Authenticated Encryption

Adam O'Neill

Based on http://cseweb.ucsd.edu/~mihir/cse107/

Motivation

In practice we often want both privacy and authenticity.

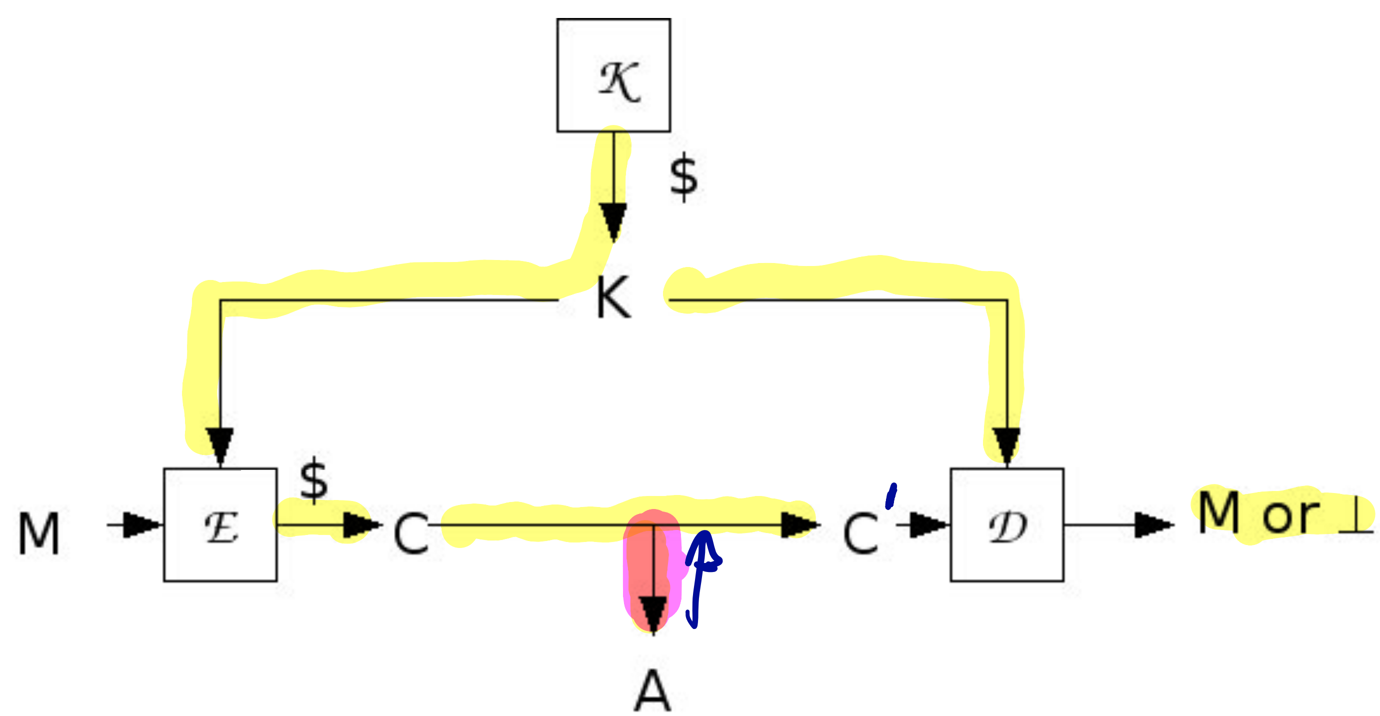
Example: A doctor wishes to send medical information M about Alice to the medical database. Then

- We want data privacy to ensure Alice's medical records remain confidential.
- We want authenticity to ensure the person sending the information is really the doctor and the information was not modified in transit.

We refer to this as authenticated encryption. Combines enc + macs.

Syntax

Syntactically, an authenticated encryption scheme is just a symmetric encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where



Security

- The same notion of privacy applies, namely IND-CPA

Security

- The same notion of privacy applies, namely IND-CPA
- For authenticity, the adversary's goal is to get the receiver to accept a "non-authentic" ciphertext (i.e., not actually transmitted by the sender)

INT-CTXT

integrity of ciphertexts

INT-CTXT

very similar to UF-CMA

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme and A an adversary.

Game $\text{INTCTXT}_{\mathcal{AE}}$

procedure Initialize
$K \xleftarrow{\$} \mathcal{K}$; $S \leftarrow \emptyset$

procedure Enc(M)
$C \xleftarrow{\$} \mathcal{E}_K(M)$
$S \leftarrow S \cup \{C\}$
Return C

procedure Finalize(C^*)
$M \leftarrow \mathcal{D}_K(C^*)$
if $(C^* \notin S \wedge M \neq \perp)$ then
    return true
Else return false

The int-ctxt advantage of A is

$$\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{AE}}(A) = \Pr[\text{INTCTXT}^A_{\mathcal{AE}} \Rightarrow \text{true}]$$

Integrity + Privacy

The goal of authenticated encryption is to provide both integrity and privacy. We will be interested in IND-CPA + INT-CTXT.

Plain Encryption: CBC$

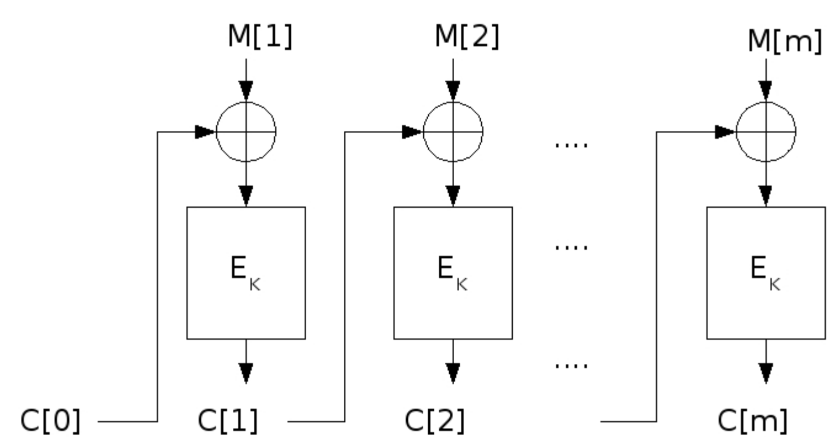
It is IND-CPA assuming
E is a good PRF

Alg $\mathcal{E}_K(M)$

$C[0] \xleftarrow{\$} \{0,1\}^n$
For $i = 1, \ldots, m$ do
  $C[i] \leftarrow \mathrm{E}_K(C[i-1] \oplus M[i])$
Return C

Alg $\mathcal{D}_K(C)$

For $i = 1, \ldots, m$ do
  $M[i] \leftarrow \mathrm{E}_K^{-1}(C[i]) \oplus C[i-1]$
Return M



Question: Is CBC$ encryption INT-CTXT secure?

NO!

Plain Encryption Does Not Provide Integrity

$\perp$ "bot" error or undefined

Alg $\mathcal{E}_K(M)$

$C[0] \xleftarrow{\$} \{0,1\}^n$
For $i = 1, \ldots, m$ do
  $C[i] \leftarrow \mathsf{E}_K(C[i-1] \oplus M[i])$
Return C

Alg $\mathcal{D}_K(C)$

For $i = 1, \ldots, m$ do
  $M[i] \leftarrow \mathsf{E}_K^{-1}(C[i]) \oplus C[i-1]$
Return M

adversary A

$C[0]C[1]C[2] \xleftarrow{\$} \{0,1\}^{3n}$
Return $C[0]C[1]C[2]$

breaks INT-CTXT

Then

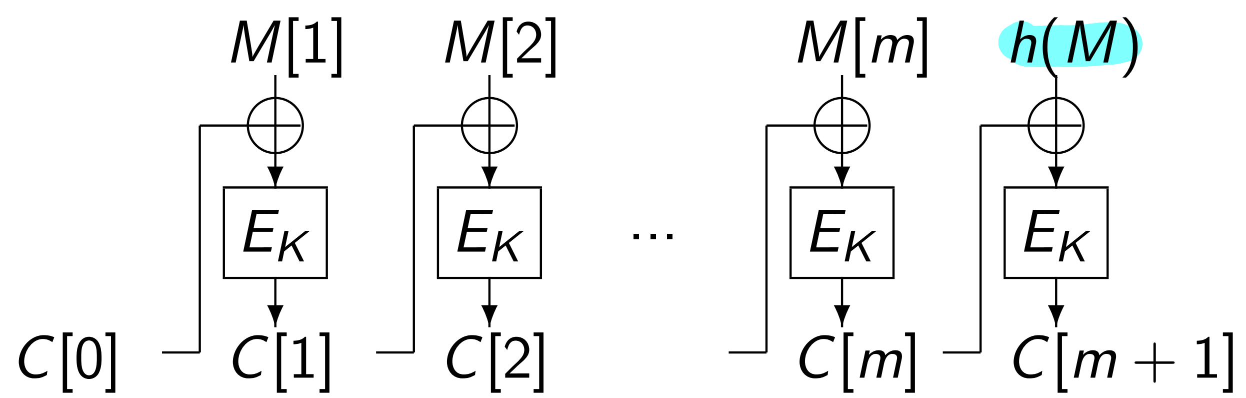
$$\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE}}(A) = 1$$

This violates INT-CTXT.

A scheme whose decryption algorithm never outputs $\perp$ cannot provide integrity!

Encryption with Redundancy

CBC with redundancy

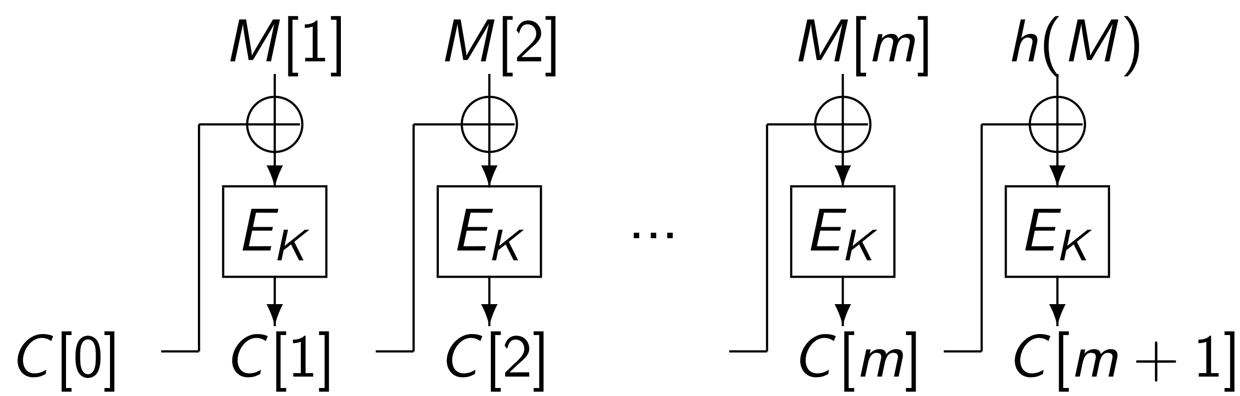


Here $E\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is our block cipher and $h\colon \{0,1\}^* \to \{0,1\}^n$ is a "redundancy" function, for example

- $h(M[1] \ldots M[m]) = 0^n$
- $h(M[1] \ldots M[m]) = M[1] \oplus \cdots \oplus M[m]$ ← checksum
- A CRC
- $h(M[1] \ldots M[m])$ is the first n bits of SHA1$(M[1] \ldots M[m])$.

The redundancy is verified upon decryption.

Encryption with Redundancy



Let E: $\{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be our block cipher and h: $\{0,1\}^* \to \{0,1\}^n$ a redundancy function. Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ be CBC$ encryption and define the encryption with redundancy scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ via

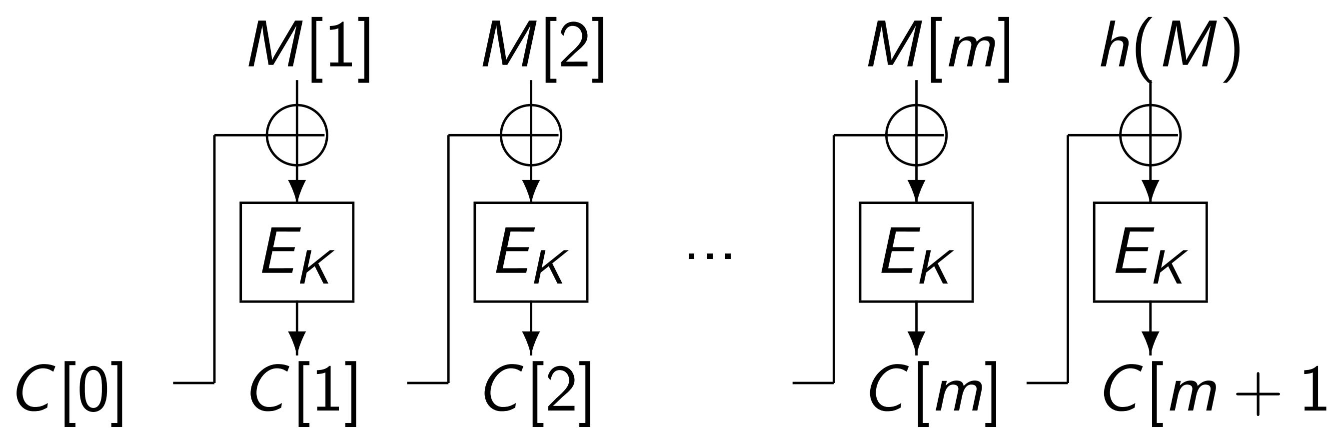
Alg $\mathcal{E}_K(M)$

$M[1] \ldots M[m] \leftarrow M$
$M[m+1] \leftarrow h(M)$
$C \xleftarrow{\$} \mathcal{E}'_K(M[1] \ldots M[m]M[m+1])$
return C

Alg $\mathcal{D}_K(C)$

$M[1] \ldots M[m]M[m+1] \leftarrow \mathcal{D}'_K(C)$
if $(M[m+1] = h(M))$ then
  return $M[1] \ldots M[m]$
else return $\perp$

Does it Work?



The adversary will have a hard time producing the last enciphered block of a new message.

intuition

Is the intuition correct?

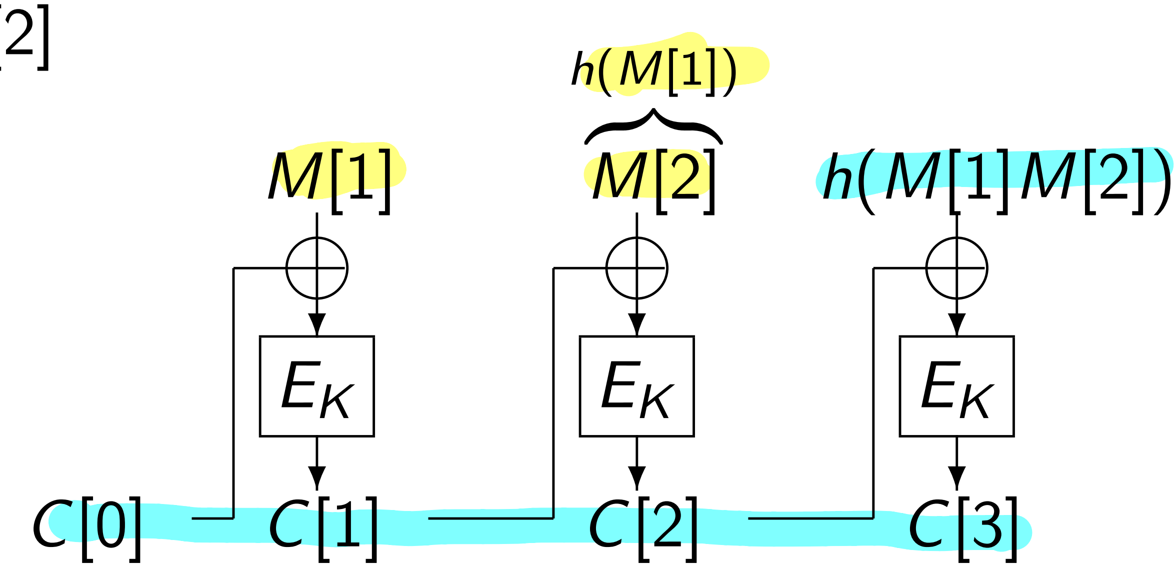
Attacks

adversary A

$M[1] \xleftarrow{\$} \{0,1\}^n$; $M[2] \leftarrow h(M[1])$

$C[0]C[1]C[2]C[3] \xleftarrow{\$} \textbf{Enc}(M[1]M[2])$

Return $C[0]C[1]C[2]$



This attack succeeds for any (not secret-key dependent) redundancy function h.

doesn't matter how "complicated" your redundancy fn. is.

WEP Attack

A “real-life” rendition of this attack broke the 802.11 WEP protocol, which instantiated h as CRC and used a stream cipher for encryption [BGW].

What makes the attack easy to see is having a clear, strong and formal security model.

Generic Composition

Build an authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ by combining

how to combine?

- a given IND-CPA symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$
- a given PRF $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^n$

	CBC\$-AES	CTR\$-AES	...
HMAC-SHA1	AE scheme		
CMAC			
ECBC			
$\vdots$			

Want generic composition methods that work for arbitrary secure starting schemes.

Generic Composition

Build an authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ by combining

- a given IND-CPA symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$
- a given PRF $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^n$

A key $K = K_e || K_m$ for $\mathcal{AE}$ always consists of a key K_e for $\mathcal{SE}$ and a key K_m for F:

MAC
(prf)

Alg $\mathcal{K}$

$K_e \xleftarrow{\$} \mathcal{K}'$; $K_m \xleftarrow{\$} \{0,1\}^k$
Return $K_e || K_m$

Generic Composition

The order in which the primitives are applied is important. Can consider

Method	Usage
Encrypt-and-MAC (E&M)	SSH
MAC-then-encrypt (MtE)	SSL/TLS
Encrypt-then-MAC (EtM)	IPSec

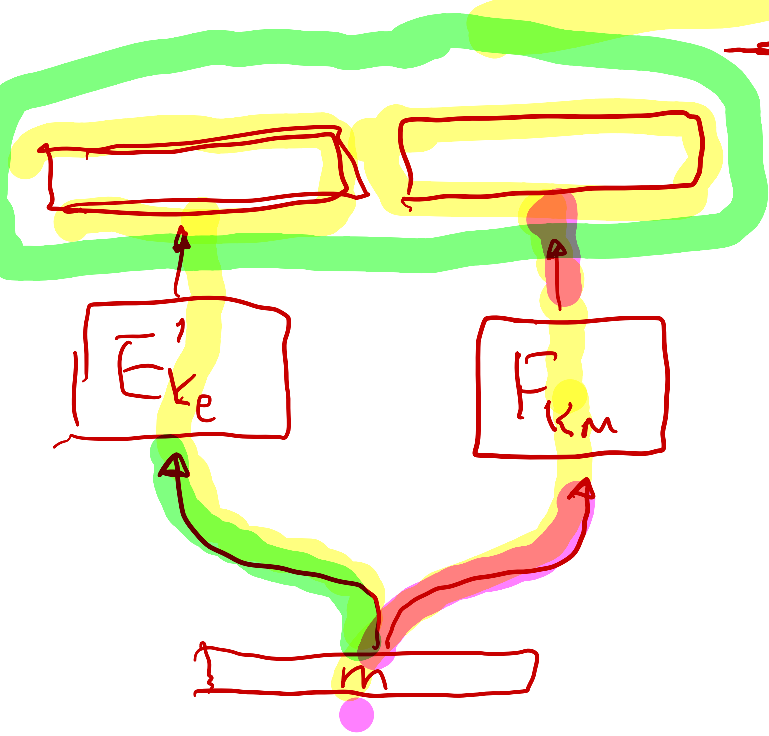
Encrypt-and-MAC

PRF

PRF == MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e \| K_m}(M)$	**Alg** $\mathcal{D}_{K_e \| K_m}(C' \| T)$
$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$	$M \leftarrow \mathcal{D}'_{K_e}(C')$
$T \leftarrow F_{K_m}(M)$	If $(T = F_{K_m}(M))$ then return M
Return $C' \| \boxed{T}$	Else return $\perp$



Security	Achieved?
IND-CPA	NO!
INT-CTXT	NO!

why?
F_{K_m} is deterministic so usual attack applies

ciphertexts could have a "superfluous" bit

Decryption never uses last bit of ciphertext

Adversary $A^{\mathcal{E}(LR(\cdot,\cdot,b)) \,//\, \mathcal{O}(\cdot,\cdot)}$

$c_1 \| t_1 \leftarrow \mathcal{O}(1^n, 1^n)$

$c_2 \| t_2 \leftarrow \mathcal{O}(1^n, 0^n)$

If $t_1 = t_2 \ldots$

Idea: // For INT-CTXT adversary

Adversary $A^{\mathrm{ENC}(\cdot)}$

$c \| t \leftarrow \mathrm{Enc}(0^n)$

Parse c as $c' \| b$ superfluous bit

$c_{new} \leftarrow c' \| \bar{b}$ ← bitwise comp.

ret $c_{new} \| t$

MAC-then-Encrypt

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e\|\|K_m}(M)$	**Alg** $\mathcal{D}_{K_e\|\|K_m}(C)$
$T \leftarrow F_{K_m}(M)$	$M\|\|T \leftarrow \mathcal{D}'_{K_e}(C)$
$C \xleftarrow{\$} \mathcal{E}'_{K_e}(M\|\|T)$	If $(T = F_{K_m}(M))$ then return M
Return C	Else return $\perp$

Security	Achieved?
IND-CPA	YES!
INT-CTXT	NO!

superfluous bits

encrypt $\boxed{m \| t}$

where t is tag of m.

Encrypt-then-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$
$T \leftarrow F_{K_m}(C')$
Return $C'||T$ $= C$

Alg $\mathcal{D}_{K_e||K_m}(C'||T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$
If $(T = F_{K_m}(C'))$ then return M
Else return $\perp$

Security	Achieved?
IND-CPA	YES
INT-CTXT	YES

Two keys?

1 physical key ⇝ {2 synthetic key

We have used separate keys K_e, K_m for the encryption and message authentication. However, these can be derived from a single key K via $K_e = F_K(0)$ and $K_m = F_K(1)$, where F is a PRF such as a block cipher, the CBC-MAC or HMAC.

Trying to directly use the same key for the encryption and message authentication is error-prone, but works if done correctly.

→ CMAC (OMAC)

one-key MAC

no re-keying

Generic Composition in Practice

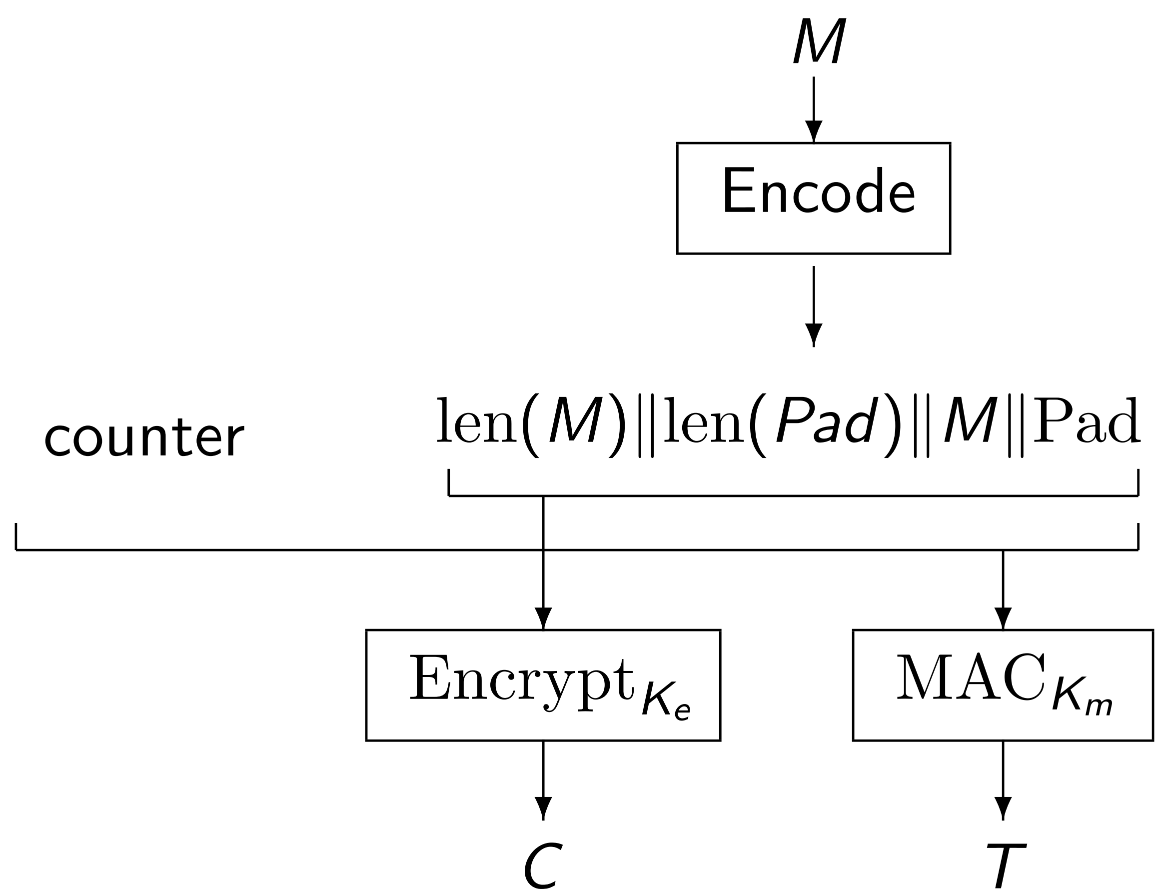
SSL 3.0 : POODLE CBC-$
(padding)

AE in	**is based on**	**which in general is**	**and in this case is**
SSH	E&M	insecure	secure
SSL	MtE	insecure	insecure
SSL + RFC 4344	MtE	insecure	secure
IPSec	EtM	secure	secure
WinZip	EtM	secure	insecure

Why?

- Encodings
- Specific "E" and "M" schemes
- For WinZip, disparity between usage and security model

AE in SSH



SSH2 encryption uses inter-packet chaining which is insecure [D, BKN]. RFC 4344 [BKN] proposed fixes that render SSH provably IND-CPA + INT-CTXT secure. Fixes recommended by Secure Shell Working Group and included in OpenSSH since 2003. Fixes included in PuTTY since 2008.