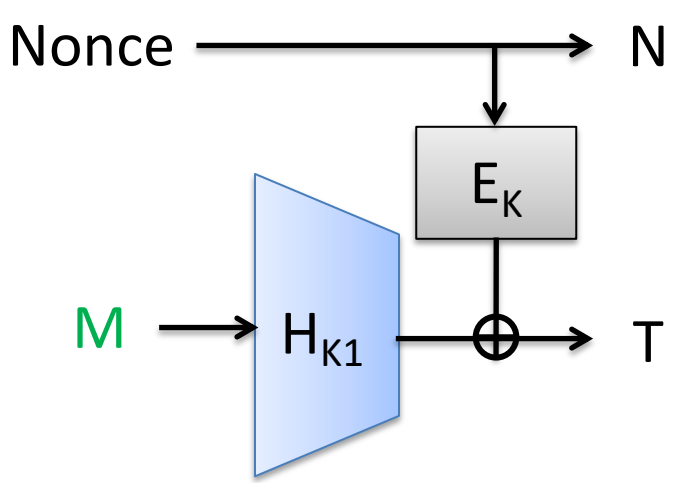
where the probability is over choice of $K$.

Turns out this is a good construction for any H that is a (computational) ***universal hash function***

***Intuition***: $E_{K2}$ hides info about K1 unless collision. For q queries, chance of collision at most

$q^2 \epsilon / 2$

# Carter-Wegman MA Schemes




Nonce must be fresh for each M
Random value or counter
If repeated, security fails

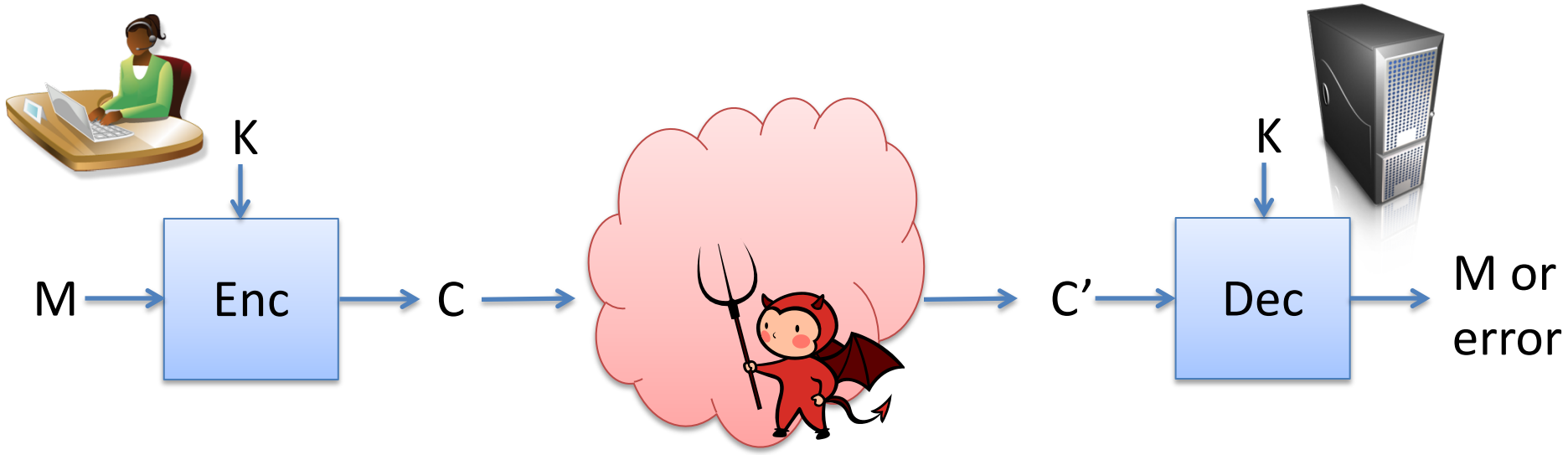
Intuition:
Hide universal hash by encrypting it

A keyed function $H : \{0,1\}^k \times \mathcal{M} \rightarrow \{0,1\}^n$ is an $\epsilon$-almost universal (AU) hash function if for all $M \neq M'$

$$\Pr\left[\, H_K(M) = H_K(M') \,\right] \leq \epsilon$$

where the probability is over choice of $K$.

# Authenticated encryption (AE)



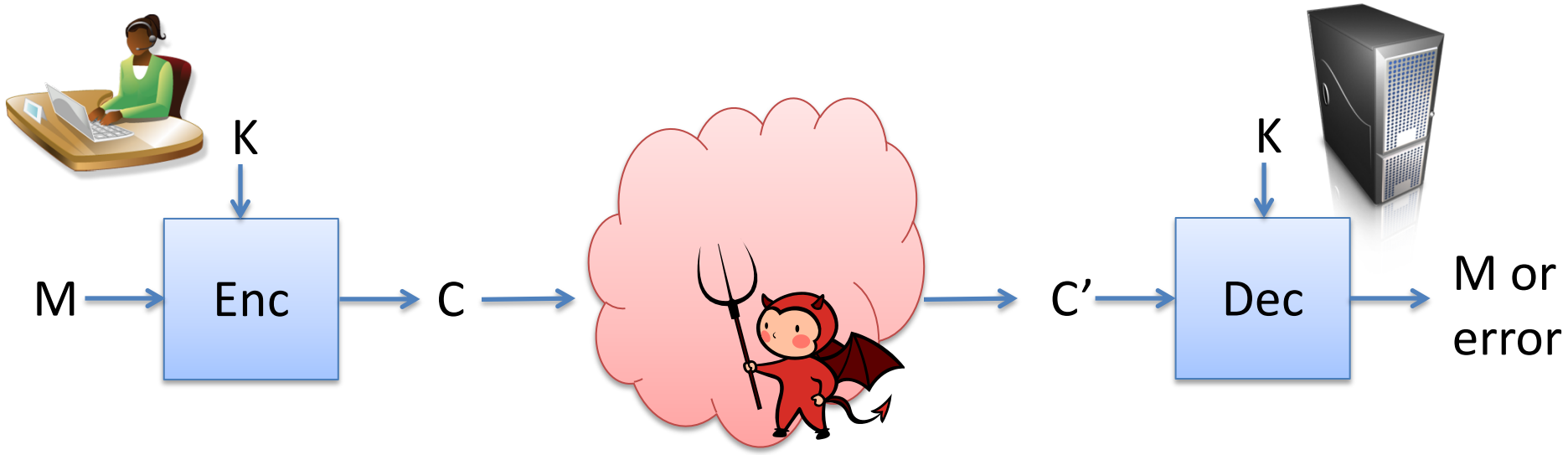

What security properties do we need from symmetric encryption?

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)



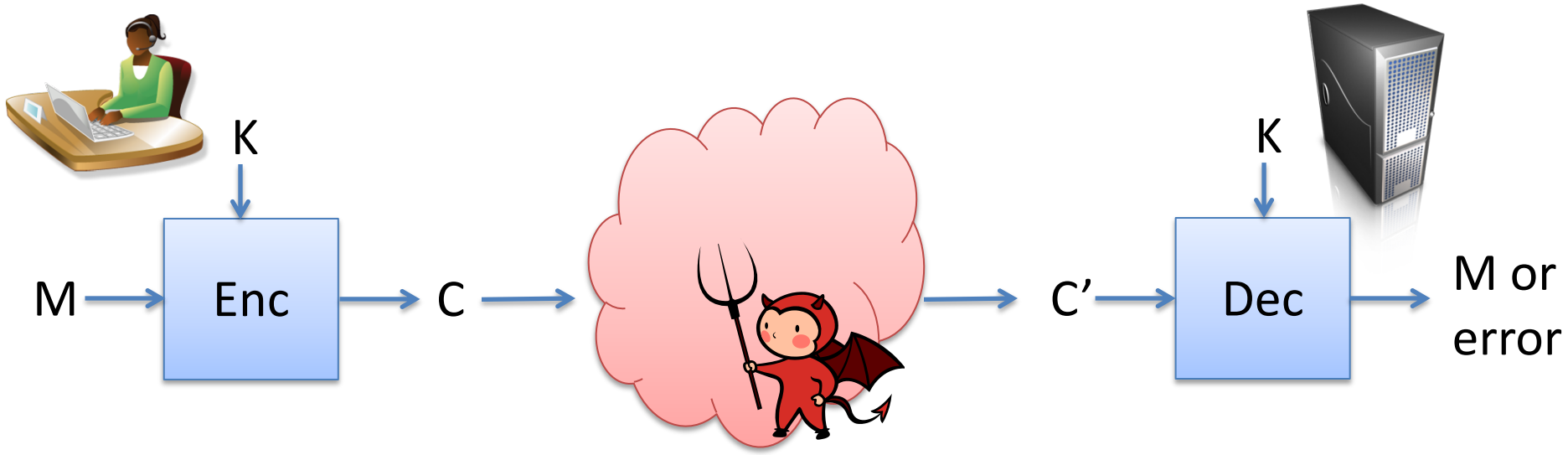

What security properties do we need from symmetric encryption?

1) Confidentiality: computational indistinguishability (IND-CPA)

2) Authenticity: ???

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

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

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

# Formalizing CTXT

***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 $1 \leq i \leq q$

Measure advantage of adversary via probability wins game:

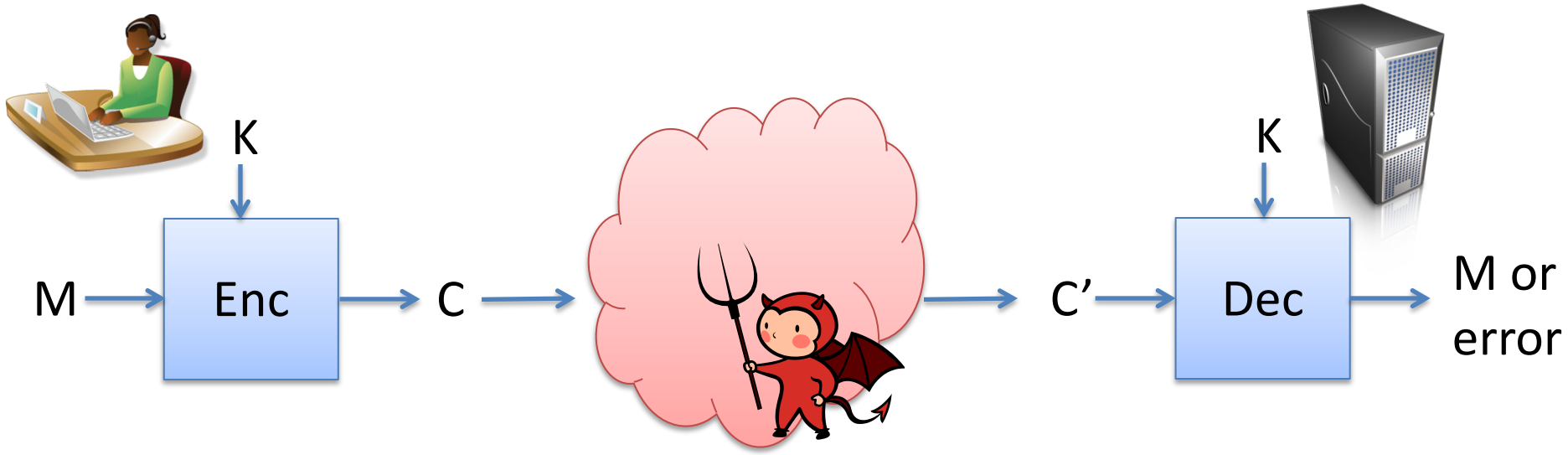
$$\Pr[\text{CTXT}(\text{SE}, \mathcal{A}) \Rightarrow 1]$$

CTXT(SE, $\mathcal{A}$ ):
K <-$ Kg
win <- 0
$\mathcal{A}^{\text{Enc,Dec}}$
Return win

Enc(M)
C <-$ Enc(K,M)
CtxtSet <- CtxtSet U {C}
Return C

Tag(C')
If C' in CtxtSet then Return $\perp$
M <- Dec(K,C)
If M $\neq \perp$ then win <- true
Return M

# Authenticated encryption (AE)




What security properties do we need from symmetric encryption?

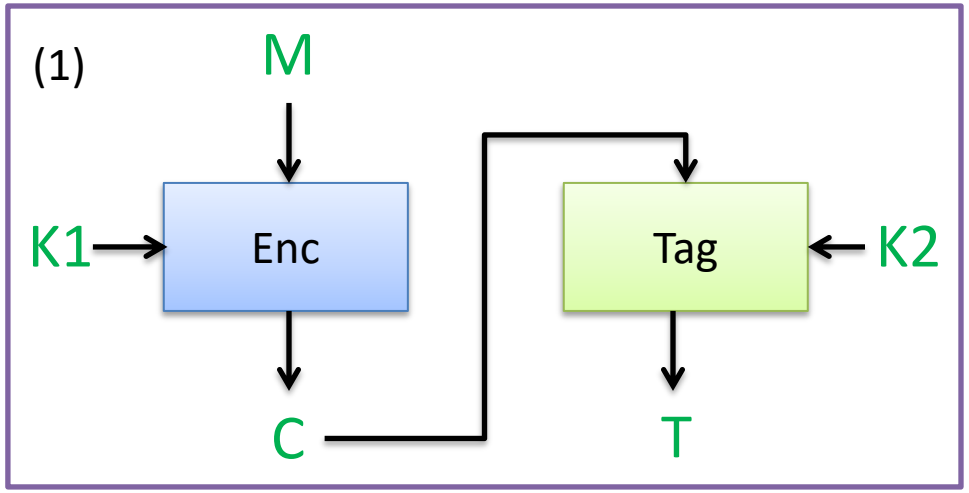
1) Confidentiality: computational indistinguishability (IND-CPA)

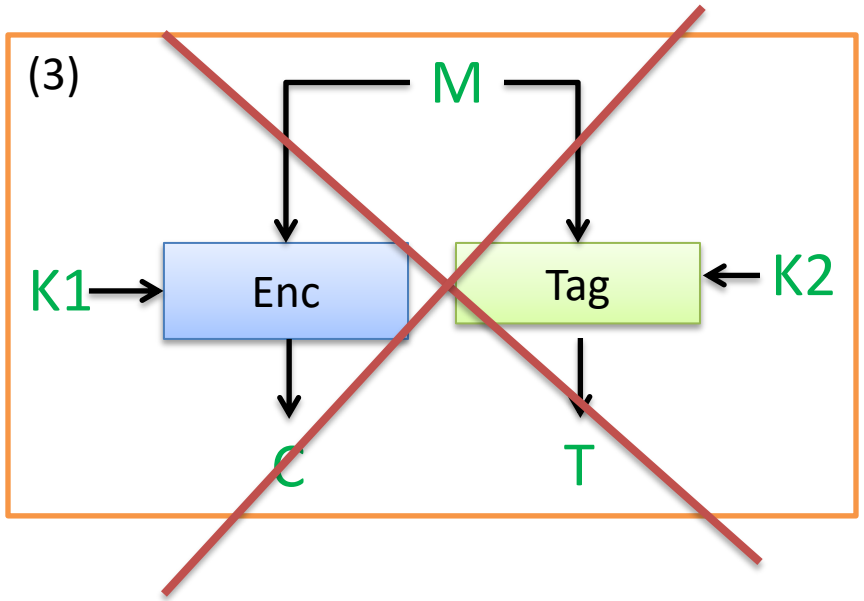
2) Authenticity: ciphertext unforgeability (CTXT)

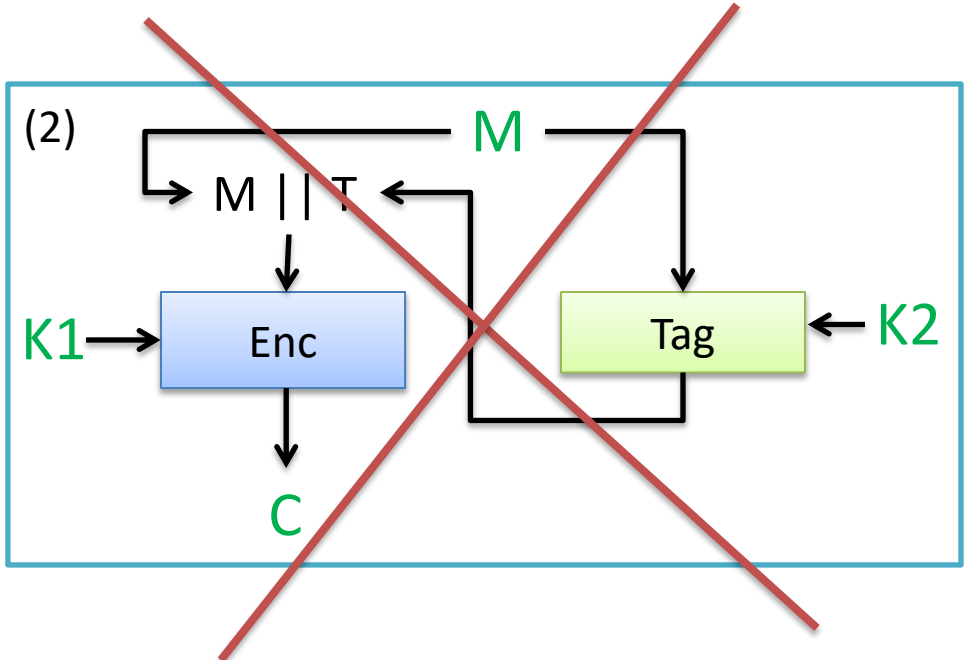
Often referred to as Authenticated Encryption security

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

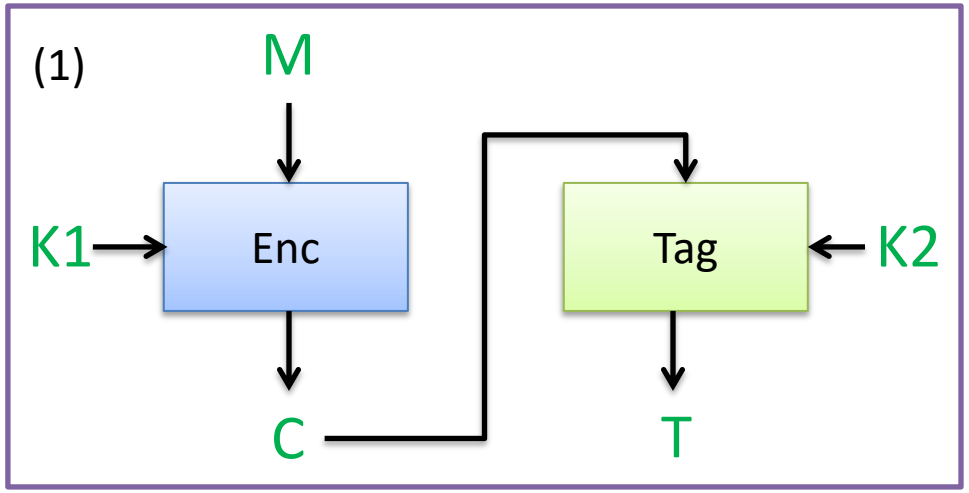


Build a new scheme from Enc mode (CBC, CTR) and MAC
Kg outputs Enc key K1 and MAC key K2

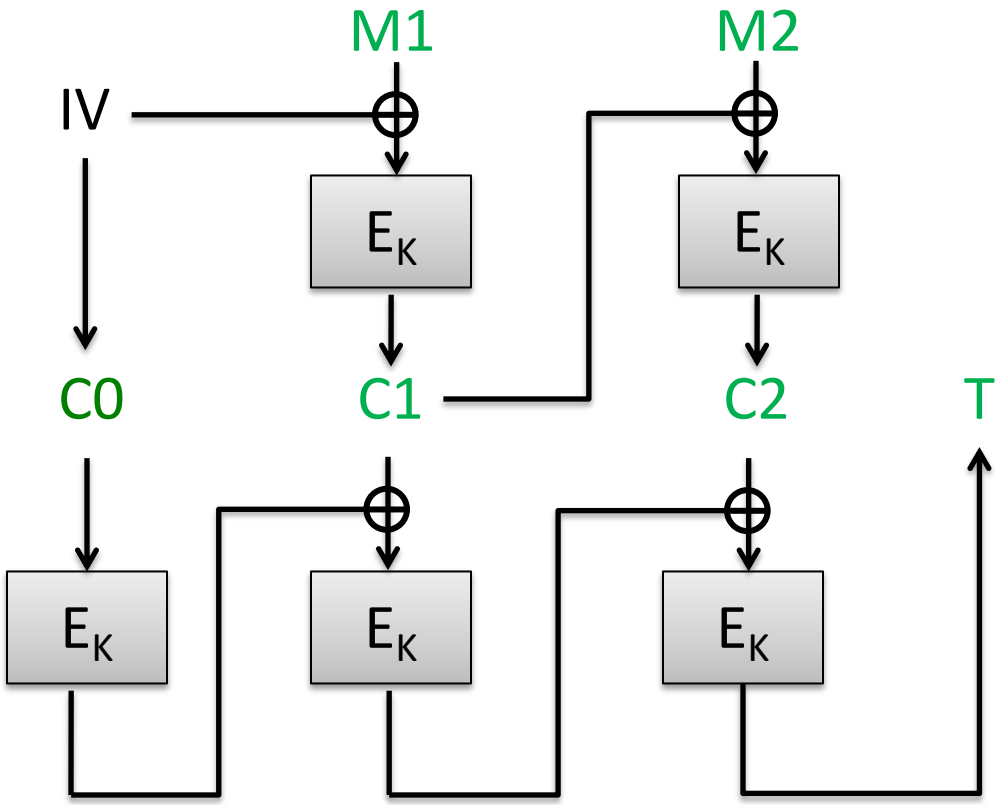
Several ways to combine:
(1) encrypt-then-mac
(2) mac-then-encrypt
(3) encrypt-and-mac




Thm. If SE scheme is IND-CPA and MAC is UF-CMA, then Encrypt-then-MAC is IND-CPA and CTXT

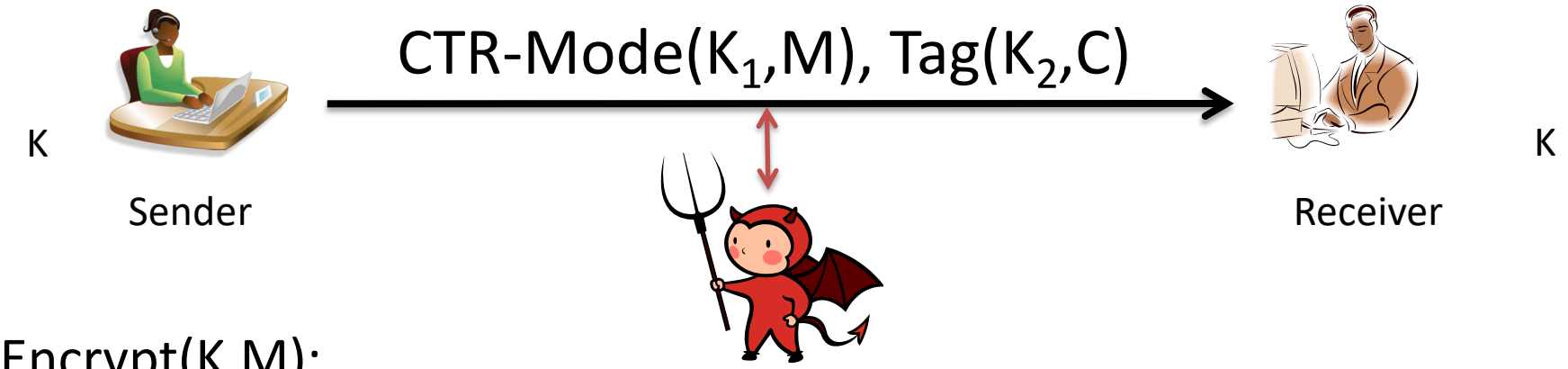
# Key separation is critical.

## Using same key with CBC-Mode + CBC-MAC:




This is insecure!

# Key derivation functions




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-Mode($K_1$, M)

T <- Tag($K_2$,C)

Output C || T

Decrypt(K,C || T)

$K_1$ <- AES(K,$0^n$) ; $K_2$ <- AES(K,$1^n$)

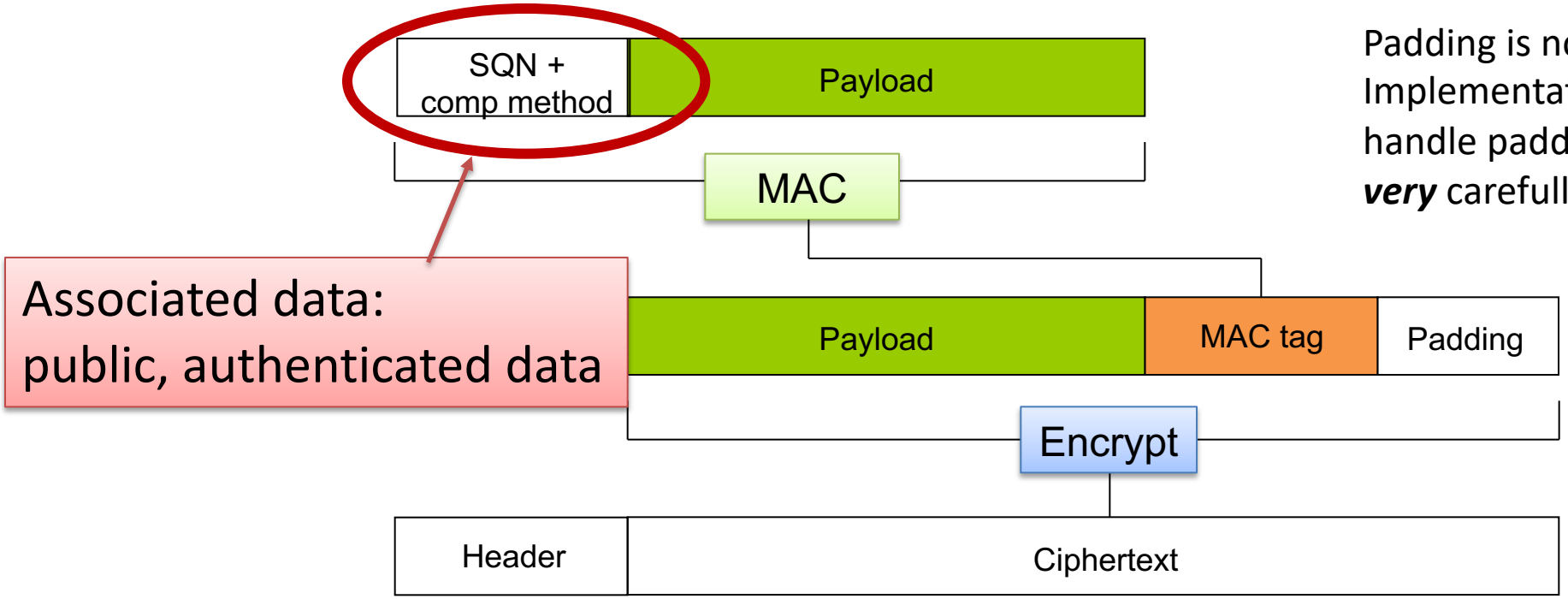
If Verify($K_2$,C,T) ≠ 1 then Return error

Return CTR-Mode-Decrypt($K_1$,C)

Key derivation function (KDF):
Way to use one key to generate several

HKDF is widely used general-purpose KDF

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

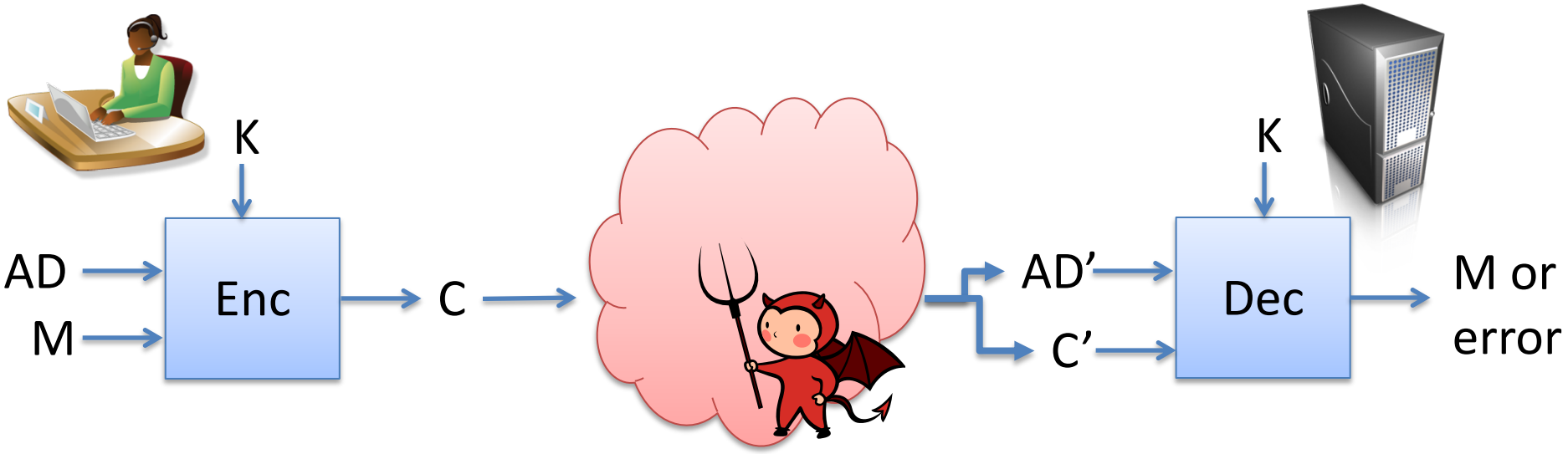



Padding is not MAC'd. Implementations must handle padding checks ***very*** carefully.

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

Now TLS uses better authenticated-encryption schemes, this is deprecated

# Authenticated encryption w/ associated data (AEAD)




Associated data need not be confidential, but must be authenticated

Typically used for cleartext headers sent

- TLS 1.2 example: sequence number, compression methods
- Context information (database row number, disk sector number, etc.)

Can extend IND-CPA and CTXT definitions to cover AEAD

# Dedicated authenticated encryption schemes

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

| Attack | Inventors | Notes |
|---|---|---|
| OCB<br>(Offset Codebook) | Rogaway | One-pass (one blockcipher call per block of message) |
| GCM<br>(Galois Counter Mode) | McGrew, Viega | CTR mode plus specialized MAC |
| Salsa20/Poly1305 | Bernstein | Stream cipher plus Carter-Wegman 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 best way to do generic composition

Dedicated modes that have been analyzed thoroughly are way to go in practice