Chosen Ciphertext Attacks:

Revisit Padding oracle attacks

- If server that accepts c^* and leaks whether $\text{CBC-Dec}(k, c^*)$ has valid padding
- If $c_0 c_1 \cdots c_\ell = \text{CBC-Enc}(k, m_1, \ldots, m_\ell)$ (unknown)

  then $\text{CBC-Dec}(k, (c_{i-1} \oplus x, c_i)) = m_i \oplus x$

  then Adv can choose x, i

  learn whether $m_i \oplus x$ ends in

  01, 0002, 000003, etc ...
- use this ability to decrypt all of m

  (just like web demo)

Problem:

Adversary got some partial information about $\text{Dec}(k, c)$, for adversarially chosen c } not captured in CPA definition

no Dec! →

$k \leftarrow \text{KeyGen}$	$k \leftarrow \text{KeyGen}$
$\text{CHAL}(m_L, m_R)$:	$\text{CHAL}(m_L, m_R)$:
$c = \text{Enc}(k, m_L)$	$c = \text{Enc}(k, m_R)$
ret c	ret c

$\approx$

need a stronger sec. definition!

Want: ctxt leak nothing about ptxt (CPA)

... even when Adv can get ~~partial info about~~ decryption of chosen ciphertexts

hard to anticipate which partial info might exist

```
k ← KeyGen
CHAL(m_L, m_R):
    c = Enc(k, m_L)
    ret c
GETDEC(c):
    ret Dec(k, c)
```

$\approx$

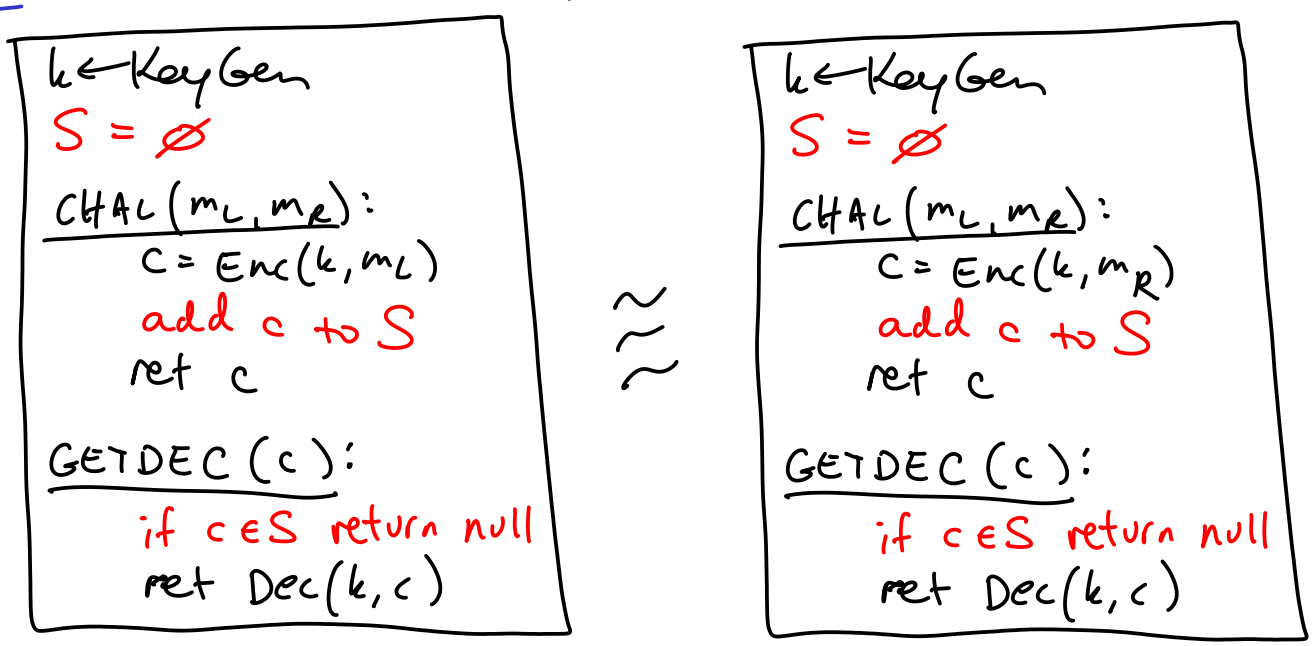
```
k ← KeyGen
CHAL(m_L, m_R):
    c = Enc(k, m_R)
    ret c
GETDEC(c):
    ret Dec(k, c)
```

But you can always distinguish these two (so it gives an impossible security def)

```
pick m_L, m_R arbitrarily
c = CHAL(m_L, m_R)
return m_L ≟ GETDEC(c)
```

fix: don't allow Adv to decrypt c generated by CHAL subroutine

Def: CCA (chosen ciphertext attack) security



ctxt leak nothing about ptxt

... even when Adv can get decryption of chosen ciphertexts

ANYOTHER