Symmetric Key Encryption

Alice
\[ M \]
\[ K \]
\[ c \text{ symmetric}\]

How can Alice communicate \( M \) to Bob with Eve learning \( M \)?

Bob
\[ K \]

\[ Enc \]

\[ c \leftarrow Enc(K, M) \]
\[ m \leftarrow Dec(K, c) \]

Needs to provide confidentiality i.e. \( c \) 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() \( \rightarrow K \)
Enc(k, M) \( \rightarrow C \)
Dec(K, C) \( \rightarrow M \)

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
The diagram illustrates a cryptographic protocol involving a challenger and an adversary (Adv.).

**Query Phase**
- The challenger generates an encryption of a message `M` using a key `K` and a scheme `C ≜ \text{Enc}(K, M)`.
- The adversary `O` is given `C`.

**Challenge Phase**
- `b \leftarrow \{0, 1\}` is randomly chosen.
- `M_0, M_1` are two messages.
- `C_b \leftarrow \text{Enc}(K, M_b)` is generated.

**Query Phase** (continued)
- Another encryption `C` of `M` is generated using the same key `K`.

The adversary `O` can query messages `M_0` and `M_1` already tested, and `Query phase can be used to abuse leakage or determinism`.

Additionally, it is stated that `\Pr[b = b'] \leq \frac{1}{2} + \epsilon`,
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 all adversaries!