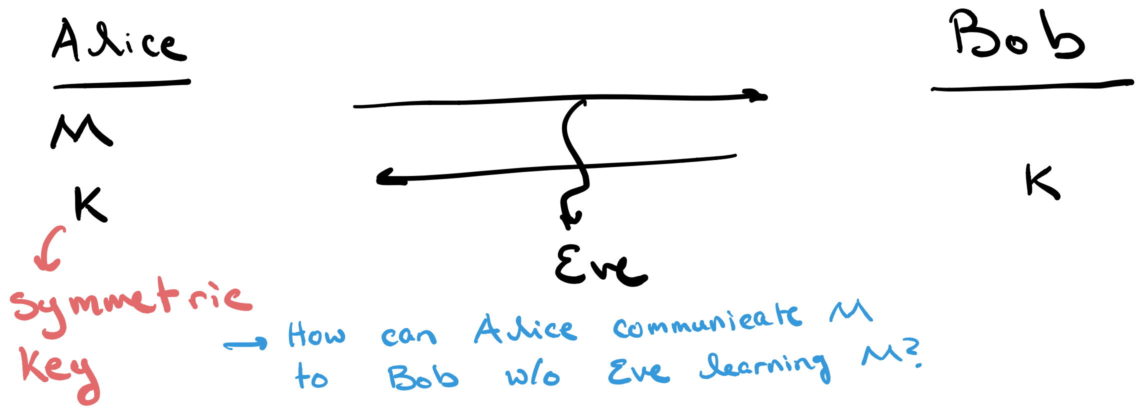
Symmetric Key Encryption



→ How can Alice communicate M to Bob w/o Eve learning M?

$c \leftarrow Enc(K, M)$ —— c ——→ $m \leftarrow Dec(K, c)$

↓ needs to provide confidentiality ie. c to hide all information about M besides the length

→ why? Assume some static CT size n

1) Can't encrypt messages longer than n
2) Encrypting small messages is wasteful

Symmetric Encryption Scheme (API):

Keygen() → K

Enc(K, M) → C

Dec(K, C) → M

Correctness: $\forall K\ \forall M,\ c \leftarrow Enc(K, M)$:
$Dec(K, C) = M$

Security: ?

→ Adv. knows Keygen, Enc, Dec but doesn't know K

Naive Idea: Given c, an Adv. can't recover m

→ not good enough. Doesn't deal w/ partial info. leakage

Ex.

1) Database which holds deterministic encryptions of students' grades

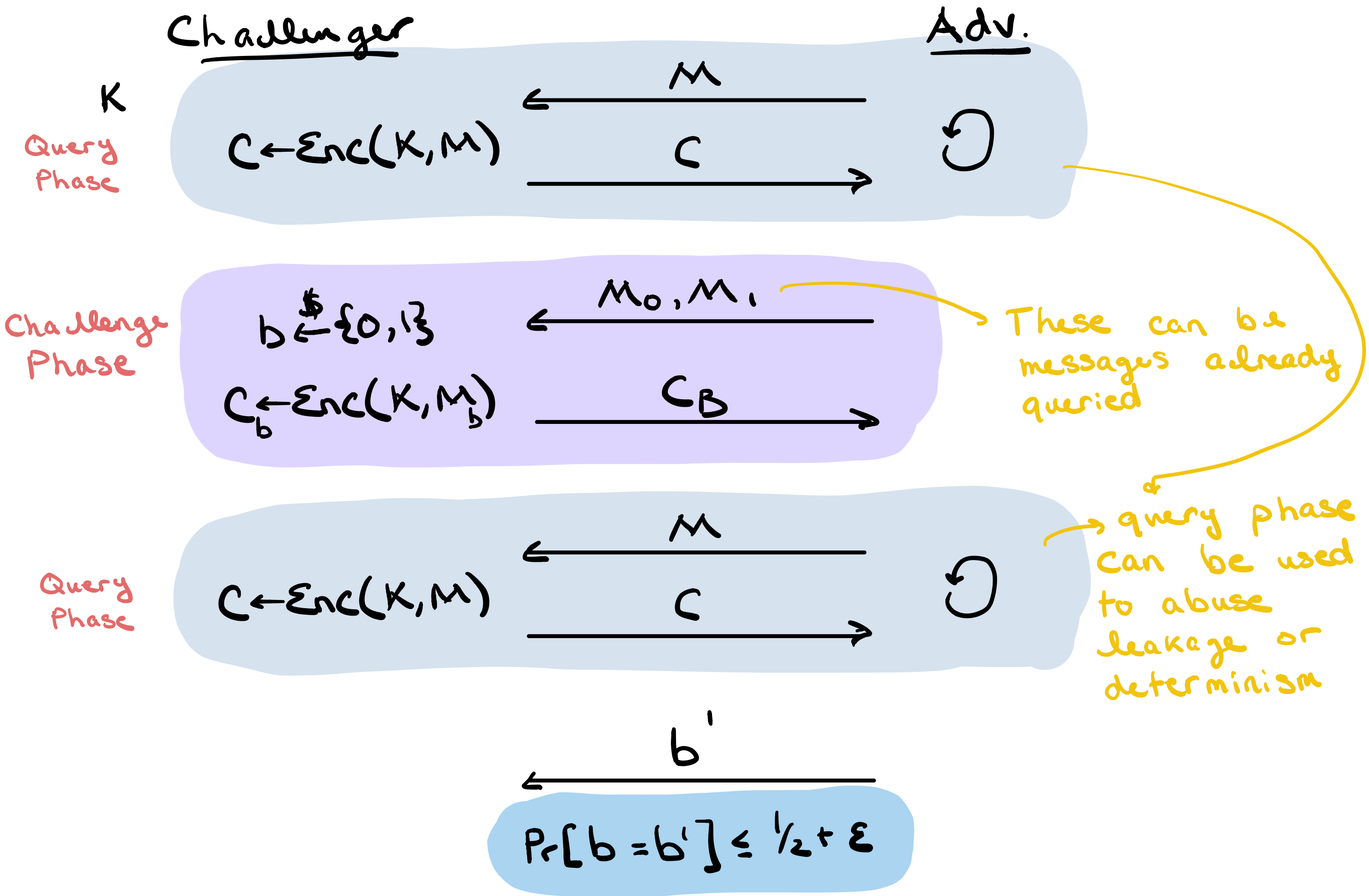
→ Adv. can learn which students have the same grade

→ Given value of one CT, the Adv. can decrypt many

2) Database which holds encrypted hospital records which indicate whether a patient has cancer or not (Yes/No). Enc leaks first letter of message.

→ Adv. can recover m 100% of the time

Goal: No partial info about M may leak b/c an Adv. can couple it w/ side info. to reconstruct M


Challenger
Adv.
K
Query Phase
M
C←Enc(K,M)
C
Challenge Phase
M0, M1
b ←$ {0,1}
Cb←Enc(K,Mb)
CB
These can be messages already queried
Query Phase
M
C←Enc(K,M)
C
query phase can be used to abuse leakage or determinism
b'
Pr[b = b'] ≤ ½ + ε

IND-CPA ensures a correct scheme is:

1) Non-deterministic

→ If not, we can query the same messages used in the challenge

2) Confidential

→ If not, we can make queries to leak which challenge message was chosen

- For <u>all</u> adversaries!