CCA-secure encryption:

Encrypt-then-MAC (*EtM*):

$$\mathcal{K} = E.\mathcal{K} \times M.\mathcal{K}$$
$$\mathcal{M} = E.\mathcal{M}$$
$$\mathcal{C} = E.\mathcal{C} \times M.\mathcal{T}$$

KeyGen:
- $k_e \leftarrow E.\text{KeyGen}$
- $k_m \leftarrow M.\text{KeyGen}$
- return (k_e, k_m)

$\text{Enc}((k_e, k_m), m)$:
- $c \leftarrow E.\text{Enc}(k_e, m)$
- $t := M.\text{MAC}(k_m, c)$
- return (c, t)

$\text{Dec}((k_e, k_m), (c, t))$:
- if $t \neq M.\text{MAC}(k_m, c)$:
  - return err
- return $E.\text{Dec}(k_e, c)$

Claim:

If E is a CPA-secure encryption scheme, and M is a secure MAC, then *EtM* is a CCA-secure encryption scheme. That is, $\mathcal{L}^{EtM}_{\text{cca-L}} \approx \mathcal{L}^{EtM}_{\text{cca-R}}$.

Overview:

Want to show:

$\mathcal{L}^{EtM}_{\text{cca-L}}$		$\mathcal{L}^{EtM}_{\text{cca-R}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$
 	
$\text{CHALLENGE}(m_L, m_R)$:
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t \leftarrow M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
  return (c, t)
 	
$\text{DEC}(c, t)$:
  if $(c, t) \in \mathcal{S}$ return null
  if $t \neq M.\text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$
	$\approx$	$k_e \leftarrow E.\text{KeyGen}$
		$k_m \leftarrow M.\text{KeyGen}$
		$\mathcal{S} := \emptyset$
		
		$\text{CHALLENGE}(m_L, m_R)$:
		  $c \leftarrow E.\text{Enc}(k_e, m_R)$
		  $t \leftarrow M.\text{MAC}(k_m, c)$
		  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
		  return (c, t)
		
		$\text{DEC}(c, t)$:
		  if $(c, t) \in \mathcal{S}$ return null
		  if $t \neq M.\text{MAC}(k_m, c)$:
		    return err
		  return $E.\text{Dec}(k_e, c)$

The proof will **use** the fact that E has CPA security and M is a secure MAC.

$$\mathcal{L}^{EtM}_{\text{cca-L}}$$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$

$\underline{\text{CHALLENGE}(m_L, m_R)\text{:}}$
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t \leftarrow M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
  return (c, t)

$\underline{\text{DEC}(c, t)\text{:}}$
  if $(c, t) \in \mathcal{S}$ return null
  if $t \neq M.\text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$

Starting point is $\mathcal{L}^{EtM}_{\text{cca-L}}$.

Security proof

$\mathcal{L}^{EtM}_{\text{cca-L}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$

CHALLENGE(m_L, m_R):
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t \leftarrow M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
  return (c, t)

DEC(c, t):
  if $(c, t) \in \mathcal{S}$ return null
  if $t \neq M.\text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$

Starting point is $\mathcal{L}^{EtM}_{\text{cca-L}}$. Can we switch m_L to m_R right away?

Security proof

$\mathcal{L}^{EtM}_{\text{cca-L}}$		$\mathcal{L}^{E}_{\text{cpa-L}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$
 	
$\underline{\text{CHALLENGE}(m_L, m_R):}$
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t \leftarrow M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
  return (c, t)
 	
$\underline{\text{DEC}(c, t):}$
  if $(c, t) \in \mathcal{S}$ return null
  if $t \neq M.\text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$
	$\diamond$	$k_e \leftarrow E.\text{KeyGen}$
		
		$\underline{\text{CHALLENGE}'(m_L, m_R):}$
		  $c := E.\text{Enc}(k_e, m_L)$
		  return c

Can we factor out in terms of $\mathcal{L}_{\text{cpa-L}}$?

$\mathcal{L}^{EtM}_{\text{cca-L}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$

CHALLENGE(m_L, m_R):
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t \leftarrow M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}$
  return (c,t)

DEC(c,t):
  if $(c,t) \in \mathcal{S}$ return null
  if $t \neq M.\text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$

$\diamond$

$\mathcal{L}^{E}_{\text{cpa-L}}$

$k_e \leftarrow E.\text{KeyGen}$

CHALLENGE$'(m_L, m_R)$:
  $c := E.\text{Enc}(k_e, m_L)$
  return c

Can we factor out in terms of $\mathcal{L}_{\text{cpa-L}}$? No, must get rid of $E.\text{Dec}$!

Security proof

$\mathcal{L}^{EtM}_{\text{cca-L}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$

CHALLENGE(m_L, m_R):
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t \leftarrow M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}$
  return (c,t)

DEC(c,t):
  if $(c,t) \in \mathcal{S}$ return null
  if $t \neq M.\text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$

Deal with MAC first

Security proof

$\mathcal{L}^{EtM}_{\text{cca-L}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$

CHALLENGE(m_L, m_R):
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t \leftarrow M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}$
  return (c, t)

DEC(c, t):
  if $(c,t) \in \mathcal{S}$ return null
  if $t \neq M.\text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$

Deal with MAC first; factor out in terms of $\mathcal{L}_{\text{mac-real}}$

Security proof

$k_e \leftarrow E.\text{KeyGen}$
$\mathcal{S} := \emptyset$

$\underline{\textsc{challenge}(m_L, m_R)}$:
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t := \textsc{getmac}(c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
  return (c, t)

$\underline{\textsc{dec}(c, t)}$:
  if $(c, t) \in \mathcal{S}$ return null
  if not $\textsc{ver}(c, t)$:
    return err
  return $E.\text{Dec}(k_e, c)$

$\diamond$

$\mathcal{L}^{M}_{\text{mac-real}}$

$k_m \leftarrow M.\text{KeyGen}$

$\underline{\textsc{getmac}(c)}$:
  return $M.\text{MAC}(k_m, c)$

$\underline{\textsc{ver}(c, t)}$:
  return $t \stackrel{?}{=} M.\text{MAC}(k_m, c)$

Deal with MAC first; factor out in terms of $\mathcal{L}_{\text{mac-real}}$

Security proof

$k_e \leftarrow E.\text{KeyGen}$
$\mathcal{S} := \emptyset$

$\underline{\text{CHALLENGE}(m_L, m_R)}$:
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t := \text{GETMAC}(c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}$
  return (c,t)

$\underline{\text{DEC}(c,t)}$:
  if $(c,t) \in \mathcal{S}$ return null
  if not $\text{VER}(c,t)$:
    return err
  return $E.\text{Dec}(k_e, c)$

$\diamond$

$\mathcal{L}^{M}_{\text{mac-real}}$

$k_m \leftarrow M.\text{KeyGen}$

$\underline{\text{GETMAC}(c)}$:
  return $M.\text{MAC}(k_m, c)$

$\underline{\text{VER}(c,t)}$:
  return $t \stackrel{?}{=} M.\text{MAC}(k_m, c)$

Deal with MAC first; factor out in terms of $\mathcal{L}_{\text{mac-real}}$

Security proof

$k_e \leftarrow E.\mathrm{KeyGen}$
$\mathcal{S} := \emptyset$

<u>CHALLENGE(m_L, m_R):</u>
  $c \leftarrow E.\mathrm{Enc}(k_e, m_L)$
  $t :=$ GETMAC(c)
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}$
  return (c,t)

<u>DEC(c,t):</u>
  if $(c,t) \in \mathcal{S}$ return null
  if not VER(c,t):
    return err
  return $E.\mathrm{Dec}(k_e, c)$

$\diamond$

$\mathcal{L}^{M}_{\text{mac-fake}}$

$k_m \leftarrow M.\mathrm{KeyGen}$
$\mathcal{T} = \emptyset$

<u>GETMAC(c):</u>
  $t := M.\mathrm{MAC}(k_m, c)$
  $\mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return t

<u>VER(c,t):</u>
  return $(c,t) \overset{?}{\in} \mathcal{T}$

Replace $\mathcal{L}_{\text{mac-real}}$ with $\mathcal{L}_{\text{mac-fake}}$

$$
\begin{array}{|l|}
\hline
k_e \leftarrow E.\text{KeyGen} \\
\mathcal{S} := \emptyset \\
\\
\underline{\text{CHALLENGE}(m_L, m_R):} \\
\quad c \leftarrow E.\text{Enc}(k_e, m_L) \\
\quad t := \text{GETMAC}(c) \\
\quad \mathcal{S} := \mathcal{S} \cup \{(c,t)\} \\
\quad \text{return } (c,t) \\
\\
\underline{\text{DEC}(c,t):} \\
\quad \text{if } (c,t) \in \mathcal{S} \text{ return null} \\
\quad \text{if not VER}(c,t): \\
\qquad \text{return err} \\
\quad \text{return } E.\text{Dec}(k_e, c) \\
\hline
\end{array}
\diamond
\begin{array}{|l|}
\hline
\mathcal{L}^{M}_{\text{mac-fake}} \\
\hline
k_m \leftarrow M.\text{KeyGen} \\
\mathcal{T} = \emptyset \\
\\
\underline{\text{GETMAC}(c):} \\
\quad t := M.\text{MAC}(k_m, c) \\
\quad \mathcal{T} := \mathcal{T} \cup \{(c,t)\} \\
\quad \text{return } t \\
\\
\underline{\text{VER}(c,t):} \\
\quad \text{return } (c,t) \overset{?}{\in} \mathcal{T} \\
\hline
\end{array}
$$

Replace $\mathcal{L}_{\text{mac-real}}$ with $\mathcal{L}_{\text{mac-fake}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\textsc{challenge}(m_L, m_R)$:
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c,t)

$\textsc{dec}(c,t)$:
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{T}$:
    return err
  return $E.\text{Dec}(k_e, c)$

Inline the library.

```
k_e ← E.KeyGen
k_m ← M.KeyGen
S := ∅; T := ∅

CHALLENGE(m_L, m_R):
  c ← E.Enc(k_e, m_L)
  t := M.MAC(k_m, c)
  S := S ∪ {(c,t)}; T := T ∪ {(c,t)}
  return (c,t)

DEC(c,t):
  if (c,t) ∈ S return null
  if (c,t) ∉ T:
     return err
  return E.Dec(k_e, c)
```

Inline the library.

Security proof

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\underline{\text{CHALLENGE}(m_L, m_R):}$
$c \leftarrow E.\text{Enc}(k_e, m_L)$
$t := M.\text{MAC}(k_m, c)$
$\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
return (c, t)

$\underline{\text{DEC}(c, t):}$
if $(c,t) \in \mathcal{S}$ return null
if $(c,t) \notin \mathcal{T}$:
  return err
return $E.\text{Dec}(k_e, c)$

Notice: $\mathcal{S}$ and $\mathcal{T}$ are always identical!

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

CHALLENGE(m_L, m_R):
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c, t)

DEC(c, t):
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{S}$:
    return err
  return $E.\text{Dec}(k_e, c)$

Notice: $\mathcal{S}$ and $\mathcal{T}$ are always identical $\Rightarrow$ replace ref to $\mathcal{T}$ with $\mathcal{S}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

<u>CHALLENGE(m_L, m_R):</u>
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c,t)

<u>DEC(c,t):</u>
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{S}$:
    return err
  return $E.\text{Dec}(k_e, c)$

Notice: $\mathcal{S}$ and $\mathcal{T}$ are always identical $\Rightarrow$ replace ref to $\mathcal{T}$ with $\mathcal{S}$

Security proof

```
k_e ← E.KeyGen
k_m ← M.KeyGen
S := ∅; T := ∅

CHALLENGE(m_L, m_R):
  c ← E.Enc(k_e, m_L)
  t := M.MAC(k_m, c)
  S := S ∪ {(c,t)}; T := T ∪ {(c,t)}
  return (c,t)

DEC(c,t):
  if (c,t) ∈ S return null
  if (c,t) ∉ S:
     return err
  return E.Dec(k_e, c)
```

Last line of DEC unreachable

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

<u>CHALLENGE(m_L, m_R):</u>
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c,t)

<u>DEC(c, t):</u>
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{S}$:
    return err

Last line of DEC unreachable $\Rightarrow$ remove it

```
k_e ← E.KeyGen
k_m ← M.KeyGen
S := ∅; T := ∅

CHALLENGE(m_L, m_R):
  c ← E.Enc(k_e, m_L)
  t := M.MAC(k_m, c)
  S := S ∪ {(c,t)}; T := T ∪ {(c,t)}
  return (c,t)

DEC(c,t):
  if (c,t) ∈ S return null
  if (c,t) ∉ S:
     return err
```

With E.Dec gone, we can factor out in terms of $\mathcal{L}^{E}_{\text{cpa-L}}$.

Security proof

$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\text{CHALLENGE}(m_L, m_R)$:
  $c := \text{CHALLENGE}'(m_L, m_R)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c,t)

$\text{DEC}(c,t)$:
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{S}$:
    return err

$\diamond$

$\mathcal{L}^{E}_{\text{cpa-L}}$

$k_e \leftarrow E.\text{KeyGen}$

$\text{CHALLENGE}'(m_L, m_R)$:
  $c := E.\text{Enc}(k_e, m_L)$
  return c

With E.Dec gone, we can factor out in terms of $\mathcal{L}^{E}_{\text{cpa-L}}$.

$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\underline{\text{CHALLENGE}(m_L, m_R):}$
  $c := \text{CHALLENGE}'(m_L, m_R)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c,t)

$\underline{\text{DEC}(c,t):}$
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{S}$:
    return err

$\diamond$

$\mathcal{L}^{E}_{\text{cpa-L}}$

$k_e \leftarrow E.\text{KeyGen}$

$\underline{\text{CHALLENGE}'(m_L, m_R):}$
  $c := E.\text{Enc}(k_e, m_L)$
  return c

With E.Dec gone, we can factor out in terms of $\mathcal{L}^{E}_{\text{cpa-L}}$.

Left library:

```
k_m ← M.KeyGen
S := ∅; T := ∅

CHALLENGE(m_L, m_R):
  c := CHALLENGE′(m_L, m_R)
  t := M.MAC(k_m, c)
  S := S ∪ {(c,t)}; T := T ∪ {(c,t)}
  return (c,t)

DEC(c,t):
  if (c,t) ∈ S return null
  if (c,t) ∉ S:
    return err
```

$\diamond$

$\mathcal{L}^{E}_{\text{cpa-R}}$

```
k_e ← E.KeyGen

CHALLENGE′(m_L, m_R):
  c := E.Enc(k_e, m_R)
  return c
```

Replace $\mathcal{L}_{\text{cpa-L}}$ with $\mathcal{L}_{\text{cpa-R}}$.

Security proof

$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\underline{\textsc{challenge}(m_L, m_R):}$
  $c := \textsc{challenge}'(m_L, m_R)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c, t)

$\underline{\textsc{dec}(c, t):}$
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{S}$:
    return err

$\diamond$

$\mathcal{L}^{E}_{\text{cpa-R}}$

$k_e \leftarrow E.\text{KeyGen}$

$\underline{\textsc{challenge}'(m_L, m_R):}$
  $c := E.\text{Enc}(k_e, m_R)$
  return c

Replace $\mathcal{L}_{\text{cpa-L}}$ with $\mathcal{L}_{\text{cpa-R}}$.

Security proof

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\underline{\text{CHALLENGE}(m_L, m_R):}$
$c \leftarrow E.\text{Enc}(k_e, m_R)$
$t := M.\text{MAC}(k_m, c)$
$\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
return (c, t)

$\underline{\text{DEC}(c, t):}$
if $(c,t) \in \mathcal{S}$ return null
if $(c,t) \notin \mathcal{S}$:
return err

Inline $\mathcal{L}_{\text{cpa-R}}$.

$k_e \leftarrow E.\mathrm{KeyGen}$
$k_m \leftarrow M.\mathrm{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\underline{\textsc{challenge}(m_L, m_R):}$
  $c \leftarrow E.\mathrm{Enc}(k_e, m_R)$
  $t := M.\mathrm{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c, t)

$\underline{\textsc{dec}(c, t):}$
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{S}$:
    return err

Inline $\mathcal{L}_{\text{cpa-R}}$.

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\underline{\text{CHALLENGE}(m_L, m_R)\text{:}}$
  $c \leftarrow E.\text{Enc}(k_e, m_R)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c, t)

$\underline{\text{DEC}(c, t)\text{:}}$
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{T}$:
    return err
  return $E.\text{Dec}(k_e, c)$

Add unreachable statement; Change ref from $\mathcal{S}$ to $\mathcal{T}$.

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset; \mathcal{T} := \emptyset$

$\underline{\text{CHALLENGE}(m_L, m_R):}$
  $c \leftarrow E.\text{Enc}(k_e, m_R)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c,t)\}; \mathcal{T} := \mathcal{T} \cup \{(c,t)\}$
  return (c, t)

$\underline{\text{DEC}(c,t):}$
  if $(c,t) \in \mathcal{S}$ return null
  if $(c,t) \notin \mathcal{T}$:
    return err
  return $E.\text{Dec}(k_e, c)$

Add unreachable statement; Change ref from $\mathcal{S}$ to $\mathcal{T}$.

Security proof

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$

CHALLENGE(m_L, m_R):

  $c \leftarrow E.\text{Enc}(k_e, m_R)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
  return (c, t)

DEC(c, t):

  if $(c, t) \in \mathcal{S}$ return null
  if $t \neq \text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$

Replace "fake" MAC verification with "real" (steps omitted).

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$

CHALLENGE(m_L, m_R):
$c \leftarrow E.\text{Enc}(k_e, m_R)$
$t := M.\text{MAC}(k_m, c)$
$\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
return (c, t)

DEC(c, t):
if $(c, t) \in \mathcal{S}$ return null
if $t \neq \text{MAC}(k_m, c)$:
return err
return $E.\text{Dec}(k_e, c)$

Replace “fake” MAC verification with “real” (steps omitted).

$\mathcal{L}^{EtM}_{\text{cca-R}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$

CHALLENGE(m_L, m_R):
$c \leftarrow E.\text{Enc}(k_e, m_R)$
$t := M.\text{MAC}(k_m, c)$
$\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
return (c, t)

DEC(c, t):
if $(c, t) \in \mathcal{S}$ return null
if $t \neq \text{MAC}(k_m, c)$:
return err
return $E.\text{Dec}(k_e, c)$

Result is $\mathcal{L}_{\text{cca-R}}$!

Summary

We showed:

$\mathcal{L}^{EtM}_{\text{cca-L}}$		$\mathcal{L}^{EtM}_{\text{cca-R}}$

$k_e \leftarrow E.\text{KeyGen}$
$k_m \leftarrow M.\text{KeyGen}$
$\mathcal{S} := \emptyset$
 	
CHALLENGE(m_L, m_R):
  $c \leftarrow E.\text{Enc}(k_e, m_L)$
  $t := M.\text{MAC}(k_m, c)$
  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
  return (c, t)
 	
DEC(c, t):
  if $(c, t) \in \mathcal{S}$ return null
  if $t \neq \text{MAC}(k_m, c)$:
    return err
  return $E.\text{Dec}(k_e, c)$
	$\approx$	$k_e \leftarrow E.\text{KeyGen}$
		$k_m \leftarrow M.\text{KeyGen}$
		$\mathcal{S} := \emptyset$
		
		CHALLENGE(m_L, m_R):
		  $c \leftarrow E.\text{Enc}(k_e, m_R)$
		  $t := M.\text{MAC}(k_m, c)$
		  $\mathcal{S} := \mathcal{S} \cup \{(c, t)\}$
		  return (c, t)
		
		DEC(c, t):
		  if $(c, t) \in \mathcal{S}$ return null
		  if $t \neq \text{MAC}(k_m, c)$:
		    return err
		  return $E.\text{Dec}(k_e, c)$

So our scheme is a CCA-secure encryption scheme.