<u>Nonce-based counter mode</u>: divide IV into two pieces: IV = nonce ‖ counter

↑ value that does not repeat

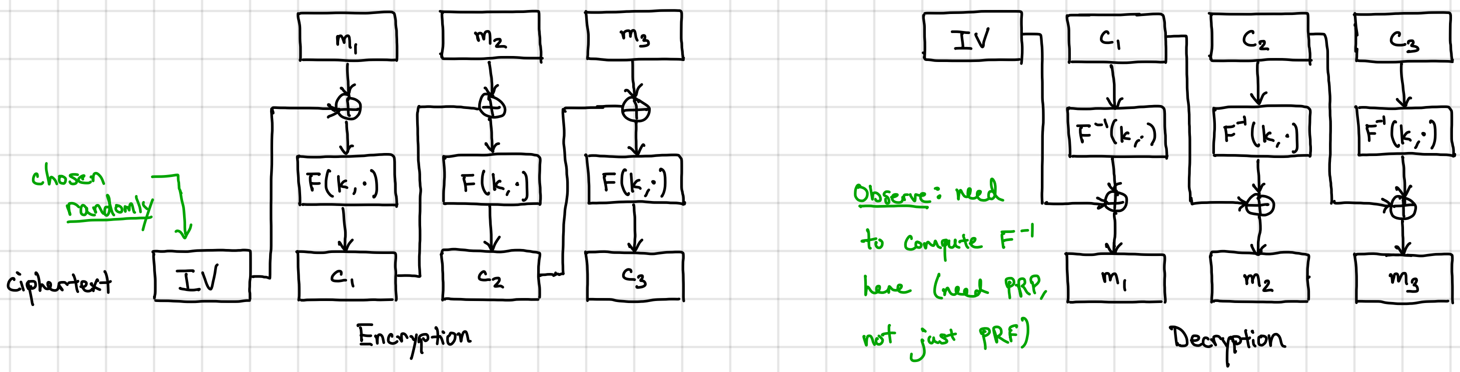
common choices: 64-bit nonce, 64-bit counter; 96-bit nonce, 32-bit counter $\Big\}$ only nonce needs to be sent! (slightly smaller ciphertexts)

Only requirement for security is that IV does <u>not</u> repeat:
- <u>Option 1</u>: Choose randomly (either IV or nonce)
- <u>Option 2</u>: If sender + recipient have shared state (e.g., packet counter), can just use a counter, in which case, IV/nonce does not have to be sent

(CTR)
Counter mode is parallelizable, simple-to-implement, just requires PRF — preferred mode of using block ciphers

Other block cipher modes of operation:

Cipherblock chaining (CBC): common mode in the past (e.g., TLS 1.0, still widely used today)



<u>Theorem</u>: Let $F: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure PRF and let Π_{CBC} denote the CBC encryption scheme for ℓ-block messages ($\mathcal{M} = \mathcal{X}^{\leq \ell}$). Then, for all efficient CPA adversaries $\mathcal{A}$, there exists an efficient PRF adversary $\mathcal{B}$ such that

$$\mathrm{CPAAdv}[\mathcal{A}, \Pi_{CBC}] \leq \frac{2Q^2\ell^2}{|\mathcal{X}|} + 2 \cdot \mathrm{PRFAdv}[\mathcal{B}, F]$$

Q: number of encryption queries
ℓ: number of blocks in message

<u>Intuition</u>: similar to analysis of randomized counter mode:
1. Ciphertext is indistinguishable from random string if PRP is evaluated on distinct inputs
2. When encrypting, PRP is invoked on ℓ random blocks, so after Q queries, we have $Q\ell$ random blocks.
   $\Rightarrow$ Collision probability $\leq \frac{Q^2\ell^2}{|\mathcal{X}|}$ ⟵ this is larger than collision prob. for randomized counter mode by a factor of $\frac{\ell}{2}$ [overlap of Q random intervals vs. $Q\ell$ random points]
3. Factor of 2 arises for same reason as before

<u>Interpretation</u>. CBC mode provides weaker security compared to counter mode: $\frac{2Q^2\ell^2}{|\mathcal{X}|}$ vs. $\frac{4Q^2\ell}{|\mathcal{X}|}$

Concretely: for same parameters as before (1 MB messages, 2^{-32} distinguishing advantage):

$$Q \leq \sqrt{\frac{|\mathcal{X}| \cdot 2^{-32}}{2\ell^2}} = \sqrt{\frac{2^{128} \cdot 2^{-32}}{2(2^{16})^2}} = \sqrt{2^{63}} = 2^{31.5} \quad (\sim 1 \text{ billion messages})$$

↳ $2^{7.5} \sim 180\times$ smaller than using counter mode

Padding in CBC mode: each ciphertext block is computed by feeding a message block into the PRP

$\Rightarrow$ message must be an even multiple of the block size

$\Rightarrow$ when used in practice, need to pad messages

Can we pad with zeroes? Cannot decrypt! What if original message ended with a bunch of zeroes?

Requirement: padding must be invertible

CBC padding in TLS 1.0: if k bytes of padding is needed, then append k bytes to the end, with each byte set to $k-1$

(for AES-CBC) if 0 bytes of padding is needed, then append a block of 16 bytes, with each byte equal to 15

↳ dummy block needed to ensure pad is invertible

[injective functions must expand: $|\{0,1\}^{\leq 256}| > |\{0,1\}^{256}|$]

↳ called PKCS#5/PKCS#7 (public-key cryptography standards)

Need to pad in CBC encryption can be exploited in "padding oracle" attacks

Padding in CBC can be avoided using idea called "ciphertext stealing" (as long as messages are more than 1 block)

Comparing CTR mode to CBC mode:

CTR mode	CBC mode
1. no padding needed (shorter ciphertexts)	1. padding needed
2. parallelizable	2. sequential
3. only requires PRF (no need to invert)	3. requires PRP
4. tighter security	4. less tight security (re-key more often)
5. IVs have to be non-repeating (and spaced far apart)	5. requires unpredictable IVs

imagine 1 byte messages (e.g., encrypted key strokes over SSH)

1 block + 1 byte with CTR

2 blocks with CBC

interesting traffic analysis attack: each keystroke is sent in separate packet, so # packets leaks info on length of user's password!

requires more structured primitive, more code to implement forward and backward evaluation

[easy to implement: IV = nonce || counter]

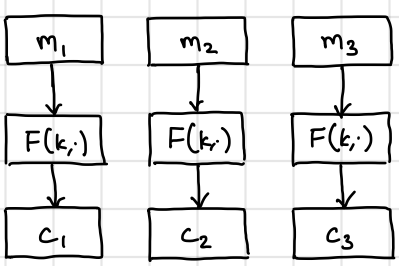
only needs to be non-repeating (can be predictable)

TLS 1.0 used predictable IVs (see HW1 for an attack)

SSH v1 used a 0 IV (even worse!)

Bottom-line: use randomized or nonce-based counter mode whenever possible: simpler, easier, and better than CBC!

A tempting and bad way to use a block cipher: ECB mode (electronic codebook)



Scheme is deterministic! Cannot be CPA secure!

Not even semantically secure!

(m_0, m_0) vs. (m_0, m_1) where $m_1 \neq m_0$

ciphertext blocks output are same

ciphertext blocks output are different

Encryption: simply apply block cipher to each block of the message

Decryption: simply invert each block of the ciphertext

NEVER USE ECB MODE FOR ENCRYPTION!