# Key derivation

**Typical scenario**: a single **source key (SK)** is sampled from:

* Hardware random number generator
* A key exchange protocol (discussed later)

Need many keys to secure session:

* unidirectional keys, multiple keys for nonce-based CBC

**Goal**: Generate many keys from this one source key

F: a PRF with key space (K) and outputs in {0,1}n

Suppose source key SK is uniform in K

Define Key Derivation Function (KDF) as:

KDF (SK, CTX, L) := F ( SK, (CTX||0)) || F(SK, (CTX||1)) || … || F(SK, (CTX||L))

CTX = a string that uniquely identifies the application

## What if source key is not uniform?

Recall: PRFs is pseudorandom only when key is uniform in K

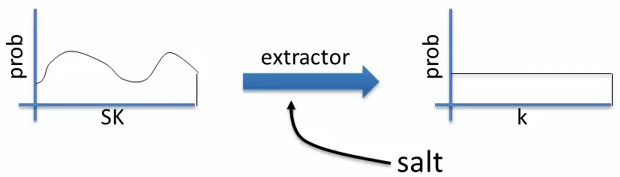
SK not uniform ⇒ PRF output may not look random

Source key often not uniformly random:

* Key exchange protocol: key uniform in some subset of K
* Hardware RNG: may produce biased output

## Solution: Extract-and-Expand paradigm

**Step 1**: extract pseudo-random key k from source key SK



The salt is a fixed non-secret string chosen at random.

**Step 2**: Expand k by using it as a PRF key as before

## HKDF: a KDF from HMAC

Implements de extract-and-expand paradigm

Extract: use k ← HMAC(salt, SK)

The salt is used as the HMAC key, and the Source Key as the HMAC data

Then expand using HMAC as a PRF with key k

## Password based KDF (PBKDF)

Deriving keys from passwords:

* Do not use HKDF: passwords have insufficient entropy
* Derived keys will be vulnerable to dictionary attacks

PNKDF defenses: salt and a slow hash function

Standard approach: **PKCS#5** (PBKDF1)

**H(c) (pwd || salt)**: iterate hash function c times

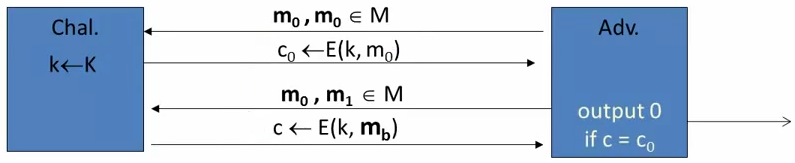
# Deterministic encryption

The problem: Deterministic encryption can’t be CPA secure

Attacker can tell when two ciphertext encrypt the same message ⇒ leaks information

This leads to significant attacks when space M is small

According to the classic CPA security definition, the attacker could do:



And can always determine the right output. So Adv = 1

## A solution: The case of unique messages

Suppose encryptor never encrypts same message twice, i.e. the pair (k,m) never repeats

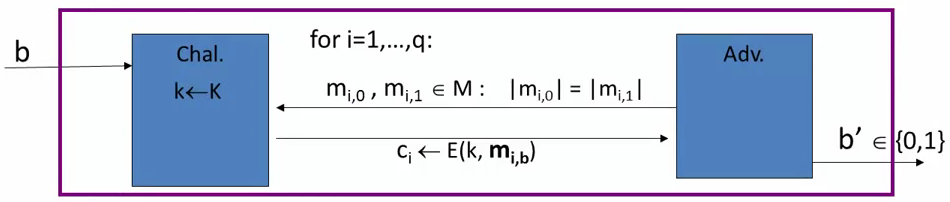
This happens when encryptor:

Chooses messages at random from a large msg space (e.g. keys)

Message structure ensures uniqueness (e.g. unique user ID)

## Deterministic CPA security

EBIG = (E,D) a cipher defined over (K,M,C). For b=0,1 define EXP(b) as:



where m1,0, … , mq,0 are distinct and m1,1, … , mq,1 are distinct.

Def: EBIC is semantically secure under deterministic CPA if for all “efficient” A:

AdvdCPA[A, EBIG] = | Pr[EXP(0)=1] - Pr[EXP(1)=1] | is negligible